

Improving Drug Discovery Decision Making using Machine Learning and Graph Theory in QSAR Modeling

Ernst Ahlberg Helgee



UNIVERSITY OF GOTHENBURG

Doctoral Thesis
Department of Chemistry
University of Gothenburg
Gothenburg, Sweden 2010

Improving Drug Discovery Decision Making using
Machine Learning and Graph Theory in QSAR Modeling

E A H

Cover illustration shows how different machine-learning models approximate a function.

Original Function	PLS approximation
RF approximation	SVM approximation

ISBN 978-91-628-8018-7

Electronically published at: <http://hdl.handle.net/2077/21838>

© 2010 by Ernst Ahlberg Helgee

Department of Chemistry

University of Gothenburg

SE-412 96 Göteborg, Sweden

Author email address: ernst.ahlberghelgee@gmail.com

Printed by Chalmers Reproservice

Göteborg, Sweden 2010

In a world without walls and fences, who needs windows and gates?

Abstract

During the last decade non-linear machine-learning methods have gained popularity among QSAR modelers. The machine-learning algorithms generate highly accurate models at a cost of increased model complexity where simple interpretations, valid in the entire model domain, are rare.

This thesis focuses on maximizing the amount of extracted knowledge from predictive QSAR models and data. This has been achieved by the development of a descriptor importance measure, a method for automated local optimization of compounds and a method for automated extraction of substructural alerts. Furthermore different QSAR modeling strategies have been evaluated with respect to predictivity, risks and information content.

To test hypotheses and theories large scale simulations of known relations between activities and descriptors have been conducted. With the simulations it has been possible to study properties of methods, risks, implementations and errors in a controlled manner since the correct answer has been known. Simulation studies have been used in the development of the generally applicable descriptor importance measure and in the analysis of QSAR modeling strategies. The use of simulations is spread in many areas, but not that common in the computational chemistry community. The descriptor importance measure developed can be applied to any machine-learning method and validations using both real data and simulated data show that the descriptor importance measure is very accurate for non-linear methods.

An automated method for local optimization of compounds was developed to partly replace manual searches made to optimize compounds. The local optimization of compounds make use of the information in available data and deterministically enumerates new compounds in a space spanned close to the compound of interest. This can be used as a starting point for further compound optimization and aids the chemist in finding new compounds. An other approach to guide chemists in the process of optimizing compounds is through substructural warnings. A fast method for significant substructure extraction has been developed that extracts significant substructures from data with respect to the activity of the compound. The method is at least on par with existing methods in terms of accuracy but is significantly less time consuming.

Non-linear machine-learning methods have opened up new possibilities for QSAR modeling that changes the way chemical data can be handled by model algorithms. Therefore properties of *Local* and *Global* QSAR modeling strategies have been studied. The results show that *Local* models come with high risks and are less accurate compared to *Global* models.

In summary this thesis shows that *Global* QSAR modeling strategies should be applied preferably using methods that are able to handle non-linear relationships. The developed methods can be interpreted easily and an extensive amount of information can be retrieved. For the methods to become easily available to a broader group of users packaging with an open-source chemical platform is needed.

Keywords: machine-learning, QSAR, descriptor importance, local and global models, method of manufactured solutions, automated compound optimization, drug design

Contents

List of Figures	iii
List of Tables	iii
List of Publications	iv
Contribution Report	v
Assorted Definitions and Abbreviations	vi
1 Introduction	1
1.1 Background and Significance	1
1.2 QSAR	2
1.2.1 QSAR History	2
1.2.2 Concept of Inverse QSAR	3
1.2.3 Traditional QSAR Modeling Approaches	4
1.3 Machine Learning	4
1.3.1 Random Forests	5
1.3.2 Support Vector Machines	6
1.3.3 Partial Least Squares	6
1.3.4 Descriptor Importance Measures	7
1.4 Simulations - a Way to Deeper Understanding	8
1.4.1 Example	8
1.5 Graph Theory	10
1.6 Molecular Representation	10
1.6.1 Signatures	11
1.7 Enumeration of New Compounds from a Set of Signatures	12
1.8 Finding Substructural Alerts in Data	17
2 Contribution to the Field	19
2.1 Model Interpretability	19
2.1.1 Theory	19
2.2 Automated Compound Optimization	21
2.2.1 Theory	22
2.3 Finding Significant Substructures	24
2.4 Evaluation of QSAR Modeling Strategies	26

3	Results and Discussion	31
3.1	Model Interpretability	31
3.2	Automated Compound Optimization	31
3.3	Finding Significant Substructures	35
3.4	Evaluation of QSAR Modeling Strategies	37
3.4.1	Experimental Setup	37
3.4.2	Computational Costs	43
3.4.3	Discussion	43
4	Concluding Remarks and Future Perspective	45
	References	50
	Acknowledgments	vii

List of Figures

1	Contour plots of the function for <i>Data set I</i>	9
2	A tree representation of methane and the corresponding atom signatures	12
3	Visualization of how to create compounds from signatures	13
4	Compounds based on linear combinations of the solutions in Table 1.	15
5	A signature and a SMARTS highlighted on a compound	20
6	Flowchart describing local optimization of compounds	23
7	QSAR modeling strategies	27
8	Contour plots of gradients for <i>Data set I</i>	32
9	Example of compound to optimize	33
10	Average test and validation accuracy for the methods used on the AMES data .	36
11	Average training and prediction times for the methods used on the AMES data .	36
12	The logarithm of the number of generated alerts for each method	37
13	Random Forest results from the <i>Local</i> and <i>Global</i> modeling strategies	40
14	Results from the <i>Local</i> and <i>Global</i> modeling strategies	41
15	Partial Least Squares results from the <i>Local</i> and <i>Global</i> modeling strategies . .	42
16	<i>Local</i> and <i>Global</i> modeling strategies, computational effort	43

List of Tables

1	Solutions to the system of equations in Figure 3(c).	14
---	--	----

List of Publications

The thesis is based on the following publications, which are referred to in the text by the Roman numerals I – IV. The papers are appended at the end of the thesis. Reprints are made with kind permission from the publishers.

I. Interpretation of Non-Linear QSAR Models Applied to Ames Mutagenicity Data

Carlsson, Lars; Ahlberg Helgee, Ernst; Boyer, Scott

J. Chem. Inf. Model. **2009**, *49*, pp. 2551 - 2558

Supporting information available at <http://pubs.acs.org/doi/suppl/10.1021/ci9002206>

II. A Method for Automated Molecular Optimization Applied to Ames Mutagenicity Data

Ahlberg Helgee, Ernst; Carlsson, Lars; Boyer, Scott

J. Chem. Inf. Model. **2009**, *49*, pp. 2559 - 2563

Supporting information available at <http://pubs.acs.org/doi/suppl/10.1021/ci900221r>

III. Mining Chemical Data for Significant Substructures using Signatures

Ahlberg Helgee, Ernst; Carlsson, Lars; Boyer, Scott

to be submitted to *BMC Bioinf.*

IV. Evaluation of Quantitative Structure Activity Relationship Modeling Strategies: Local and Global Models

Ahlberg Helgee, Ernst; Carlsson, Lars; Boyer, Scott; Norinder, Ulf

J. Chem. Inf. Model. under revision

Contribution Report

- P I Contributed to the formulation of the research problem.
Implemented the method for the simulation studies using R and performed the experimental parts of the paper conducted in R. Contributed to the interpretation of the results and writing the paper.
- P II Major contribution to the formulation of the research problem, implemented and tested the algorithms using Python and C++. Performed all computations for the paper including analysis of the results. Main author of the paper.
- P III Major contribution to the formulation of the research problem, implemented and tested the method using Python. Performed all computations for the paper including analysis of the results. Main author of the paper.
- P IV Major contribution to the formulation of the research problem, implemented and tested the strategies using Python and R. Performed all computations for the paper including analysis of the results. Main author of the paper.

Abbreviations

PLS Partial Least Squares

QSAR Quantitative Structure-Activity Relationship

RBF Radial Basis Function

RF Random Forest

RMSE Root-Mean Square Error

SMARTS SMiles ARbitrary Target Specification

SMILES Simplified Molecular Input Line Entry Specification

SVM Support Vector Machines

1. Introduction

1.1. Background and Significance

In general terms, the drug discovery process consists of several parts. The first objective is to decide on a disease area in which research will be invested. The disease area defines the field where a future drug is supposed to act, for instance the gastrointestinal system. Within this field it is important to find and establish a biological target which affects the disease and which is possible to regulate using a pharmaceutical intervention. Once this is established an assay is set up to test compounds for activity towards the target. When the assay is in place a screening is performed to find compounds that have some initial activity towards the target, altering the biological activity of the target such that a change in disease course or symptoms is achieved. When a set of active compounds have been found an iteration loop starts that aims at optimizing one or several series of compounds for the primary target and a range of safety targets and parameters to make sure that the compound will enter the body in an active form and find its way to the active site. At the active site the compound should perform its task for a period of time, usually a few hours, and then the compound should be degraded by the mechanisms in the body and leave, without turning into reactive metabolites.

All the above criteria need to be fulfilled before a compound can become a drug. To develop a drug based on experimental testing only is a very expensive task and that is where *computational chemistry* comes in. With computational methods it is possible to predict some aspects of the behavior of compounds based on their structure. One common method for this is referred to as Quantitative Structure-Activity Relationship (QSAR) modeling. To construct a QSAR model a set of compounds with known activity is needed. New compounds are then predicted using the model. QSAR models are widely used for prediction of compound activities at various biological targets and the general inferences from these models guide chemists seeking changes to optimize their compounds.

Making a predictive QSAR model is not an easy task, and making it interpretable is even harder. In this work methods and theories have been developed to aid and guide modelers and chemists in their everyday work. Firstly a method of non-linear QSAR model interpretation was developed and then extended with a method to automatically replace substructures with undesired properties in potential drug compounds. Then a method for automated extraction of sub-structural alerts from a data set was developed. Finally an investigative study is presented that compares different QSAR modeling strategies.

1.2. QSAR

The aim of QSAR modeling is to obtain a relation between structures or properties of compounds and a measured activity to be able to predict the activity of new compounds and determine mechanisms of action on this activity. The structures and properties of the compounds are expressed by variables referred to as descriptors. Thus, QSAR represents an attempt to mathematically correlate a set of descriptors with activities by the use of statistics. This means that any QSAR model is an approximation of a relation between the activity, y and the descriptors, \mathbf{x} that can be viewed as a mathematical function, $y = f(\mathbf{x})$.

To set up a QSAR model a data set is needed containing compounds with known activity. Activities used in QSAR equations include chemical measurements and responses from biological assays. From the chemical structure of the compounds descriptors are derived representing properties or structures (which for example can include physicochemical parameters to account for hydrophobicity, topology, electronic properties and steric effects) that are present in the compounds. The descriptors can be determined empirically or more commonly by computational methods.¹ When both activity and descriptors exist for the data a machine-learning method can be used to approximate the function f .

QSAR methods are currently being applied in many disciplines, among them are drug design and environmental risk assessment.²⁻⁴ Historically only linear models were used and these are still popular today due to the straight forward interpretation of the results. For this reason non-linear models are commonly viewed as hard to interpret and these model algorithms or models are sometimes referred to as black-box models.^{5,6} However, non-linear models have the potential to more accurately describe important phenomena but there is a need for a simple method for knowledge extraction from these models, which is an objective of the work presented in this thesis.

1.2.1. QSAR History

QSAR dates back to the 19th century. In 1863, A.F.A. Crois at the University of Strasbourg observed that toxicity of alcohols to mammals increased as the water solubility of the alcohols decreased.⁷ In the 1890's, Hans Horst Meyer of the University of Marburg and Charles Ernest Overton of the University of Zurich, working independently, noted that the toxicity of organic compounds were dependent on the lipophilicity.^{7,8}

Little additional development of QSAR occurred until the work of Louis Hammett,⁹ who correlated electronic properties of organic acids and bases with their equilibrium constants and reactivity. Hammett observed that adding substituents to the aromatic ring of benzoic acid had an orderly and quantitative effect on the dissociation constant. Hammett also observed that substituents have a similar effect on the dissociation of other organic acids and bases.

QSARs based on Hammett's relationship utilize electronic properties as descriptors. Diffi-

culties were encountered when investigators attempted to apply Hammett-type relationships to biological systems, indicating that other structural descriptors were necessary.

Robert Muir, a botanist at Pomona College, was studying the biological activity of compounds that resembled indoleacetic acid and phenoxyacetic acid,¹⁰ which function as plant growth regulators. In an attempt to correlate the structures of the compounds with their activities, he consulted Corwin Hansch. Using Hammett sigma parameters to account for the electronic effect of substituents did not lead to a meaningful QSAR. However, Hansch recognized the importance of lipophilicity, expressed as the octanol-water partition coefficient, on biological activity.¹¹ This parameter is recognized to provide a measure of membrane permeability, since a compound needs to have lipophilic properties to enter a membrane and hydrophilic properties to pass through. The octanol-water partition coefficient is also a driving force when drugs bind into targets.

QSAR models are now developed using a variety of parameters such as descriptors of the structural properties of molecules, descriptors to account for the shape, size, lipophilicity, polarisability, and other properties.¹²

1.2.2. Concept of Inverse QSAR

The concept of Inverse QSAR, IQSAR, is to take a desired activity and find the descriptors,^{13,14} *i.e.* finding $\mathbf{x} = f^{-1}(y)$. With this information it is possible to find compounds that have the desired properties.¹⁵ This translates into an effort to build compounds with superior properties towards one or several chemical or biological targets. This means that the modeler select activity ranges that are beneficial and the model determines descriptor values that correspond to the preferences. Based on the selected parameters matching compounds are built. Designing molecules by the use of inferences from QSAR models is not new, but the complexity of many problems addressed by QSAR models today renders highly complex models where simple interpretations are often rare. When a QSAR model gives an undesired prediction it is a signal to the chemist that the compound needs to be modified to become a potential drug. This work traditionally consists of database and literature searches which together with inferences from the QSAR model aid the chemist in finding novel promising modifications to the compound. The approaches used leave a difficult task to the chemists in finding new substituents that will result in more favorable properties. Resulting in a very time consuming procedure that is highly dependent on the skill and expertise of the chemists.

The IQSAR approach is intended to replace large parts of the work that chemists do when searching and enumerating new molecules and fragments.

1.2.3. Traditional QSAR Modeling Approaches

In the literature two distinctly different QSAR modeling strategies have been applied, commonly denoted “Local” and “Global” QSAR models. For example, Guha, *et al.*¹⁶ defines the global model as the model built on the entire training data and that there may be groups of molecules in the data set that exhibits specific sets of features that relates to the activity or inactivity of the compounds. Such a set should in that case represent a “Local” structure activity relationship. This local set is suggested to be extracted by fingerprint or descriptor similarity. Zhang, *et al.*¹⁷ use the Euclidean norm in descriptor space to determine which compounds are chemically similar, and thereby “Local”. The assumption that molecules that are close in descriptor space or fingerprint space will tend to have the same properties has been studied by Martin, *et al.*¹⁸ They try to relate fingerprint similarity to biological activity, but find no clear connection. Boström, *et al.*¹⁹ have made a pairwise comparison of compounds binding to the same biological target where all pairs have a Tanimoto similarity of at least 80%. They conclude that the binding mode is preserved but the shape and water architecture of the binding site can be significantly different, mainly due to side-chain movements, resulting in unexpected activity changes in QSAR models.

When generating QSAR models on a subset of the available data compounds or examples are left out which means that information is also left out. This raises three important questions. Can one actually gain accuracy by doing this? Are there any risks or drawbacks with this kind of removal of information? Is important information left out? In the literature there are examples of QSAR models based on subsets of the data²⁰ as well as all available data²¹ which give good accuracy.

1.3. Machine Learning

Machine learning is the science that focuses on making machines able to learn. The field evolved from the broader artificial intelligence field which aims to mimic intelligent abilities of humans by machines. Learning in this context is restricted to inductive inference where data is used to build knowledge that is later used to predict new data. Machine learning can be divided into two major categories, supervised and unsupervised learning. Unsupervised learning tries to find regularities or irregularities in the data whereas supervised learning uses data coupled to a known response, which should be an answer to a question regarding the data, reported for each point in the data, denoted example. If the response is discrete the task is a classification problem while if the response is continuous it is a regression problem.

In this work supervised learning has been used. The goal of supervised learning is to approximate the function that maps the properties, descriptors, of the examples with a response, activity. The mapping is constructed using training data and can be tested on a validation set or

using cross validation. A validation set is a part of the data set withheld during training of the model. The cross validation approach splits the data set in n subsets and for each subset a model is built using the remains of the data and tested on the subset. There are also other measures to assess for example model errors and if a model generalizes. These methods can be algorithm specific, like the out of bag error for Random Forest (RF)²² and the number of support vectors for Support Vector Machines (SVM).²³

Different machine-learning methods have different ways of deriving these approximations. More detailed descriptions of the methods used in this work can be found in the following sections and in the references for Partial Least Squares (PLS),²⁴ RF²⁵ and in SVM.²⁶

1.3.1. Random Forests

Random Forest (RF) is an ensemble classifier, which means that it builds ensembles or sets of classifiers which used together become more accurate than a single classifier. An RF consists of a set of decision trees which all cast a vote that is weighted and added to the final prediction. The algorithm for inducing an RF was developed by Leo Breiman and Adele Cutler.²² The term originates from random decision forests introduced by Tin Kam Ho.²⁷ The method combines “bagging”, **bootstrap aggregating**, as described by Breiman²⁸ and random selection of features, introduced independently by Ho^{27,29} and Amit and Geman³⁰ to construct a collection of decision trees where variation is controlled.

The bagging approach takes a training set, J with n examples and generates m new training sets J_i of size $n_i \leq n$ by sampling uniformly from J with replacement, the remaining examples are used to calculate an error estimate of the J_i th training set.

In an RF each J_i training set is used to construct a decision tree. For each node in the tree, a small subset of descriptors is chosen at random, with replacement, from the complete set of descriptors. The best split is calculated and the data is separated with respect to the split. The tree is built from the root up adding nodes until all examples have been separated, *i.e.* the tree is not pruned.

This generates m trees and thus m models which are combined to a single predictor, the RF model. The way that the models are combined depends on the type of response, for regression the output is averaged and for classification voting is used.

Breiman²² prove two important properties of RF. Using the strong law of large numbers, Breiman shows that with increasing number of decision trees the generalization error for RF almost surely converges which means that the RF algorithm does not over-fit data with respect to the number of trees used. Secondly Breiman obtained an upper bound on the generalization error in terms of the strength of the classifiers and correlation between them in terms of the raw margin function.

Since RF is a tree based ensemble method where conditions are imposed on the descriptors

at each individual node it will take discrete steps in the descriptor space so the model function will be comprised of piece-wise constant functions.

1.3.2. Support Vector Machines

Support Vector Machines (SVM)^{31,32} have their theoretical foundation in Statistical Learning Theory provided by Vapnik.²³ The work of Vapnik provides conditions and proofs for good generalization of learning algorithms. Large margin classification and regression techniques have emerged from the theory of generalization and works by maximizing the margin, *i.e.* optimizes the location of the decision boundary so that examples end up on the correct side with as large margin as possible. This results in a decision boundary with large margins to almost all training examples. The most widely studied class of large margin classifiers are SVM.

SVM have an interpretation as a hyperplane separation in a high dimensional feature space²³ and maps the training data using a kernel function and to achieve the separation. The kernel computes similarities for all examples. Most commonly used kernel functions are Radial Basis Function (RBF) kernels and polynomial kernels. Training examples and previously unseen examples are assumed to be close to the training examples, independently identically distributed. Hence, the large margin then ensures that these examples are correctly classified as well, *i.e.* the decision rule generalizes. The kernel function needs to be positive definite assuring that the optimization problem can be solved efficiently.

Support Vector Machines are based on a substantial amount of statistical learning theory. Conditions for the kernel, both kernel function and kernel applicability, are supplied by Mercer's theorem³³ where a symmetric positive definite function is represented as a sum of a convergent sequence of product functions. Furthermore Karush, Kuhn and Tucker^{34,35} stated conditions that must be fulfilled for a solution to be an optimal solution. These conditions are necessary but not sufficient, *i.e.* the solution can be locally optimal, but the conditions on the kernel from Mercer's theorem result in a convex optimization problem, hence it has no local optimum. The above states that if the conditions are fulfilled there exist an optimal solution to the problem. To apply this using a learning algorithm Valiant³⁶ introduced a theory of probably approximately correct learning. The goal of Valiant's theory was that for an arbitrary distribution the probability that a learning algorithm would select a decision function with a low generalization error, approximately correct, should be high. Based on the above work Vapnik and Chervonenkis gave necessary and sufficient conditions for consistency of a learning process using risk minimization.³⁷

1.3.3. Partial Least Squares

Partial Least Squares (PLS), was developed by Herman Wold^{38,39} for econometrics but first gained popularity in chemometric research and later in industrial applications. It was designed

primarily to deal with data sets with missing values and more descriptors than examples. When y is a column vector with the corresponding row vector \mathbf{x}_i in a matrix \mathbf{X} where the length of \mathbf{x}_i is at maximum the length of y , modeling can be accomplished using ordinary multiple regression. When the number of descriptors is large compared to the number of examples, the covariance matrix is likely to be singular and the regression approach is no longer possible. Unlike principal component analysis that is based on the eigenvectors of the covariance matrix of \mathbf{X} . PLS, however, finds principal components from \mathbf{X} that are correlated with y . This means that PLS searches for a set of components that performs a simultaneous decomposition of \mathbf{X} and y with the constraint that these components explain as much as possible of the covariance between \mathbf{X} and y . This tends to greatly reduce the number of descriptive variables used for the actual regression problem and will reduce co-linearity and select descriptors that are linearly correlated with the responses.

1.3.4. Descriptor Importance Measures

One of the most important aspects of a model, besides that it should be predictive, is that of model interpretation. For linear models that is fairly simple, since it is possible to look at the contribution to the model from any descriptor. The descriptor with the highest absolute valued coefficient will be the most influential descriptor, thus a change in the property described by the descriptor is likely to give the largest change in outcome. This is however not true for modeling techniques that handle non linearity.

There is work presented where modelers have tried to derive general rules and importance measures for non-linear models based on all the data. Franke, *et al.*⁴⁰ computes gradients once for each variable and then the contributions are added to each molecule to achieve the globally most important variables. Guha, *et al.*¹⁶ divide the global space modeled into subspaces and uses linear regression to model these smaller sets of data. They then discuss the issue of interpretability for the data as a whole. These methods implicitly assumes that the most important descriptor in a specific point or subspace of the complete model space will be the most important descriptor everywhere. This kind of assumption reduces the non-linear model back to a linear one, and it is likely that most of the inferences made with this reduced set of rules are less accurate if the data actually contains non-linear features. An example of a global descriptor importance measure for a non-linear machine-learning algorithm is the function importance in RF, R.⁴¹

The contribution here is a way of local interpretation of non-linear models, where the importance measure is isolated for each prediction in the data, resulting in a local guidance, allowing the chemist to improve activity for that particular compound.

1.4. Simulations - a Way to Deeper Understanding

When doing computer based modeling it is of the highest importance to test methods and implementations. This tells the modeler how the algorithms work and if they contain errors. This can of course to some extent be accomplished using real data, but the difficulties with real data is that it is not completely controlled. There are inevitably errors in real data and the underlying relationship is not fully known. To test new algorithms and implementations it is therefore good to use simulated data, where all parameters can be controlled. This strategy of simulating data has been successfully applied in other fields and introduced as twilight zone simulations⁴² or the method of manufactured solutions. It is a technique where a predefined solution to the problem is used. This can be applied to QSAR modeling by drawing descriptors from statistical distributions and deciding on a mathematical function, based on the descriptors, that is to be the response. In this way the exact relationship between the descriptors and the response is known. The concept involves looking at the real problem at hand, then trying to construct an example that is as simple as possible yet retaining the difficulties of the real data. For example, take a classification problem in drug discovery where there is a binary response and a set of descriptors computed from the structure of the compounds. The task is to classify the data. First, look at the descriptors and try to approximate them with statistical distributions. To make it simple, reduce the dimensionality of the problem and only use a small set of descriptors, based on the obtained distributions. This results in half a data set, the response is needed as well and therefore a function is decided upon that generates the response based on the simulated descriptors. At this stage everything has been controlled; the descriptors, the response function that the method should approximate and the response for the constructed examples. The results from this case represent the ideal conditions and to obtain realistic conditions it is possible to introduce errors in the descriptors, in the response and in the mapping between the descriptors and the response. All these properties are now controlled by the modeler and he or she can study both behavior and properties of the algorithm and implementation of interest resulting in an increased knowledge of how the methods and the specific software work and at the same time it can be used to find implementation errors. When the method has been tested and qualified using simulations it can be trusted with real data.

1.4.1. Example

To show how the method of manufactured solutions can be used and at the same time show some properties of the machine-learning algorithms presented earlier, the following example has been chosen. Here the descriptor space is two dimensional and spans a range from $-\pi$ to π , in each direction respectively. The data points in descriptor space are uniformly distributed and there are 200 points for each descriptor. A training set with 800 points has been drawn randomly without replacement. The function defining the relationship between the response, y ,

and the descriptors, x_1 and x_2 , for *Data set I* is $f_I = \cos x_2 / (1 + x_1^2)$, Figure 1(a). The response, $y = f_I(x_1, x_2)$ was computed for every data point. Figure 1(a) shows the original function and Figures 1(b)-1(d) show how the different machine-learning algorithms, trained using the training set, predict the function f_I over the complete descriptor space. It is thus possible to see that the RF is built up using piecewise constant functions and that the SVM with an RBF kernel results in a smooth decision function. It is also clear that PLS, being a linear method, can not describe the non-linear relationship. Deriving knowledge, based on only one observation, is not

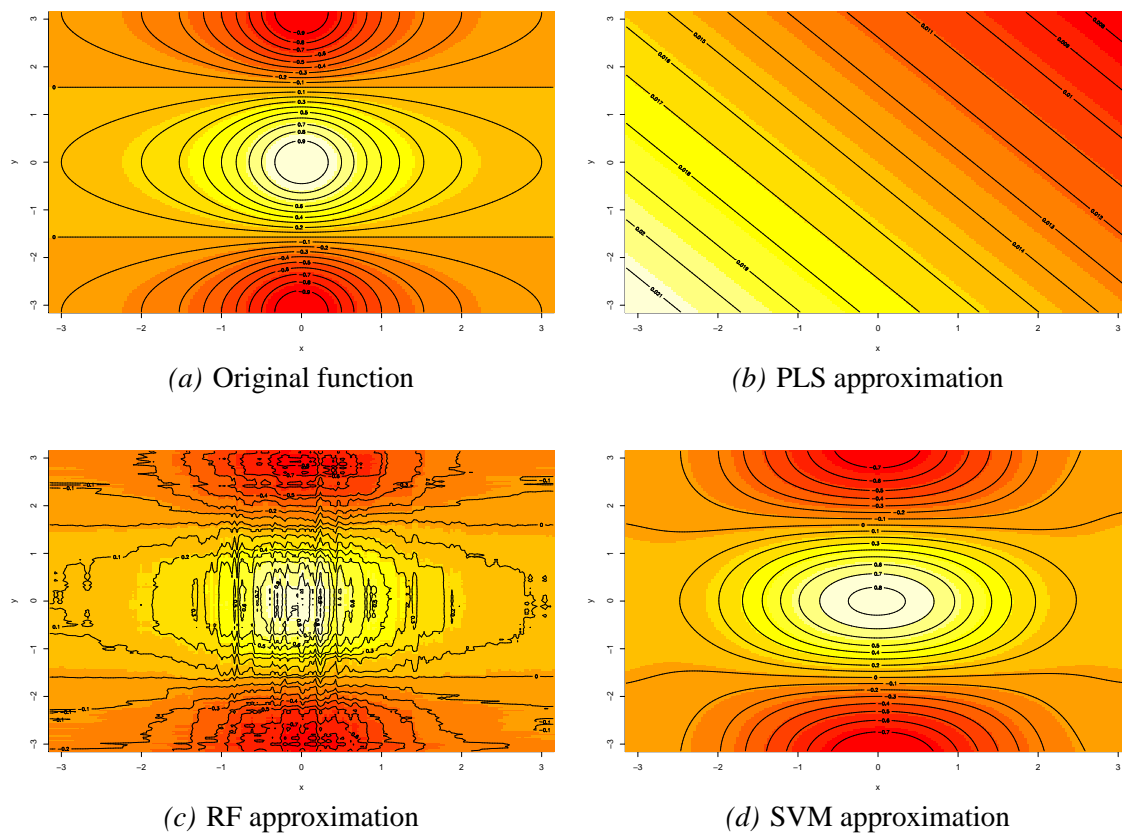


Figure 1: Contour plots of the function for *Data set I*

sufficient. If however this is performed multiple times with a range of training set sizes and a range of seeds* generating consistent results.

*Seed is a number used to initialize a pseudo random number generator in order to produce the same sequence of random numbers each time it is used.

1.5. Graph Theory

The description of compounds as graphs is important in this work. Therefore a general description of graphs and related concepts used will be presented here.^{43,44}

In mathematics a graph is a finite set of points called nodes connected by links called edges. This corresponds to the atoms and bonds in chemical structures, but to describe chemical structures as graphs different labels on the nodes and edges are needed, referred to as coloring. The normal representation of a compound is a simple colored graph, *i.e.* it contains colored nodes and edges, it can contain cycles but no loops. A loop is when a node connects to itself using an edge. The nodes can have different number of edges, a node with three edges has degree or valence three and cannot have more edges connected to it. If the node is fully connected it is saturated. In chemical structures all nodes need to be saturated in order to form a compound.

So far the graph has been used to describe the whole compound but graphs can also describe parts of compounds and such graphs are called subgraphs. In graph theory it is possible to have directional edges, which can only be traversed in one direction. In this work that concept is used together with trees. A tree is a graph that contains no cycles. When chemical structures are represented with trees the cycles in the graph must be opened up, which means that one node can be represented more than once. The directed tree structure is useful for making subgraphs comparable. To compare two graphs where the nodes and edges are enumerated differently a representation that is independent of the enumeration is needed. Such a representation is called canonical and the problem of comparing graphs to decide if they are identical is referred to as the graph isomorphism problem.

1.6. Molecular Representation

There are a number of formats for molecular representation, many of which capture the chemical structure through a graph. This is the basis of the chemical representation from which it is possible to compute molecular properties for the molecule as a whole and for its substructures.

To translate the graph theory definitions into its chemical counterpart is simple. The nodes from graph theory correspond to the atoms and the edges correspond to the bonds. If larger compounds such as proteins are represented it is common to define amino-acids as nodes to simplify the format. From the graph based approach simplifications have been made that allows compounds to be described using short ASCII strings for use in spreadsheets. One commonly used example of this is Simplified Molecular Input Line Entry Specification (SMILES) and was developed by Dave Weininger at Daylight.^{45,46} The SMILES is a string obtained by writing the atom labels of the nodes encountered in a depth-first search of a graph representation of a compound. All hydrogen atoms are removed and cycles are broken up to turn the compound into a directed acyclic graph. The atom pairs where cycles have been broken are given a numeric

suffix to allow for reconnection. Parenthesis are used to indicate branching. Since the atoms in a compound are connected by different kinds of bonds the single bond is omitted and the double and triple bonds are expressed by = and # respectively. Aromaticity is written using small letters for the atoms. All atoms represented by two syllables are written within a bracket. Stereo chemistry over a double bond is indicated using / and \. For tetrahedral carbons @ or @@ is added to account for R and S enantiomers. The way this is implemented in the SMILES language is however not R and S but rather the clockwise and anti clockwise positioning of the structures attached to the chiral carbon as they are written in the SMILES. For example OC[C@H](CC)NC is the same as OC[C@@H](NC)CC.

The above describes the whole compound, but for modeling purposes substructures or properties of compounds are often used. Substructures can be encoded using SMiles ARbitrary Target Specification (SMARTS)⁴⁷ which is similar to SMILES but offers a wider range of node labels for finding substructures that are similar but not exactly identical. For example an aromatic carbon would be represented as c, to match any aromatic carbon or nitrogen the SMARTS could be [c,n]. If any aromatic atom should match the SMARTS could be just an a. In short SMILES are used to describe compounds and SMARTS are used to search for substructures within compounds.

1.6.1. Signatures

For this work a central representation is the signature descriptor developed by Faulon, *et al.*⁴⁸ The signature of an atom is a canonical representation of the environment surrounding the atom. The signatures can be calculated for different heights which corresponds to how far away the environment to the atom is defined in the signature. At height zero only the atom itself is considered. For height one the signatures contains the information from the current atom to its nearest neighbors, including the connecting bonds.

The signature of an arbitrary atom is a tree representation of a subgraph to the graph of the molecule, such that all neighbor atoms up to a specified distance, height, from the atom are taken into account. The tree is represented with a string written in depth-first order. The atom types are given within square brackets and a step away from the parent is indicated with an ordinary bracket. The signature for an atom bound to a neighbor atom will then look like this: [atom_type](bond_type[neighbor-atom_type]), see Figure 2. With the tree representation this means that the layer underneath an atom is composed of the neighbors of that atom, the second layer is composed of the neighbors neighbors except the atom itself.

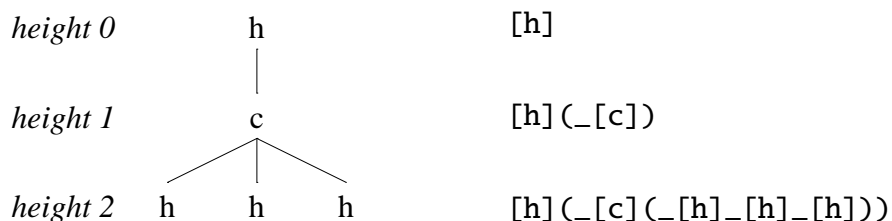


Figure 2: A tree representation of methane and the atom signatures of the hydrogen for the heights 0 to 2.

1.7. Enumeration of New Compounds from a Set of Signatures

A method to generate new compounds has been described by Visco, *et al.*¹³ and Churchwell, *et al.*⁴⁹ where compounds are decomposed into building blocks represented as signatures. These building blocks define a space in which all possible compounds are built.

The method starts off with a set of compounds and the corresponding signatures. Using the signatures of the compounds, connectivity constraints, created by comparing parts of the signatures, are set up that govern how the signatures can be combined to form new compounds. The constraints form a system of linear equations.

As stated above the signatures describe a center atom, its n layers of surrounding atoms and the bond types connecting the atoms. By looking at the environment around the center atom it is possible to see what the surroundings of another atom must be in its $n - 1$ layers to connect to the center atom. For each signature, in the set of height n signatures spanning the space, the height $n - 1$ signature, ${}^{n-1}\tau$, is computed along with the height $n - 1$ signatures for the first layer neighbors, ${}^{n-1}\sigma$. To form a bond between two atoms i, j described by the signatures ${}^n\tau_i$ and ${}^n\tau_j$, at least one of the ${}^{n-1}\sigma_{\tau_j}$ must match the ${}^{n-1}\tau_i$ and vice versa. In this comparison a direction is imposed on the bond $i \rightarrow j$. For each such pair of height $n - 1$ signatures an equation is set up such that for each ${}^n\tau$ the number of possible connection pairs is counted and added as a coefficient to the equation, see Figure 3. The sign of the coefficient depends on which signature is searched first. If no equation comparing ${}^{n-1}\tau_i \rightarrow {}^{n-1}\tau_j$ or ${}^{n-1}\tau_j \rightarrow {}^{n-1}\tau_i$ exists, then ${}^{n-1}\tau_i$ will get a positive coefficient and the ${}^{n-1}\tau_j$ will get a negative coefficient. If the height $n - 1$ signatures for i and j are identical no direction can be imposed on the comparison and a dummy variable needs to be added to balance the equation.

Figure 3 contains a visual example of the process described above. The first sub-figure, 3(a), contains one compound and all atom signatures of height 1. Each signature has a colored center atom and the light blue dots represent the surrounding atoms for each signature. There are five different signature types of height 1 in this compound, marked with brown, green, dark blue, yellow and red. The interconnectivity among the signatures is described in Figure 3(b).

For example the brown colored signature matches the green colored one since the center atom of the green signature is represented in the brown colored height 1 signature as a neighboring atom, light blue. When a pair of signatures matches it means that they can form a bond so if a compound in the example has a brown signature, it must also have a green one. This knowledge can be transformed to mathematical constraints that governs how the signatures can be combined, see Figure 3(c).

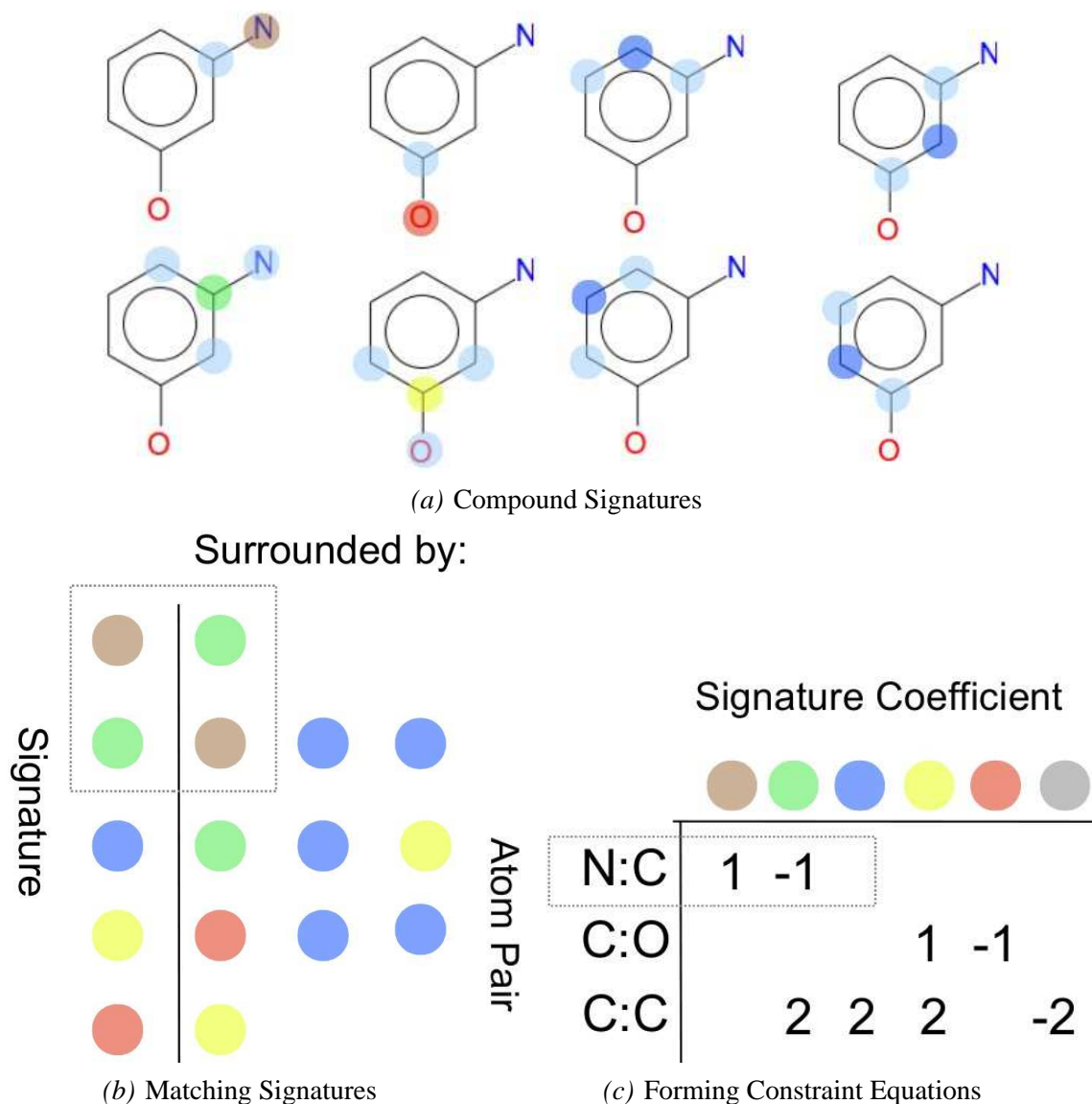


Figure 3: A visual representation of what the signatures represent, how they can be combined and how constraints are formed.

In the cases where the connection environments are identical for both atoms a dummy variable, gray labeled in the example, has to be added to balance the equation. The coefficients of these equations form a constraints matrix that defines how new compounds can be built. Since the representation of signatures in compounds is an enumeration of atom types and their neighboring environment, solutions to the system of equations must be vectors with non-negative components, *i.e.* the solutions are Diophantine solutions. To solve the system of equations a Diophantine equation solver algorithm developed by Devie, *et al.*⁵⁰ has been used. This algorithm does a stack based search and retrieves the complete set of minimal solutions, where a minimal solution is a solution which can not be obtained by combining other solutions, using integer multiples. The algorithm starts from the origin and moves stepwise in descriptor space. The vector of a valid step is pushed on to a stack and each new step starts with a pop, taking the vector from the top of the stack. For each pop the algorithm evaluates possible steps in descriptor space and pushes the vectors for the steps that were allowed. A step in a descriptor direction is only allowed if it represents a move closer towards the origin in constraint space. A minimal solution to the system of equations is found when a step reaches the origin in constraint space. The stack based version of this algorithm prevents the search from finding the same minimal solution many times by blocking descriptor directions in a way so that a particular subspace will only be searched once. The solutions to the system of equations in Figure 3(c) are presented in Table 1.







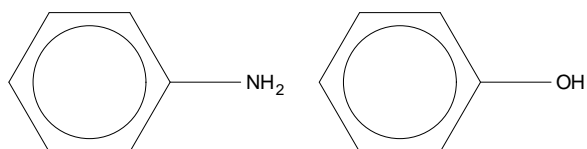
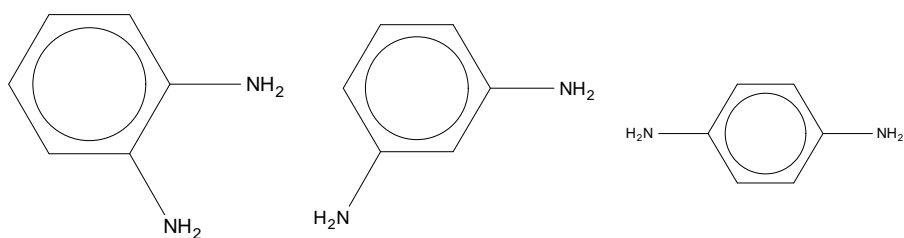
						
<i>a</i>	1	1				1
<i>b</i>				1	1	1
<i>c</i>			1			1

Table 1: Solutions to the system of equations in Figure 3(c).

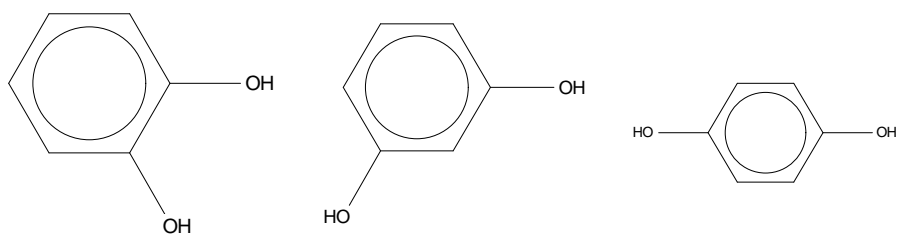


(a) $a + 5 * c$

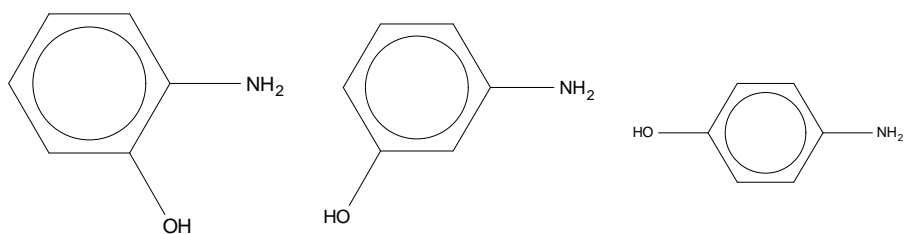
(b) $b + 5 * c$



(c) $2 * a + 4 * c$



(d) $2 * b + 4 * c$



(e) $a + b + 4 * c$

Figure 4: Compounds based on linear combinations of the solutions in Table 1.

Linear combinations of the solutions with minimal support are made to find all solutions in the subspace. A compound can be viewed as a connected graph where all vertices (atoms) are saturated and therefore resulting solutions must also fulfill a graphicality equation.¹⁵ The graphicality equation determines if a set of vertices can establish a connected graph and if so how many cycles it contains and can be derived using the following relations from graph theory. If a compound \mathcal{G} has n atoms and m bonds, then its cyclomatic number is

$$c = m - n + 1 \quad (1)$$

which is the number of independent cycles in \mathcal{G} . Let g_i be the number of atoms in \mathcal{G} with heavy atom valence $\vartheta = i$, then another way of counting the atoms is

$$n = \sum_{i=1}^{\vartheta_{max}} g_i \quad (2)$$

and the corresponding expression for the bonds is

$$2m = \sum_{i=1}^{\vartheta_{max}} i \cdot g_i. \quad (3)$$

By substituting Equation (2) and (3) into (1) the graphicality equation can be written as

$$\sum_{i=1}^{\vartheta_{max}} (i - 2)g_i + 2 = 2c. \quad (4)$$

All possible compounds are created from the signatures according to the solutions. This was accomplished using an algorithm proposed by Visco, *et al.*¹³ The algorithm recursively reassembles atoms from the signatures representing the solutions to form possible compounds and it only allows the canonical structures to be built and thus reduces the construction time. Some compounds assembled from the solutions in Table 1 are presented in Figure 4.

This method has a nice feature since it is deterministic, *i.e.* it is ensured that all compounds in the searched space are found. However, it is computationally costly and due to the complexity of the problem, slow for a complete regeneration of drug-like compounds. The number of published applications thus far has been quite limited and usually describe limitations where signatures represent larger parts of the compounds that has well known linkers like amino-acid chains and polymers.

1.8. Finding Substructural Alerts in Data

The drug discovery process is dependent on warning systems that use substructural alerts to notify chemists of potential risks. These systems include for example warnings for genotoxicity and mutagenicity.⁵¹⁻⁵³ The extraction of substructural alerts can however be accomplished without the use of commercial software.

Perhaps the most simple method to extract substructural alerts from data is to utilize chemical expert knowledge. A more advanced method is to manually cluster the data and to identify substructures by visual comparison. The extraction of substructural alerts using chemical expert knowledge or any other manual technique is time consuming and generates subjective substructures since it is dependent on the skill and expertise of the chemist. There are however computational methods that mines molecular substructures from data.⁵⁴

The best methods available today grow substructure graphs from the atom types by computing frequent cliques, where a clique is a set of pairwise adjacent vertices or an induced subgraph which is a complete graph.⁵⁵ The clique based techniques starts with the individual nodes in the graph and grows the substructures by combining nodes until no more substructures can be found that obey the user specified occurrence threshold. This is an exhaustive search of substructures in the data and is well suited for finding substructures, but it comes with a high computational effort.

There are also methods that utilize MCS computations but those are primarily designed to identify privileged structures, *i.e.* the scaffold from which compounds are built. In such cases MCS computations are applied after clustering of the compounds⁵⁶ and the substructures therefore describes chemical classes of compounds in the data.

In this thesis an approach to mine chemical data for substructures that can separate the data is presented. The method is faster than existing methods and generates fewer substructures yet retains the predictive properties.

2. Contribution to the Field

2.1. Model Interpretability

One goal has been to investigate the interpretability of predictive QSAR models. As described in Section 1.3.4 many attempts have been made to find the most important descriptors for a data set as a whole or in, by chemists predefined, subsets within that data set. There has however been little work on describing the local space around the compound of interest with respect to the model function. The local behavior is of high interest to the chemists since, if a compound is considered to be active against a primary target but needs to gain specificity to reduce side effects, one would like to make small changes to the existing compound to optimize its properties instead of finding a completely new compound. To facilitate these changes one needs to find the property or descriptor for which a small value change would result in the largest change in activity. For this purpose, as shown in Paper I, the gradient of the model function can be used.

As stated in Section 1.3 not all machine-learning algorithms have simple analytic expressions for the model function that allows for analytical derivation. The RF method generates a model function that is composed of many piecewise constant functions, such a function has no simple analytical gradient but a discrete gradient can be computed instead.

Inferences from gradient computations can be used to rank the descriptors in order of importance with respect to a specific prediction. In this work gradients have been used together with the signature descriptor described in Section 1.6.1. The use of signatures or other substructural descriptors like SMARTS, see Section 1.6, have the advantage of being easy to understand since the substructure can be mapped back onto the compound, see Figure 5. This enables a direct coupling between the descriptor and the compound and this visualization facilitates the interaction between the modeler and the synthetic chemist.

2.1.1. Theory

If the QSAR model is viewed as a function, then at any point in a function the local behavior can be approximated using its Taylor series, which is an infinite sum of the derivatives of the function in that point.

$$f(a) + \frac{f'(a)}{1!}(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \dots \quad (5)$$

If an infinite number of terms is used the function can be completely described, but in this work the Taylor series has been truncated after the second term, such that only the prediction

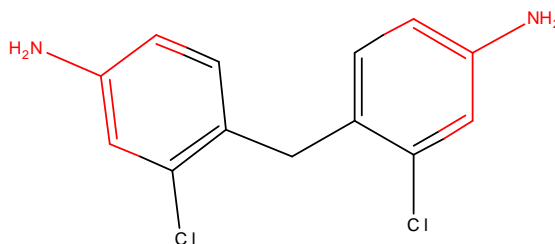


Figure 5: A compound with displayed substructures represented as signatures and SMARTS. The signature [c](p[c]p[c]_[n]) or SMARTS Nc(c)c is displayed in red.

and the gradient of the function is used to describe the local neighborhood. Gradients can be computed for any sufficiently smooth function and the gradient of a QSAR function is the partial derivatives of the function with respect to each descriptor.

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right) \quad (6)$$

By looking at the magnitude of the partial derivatives the descriptors that influence the local neighborhood the most can be found. In the work of Paper I only the descriptor corresponding to the largest component of the gradient has been considered.

Support Vector Machines

For SVM an analytical expression of the gradient can be obtained. The SVM decision function is a sum of weights times the kernel function where the weights are constants. So deriving the gradient of the decision function for SVM amounts to computing the partial derivatives of the kernel function.

Random Forest

In the case of RF models, in general, there is no easy way of obtaining an analytical expression of the model function. Instead, one can compute the j th component of the discrete gradient,

$$\frac{Df}{Dx_j} = \frac{\beta_1 f'(\mathbf{x} + h_j) + \beta_2 f'(\mathbf{x}) + \beta_1 f'(\mathbf{x} - h_j)}{2\beta_1 + \beta_2}, \quad (7)$$

where β_1 and β_2 are smoothing coefficients. The step size in the j -direction in attribute space is h_j and the corresponding second-order accurate partial derivative is $f'_j = (f(\mathbf{x} + h_j) - f(\mathbf{x} - h_j))/2h_j$.

Partial Least Squares

PLS, and any other linear model ($f_{PLS} = k_1x_1 + k_2x_2 + .. + k_nx_n$) has a trivial gradient, being the constant k_i for each descriptor in the model, and as such the gradient will be constant over the complete space spanned by the descriptors.

The above work corresponds to Paper **I**. With this method it is possible to interpret any non-linear QSAR model and by doing so chemists can be guided on what to change and how that change is believed to affect the compound. The method, does not give an answer to how this change should be facilitated or what the substructure can be replaced with. To solve this problem the idea have been to combine this method of knowledge extraction with an molecular enumeration algorithm which is the objective of Paper **II** described in the following section.

2.2. Automated Compound Optimization

Today tedious literature and database searches are made by chemists to optimize a compound with an undesired predicted or known biological activity. Most QSAR models reveal only the prediction but can also, if used as described in Section 2.1 indicate what needs to be changed. The model can however not indicate how to do the change or give suggestions of more optimal compounds.

This approach makes use of the data behind the QSAR model and the QSAR model itself. It takes a compound with an undesired prediction and isolates the substructure corresponding to the largest gradient in the QSAR model. To replace the substructure a set of compounds is needed. For this reason the QSAR training set has to be searched for compounds similar to the substructure of interest. The set of similar compounds together with the substructure, represented as a compound, can be used to form constraints describing the interconnectivity between all atoms, described as signatures, in these compounds. With the set of similar compounds at hand, the procedure for generating compounds described in Section 1.7 was used, and new substructures were generated. The substructures were then combined with the remains of the query compound, if possible, and predicted with the QSAR model. Once this process is completed one has a set of deterministically built substructures that can be used to replace the active substructure. After replacement all new compounds were predicted with the QSAR and presented to the chemist. This provides the chemist with ways to optimize the compound and learn more about the local properties around the compound of interest.

2.2.1. Theory

In the process outlined above, and described in detail in this section, the manual searches together with replacement structure generation are automated. The different steps in this method are visualized in Figure 6 where the starting point is a query compound which needs to be predicted for a biological activity or a set of activities using multiple QSAR models.

The descriptors used in the QSAR models were signatures^{48,57} and the model function can be generated using any machine-learning method, for example RF²⁵ or SVM.⁵⁸ The local interpretation of machine-learning models described in Paper I and in Section 2.1 can be used to extract the signature with the most significant contribution to the QSAR prediction of the compound. This most significant signature corresponds to positions in the compound where changes possibly need to be made to get a different prediction from the the QSAR model. The following procedure was only performed for compounds that receive unfavorable predictions.

From the significant signature located by the QSAR model a substructure based on the signature had to be cut out from the compound. This substructure was generated by cutting bonds from the atoms at a specified distance from the center atom of the signature. If an atom at this distance belonged to a ring the search was extended to embed the ring. Each atom for which a bond was cut has been kept as an anchor atom and for each such atom a SMARTS⁴⁷ was generated that described the atoms around the bond that was cut. To recombine generated substructures and the original end groups this SMARTS must match. If the query compound could not be cut, it was treated as a substructure throughout the remainder of this algorithm. However, it did not go through the recombination step where SMARTS have been used.

At this point the substructure that needs to be replaced was retrieved. To setup constraints for the Diophantine equation solver a subspace around the substructure was spanned using neighbors to the retrieved substructure. The neighbors, were located based on similarity. The near neighbor search was conducted in a database of compounds for which measured activity was available for the specific endpoints and in particular the endpoint that the QSAR model approximates. From these neighbor compounds a set was chosen that covered a range in activity around the query compound.

A method to generate new compounds has been described by Visco, *et al.*¹³ and Churchwell, *et al.*⁴⁹ This Algorithm was briefly described in Section 1.7.

The implementation used in this work differs slightly from that used by Churchwell, *et al.*⁴⁹ Most of the changes have been applied to constrain the size of the new substructures⁵⁹ and to reduce the computational time. These restrictions have been imposed mainly on the solutions to the Diophantine equation solver. The first restriction blocks steps in an attribute direction once it has reached a given threshold. Another restriction was imposed to avoid the computation of solutions where the sum of signatures exceeds a predefined threshold. When linear combina-

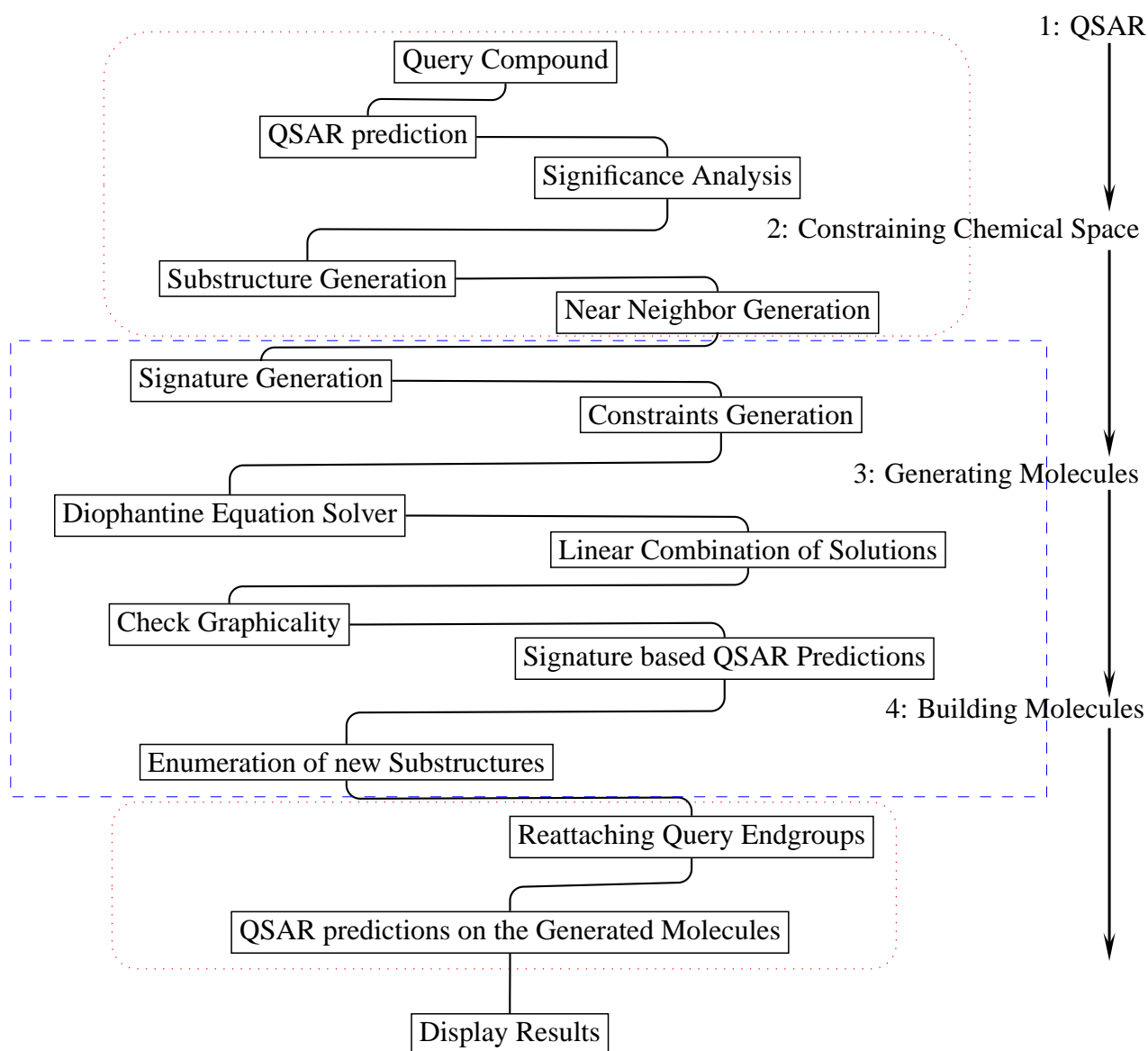


Figure 6: Flowchart displaying the different steps of the work flow, where the blue dashed box indicates the work of Visco *et al*¹³ and Churchwell *et al*⁴⁹ described in Section 1.7 and the red dotted boxes indicate the part of the method described here.

tions of the solutions were made, an upper and lower bound was set on the size of a solution to ensure that the new substructures had a similar size to the original substructure and to reduce computational time.

At this point all possible combinations of signatures that may result in potential substructures have been computed. Here it can be useful to impose more restrictions on the solutions. If QSAR models have been built using signatures of height h or less then the solutions representing non-built substructures can be used to predict the properties of the new substructures even before they are built.

In a recursive procedure, all possible substructures were built from the signatures corresponding to each solution, respectively. This was accomplished using an implementation of the algorithm proposed by Visco, *et al.*¹³ The algorithm recursively reassembled atoms from the solutions to form possible substructures and it only allowed the canonical structures to be built and thus reduced the construction time by not building duplicate structures.

Since the Diophantine equation solver is deterministic it ensures that all possible substructures within the subspace defined by the constraints matrix were found. The substructures built were preprocessed in the same way as the query compound and additional filters were applied to omit structures with specific ring sizes. Different filters can be applied based on the specific problem at hand. If the query compound had anchor atoms, SMARTS were generated that described the anchor atoms and their required neighbors in the substructure. The end groups were attached to the generated substructures if the SMARTS patterns from the end groups could be matched onto the substructure, it was then a complete compound. In the case where an end group could be attached to several points in the substructure, all permutations were assembled. The generated compounds were preprocessed in the same way as the query compound and any duplicates were removed. If the query compound could not be partitioned into a substructure and its corresponding end groups, substructures from the above step were the complete new compounds. A final filtering step can be applied to remove compounds with undesired properties. Examples of such filters are drug likeness, ring compositions and molecular weight. For the new compounds, QSAR predictions were obtained for the different biological endpoints of interest.

2.3. Finding Significant Substructures

The aim of this method is to automatically generate substructural alerts based on chemical data in an objective way. It is of highest importance to reduce the risk of subjectiveness when generating these substructures since the outcome of the algorithm should represent the data modeled and not be a matter of choice. The method should preferably generate substructures that: can create high-performing models, are easy to understand and visualize.

The method presented in Paper **III** mines chemical data using signatures and significance testing. The signatures used in this work are similar to those presented by Faulon, *et al.*^{48,57} Each node in the signature tree can be arbitrarily labeled but in this case the Sybyl atom-type⁶⁰ corresponding to the atom has been used.

For a substructure to be statistically significant at a certain level, it was required that the p -value was below this level. Furthermore the p -value alone is not sufficient for a substructure to accurately describe an activity. With increasing number of occurrences of a specific substructure in the data the accuracy for a given p -value will decrease. In addition to the level of significance a lowest level of classification accuracy needs to be imposed on the substructures.

For classification, where substructures are used as indicators for compounds belonging to a certain class, the class representation follows a binomial distribution. An outcome for a specific substructure can be said to be the occurrence of itself in a number of compounds with a certain activity and the occurrence in a number of compounds without that activity. To see whether such an outcome is likely to come from a specific binomial distribution the p -value was used. The p -value is the probability of obtaining the outcome or any other less probable outcome. The outcome has to be related to the occurrence of the activity in the total number of compounds, *i.e.* even compounds where the substructure does not exist. For example, a data set with n compounds has m compounds with a specific activity label where $m \leq n$. In the data set a substructure is found in n' , $n' \leq n$ compounds and the amount of those compounds with the specific activity label is m' , $m' \leq m$. The accuracy for the substructure in the training data is $\frac{m'}{n'}$ and the p -value is $\sum_{i=m'}^{n'} \frac{n'!}{m'!(n'-m')!} * (\frac{m}{n})^{m'} * (1 - (\frac{m}{n}))^{(n'-m')}$.

For data where a specific activity was overrepresented it was possible to obtain significant substructures with only one or two occurrences in the data. To avoid this a lower bound on the number of compounds a substructure exists in have to be used.

The algorithm takes a data set with a classifier response, thresholds for the p -value, substructure occurrence and the accuracy. In the initial step all height zero signatures were computed from the compounds in the data set and for each signature the total number of compounds it exists in was recorded together with the number of occurrences with respect to the activity of each compound. For each signature, if the number of occurrences was above the threshold the accuracy of the signature was computed for each activity compared to the all the other activities. If the accuracy for an activity was above the accuracy threshold the p -value was computed. If the p -value was below its threshold the signature was labeled significant in discriminating the activity. If the signature was significant, the search for significant substructures was terminated in that direction. For the signatures that passed the occurrence threshold the search was extended to the next height. The above procedure was repeated until no signature could fulfill the thresholds on accuracy, p -value and occurrence.

2.4. Evaluation of QSAR Modeling Strategies

Section 1.2.3 describes different QSAR modeling approaches used today. The questions asked there are the cornerstones for this work on QSAR modeling strategies.

There is a need to test and validate differences in *Local* and *Global* QSAR modeling strategies and how different numerical routines and modeling algorithms handle those differences. The aim of this work was to:

- gain knowledge about the expected predictive performance of *Local* and *Global* modeling strategies
- investigate possible risks in terms of the definition and usage of applicability domains for *Local* and *Global* modeling strategies

It was also interesting to see how the different machine-learning algorithms make use of the available information. To allow for a deeper understanding of the strategies and a significant amount of this work has been conducted on simulated data.

The information content in a QSAR model is defined by its response and the descriptors used. Depending on the information at hand different modeling strategies can be applied. In this work, *Local* and *Global* modeling strategies have been compared using two levels of information content, denoted *Ideal* and *Restricted*. For the *Ideal* case all relevant information to accurately describe the underlying relationship is contained by the descriptors, a *complete descriptor set*. For the *Restricted* case the descriptors are missing relevant information, an *incomplete descriptor set*, and cannot be used to describe the underlying relationship but merely an approximation to it. For these two levels either an entire data set, *Global*, or a subset of the data, *Local*, can be used. This defines a *Global* model as a model built using the entire data set, all available information. A *Local* model, on the other hand, is a model built for a specific example using neighbors from the entire data set. The definition of neighbors can vary, but in this work it was based on a descriptor or fingerprint similarity.

The *Global* modeling strategy has been applied with the two levels of information content, *Ideal Model Global*, *IMG*, and *Restricted Model Global*, *RMG*, as illustrated in Figure 7. *IMG* uses a complete descriptor set and *RMG* uses only a subset of these descriptors for model building. For each *Global* case two corresponding *Local* cases have been applied that locates near neighbors in a *Restricted* or an *Ideal* fashion. Following the structure in Figure 7, in the *IMG* branch the *Local* cases both use a complete descriptor set for building models and making predictions. The *Ideal Model Ideal Local*, *IMIL*, use the complete descriptor set to identify near neighbors but the *Ideal Model Restricted Local*, *IMRL*, only make use of a *Restricted* subset of descriptors for identifying near neighbors. In the *RMG* branch *Local* modeling cases both use an incomplete descriptor set for model building and predictions but *Restricted Model Ideal*

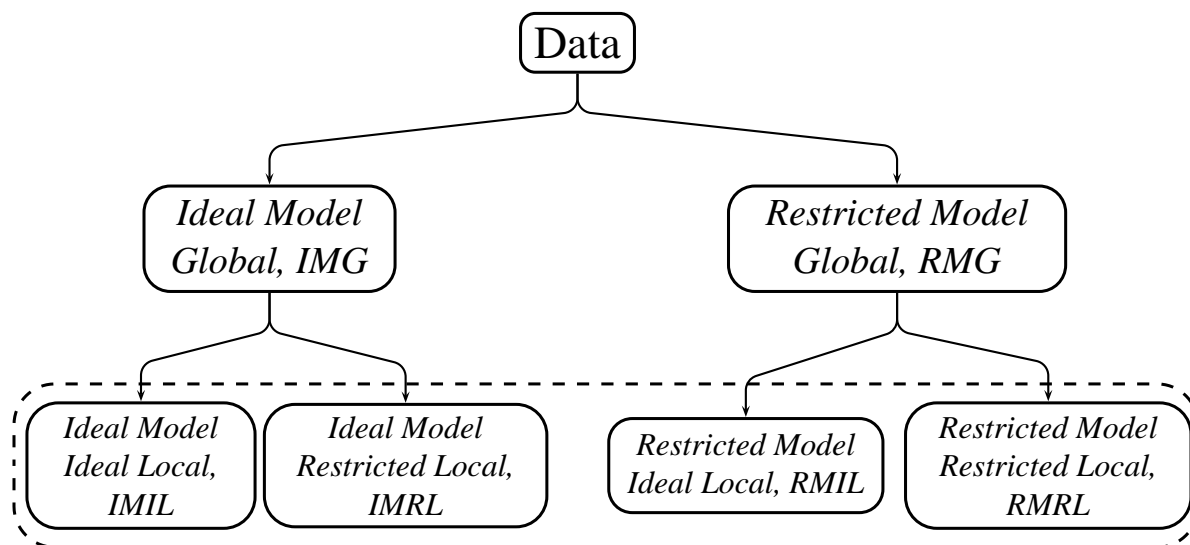


Figure 7: The different *Local* and *Global* strategies that have been applied to the data. The dashed box indicates the *Local* strategies.

Local, *RMIL*, use a complete descriptor set for finding near neighbors and *Restricted Model Restricted Local*, *RMRL*, use the *Restricted* subset of descriptors for identifying near neighbors.

The global and local modeling strategies above can be summarized as follows:

- *IMG* constructs *Global* models where all the information about the underlying relationship is known and expressed by the descriptors.
- *RMG* describes *Global* models which can not correctly describe the underlying relationship.
- The *RMRL* case results in *Local* models that make use of a neighborhood and a descriptor set that can not completely describe the problem at hand. This can be described as the normal case when building QSAR models since the underlying relationship can not be properly described but one makes use of all information at hand for finding the best possible models and near neighbors.
- In the *IMIL* case results in *Local* models where all the information about the underlying relationship is known. *IMIL* can be directly compared with the *RMRL* case where the difference is loss of information for *RMRL*.
- *RMIL* represents the *Local* model case where external information is added in the neighbor search, which can be relevant in describing the underlying relationship. The addition of this type of information can lead to a model that is truly *Local* with respect to the underlying relationship.
- The *IMRL* case describes *Local* models where the *Local* neighborhood is partially unaccounted for or cannot be correctly described, as opposed to *RMIL*. In fact the underlying relationship is properly described by the descriptors but the near neighbors has not been selected in accordance to the underlying relationship.

The different modeling strategies and risk assessments were evaluated using various machine-learning algorithms. To assess individual model performance a cross-validation approach was used, which is commonly used in literature.⁶¹ A data set was divided into n subsets by uniform sampling of examples without replacement. Each subset was treated as a test set with the remaining examples as the training set. For each test set an overall prediction metric was computed. If the response was binary this metric was defined as the prediction accuracy and if the response was real valued the root-mean square error was used instead. The prediction metric was averaged for all test sets.

The generation of *Global* models was straightforward; for each test set a model was built on the remaining examples of the data. *Local* models were generated for each example in a test set and for each such example near neighbors were retrieved from the remaining examples of the data, not included in the particular test set. Near neighbors were found by using different

similarity operators such as Euclidean norm on descriptors or Tanimoto distance on chemical fingerprints. The number of neighbors was either explicitly set or a cut-off value for the similarity was used. If a *Local* model could not be built under the specified similarity constraint the corresponding *Global* model was used to predict that compound. With the predictions from both *Local* and *Global* modeling strategies at hand it was possible to directly compare and assess prediction accuracies and errors.

To assess risks of *Local* versus *Global* modeling strategies, by comparing errors and accuracies for compounds within and outside of the domain of applicability, the domain of applicability had to be defined for the models. A *Global* model should be able to handle all data so the domain of applicability was defined to be the complete data set. For the *Local* models, by definition, an example for which a model was constructed by retrieval of near neighbors, was within the *applicability domain*. On the other hand if such a model was used for any other example it was used outside its domain of applicability.

3. Results and Discussion

3.1. Model Interpretability

Here the example from Section 1.4.1 has been used again but this time in an extended form.

The simulations have been implemented in R⁶² using the following packages for machine learning: `pls`⁶³ for PLS, `randomForest`⁴¹ for RF and `e1071`⁶⁴ for SVM. For each prediction the discrete gradient of the model function was computed and compared to the analytical gradient of the function. The models were trained using the default settings in the respective packages.

The descriptor space was the same as in the example in Section 1.4.1 and the function likewise. A test set of 100 data points has been drawn together with each training set consisting of 100, 200 400 and 800 data points. For each set 5 different seeds have been used, for full details see Paper I.

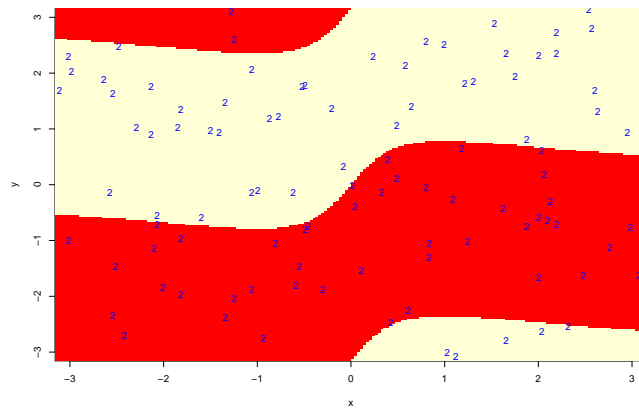
Examples of the most significant components for *Data set I* are presented in Figure 8. The figure shows the most significant component of the gradient plotted at each test point using the first seed and a training set size of 800 data points. The underlying contour plot represents the correct areas of significance, yellow indicating component 1 and red indicating component 2. For both RF and SVM the most significant component was correctly computed for almost all of the points in the test set, whereas PLS predicts the second component to be largest everywhere. This shows the usefulness of the gradient as a local importance measure for information retrieval from machine-learning models. In Paper I it has been shown that the inferences from the largest component of the gradient for QSAR models based on signatures were relevant for describing the data. This was demonstrated using AMES mutagenicity data and compared to the toxicophores reported by Kazius, *et al.*⁶⁵

3.2. Automated Compound Optimization

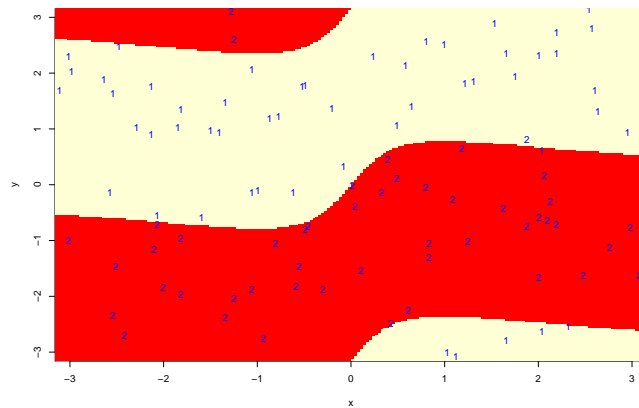
The work flow was demonstrated using AMES mutagenicity data from CCRIS⁶⁶ from which compounds and corresponding activity has been collected according to the conditions described by Kazius, *et al.*⁶⁵ including for example removal of organo-metals.

Figure 9(a) shows an example compound, the substructure that needed replacement, O=N-N, and the extended substructure that was replaced using the method. Figure 9(b) and 9(c) show examples of the generated compounds. For this particular case 2700 new compounds were generated, of which 800 were predicted to be positive and 1900 predicted negative, *i.e.* not mutagens.

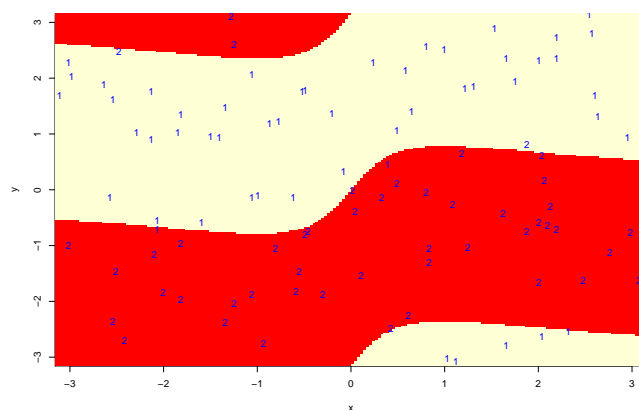
The method has been tested using 303 of the positive compounds from the AMES data set



(a) PLS approximation

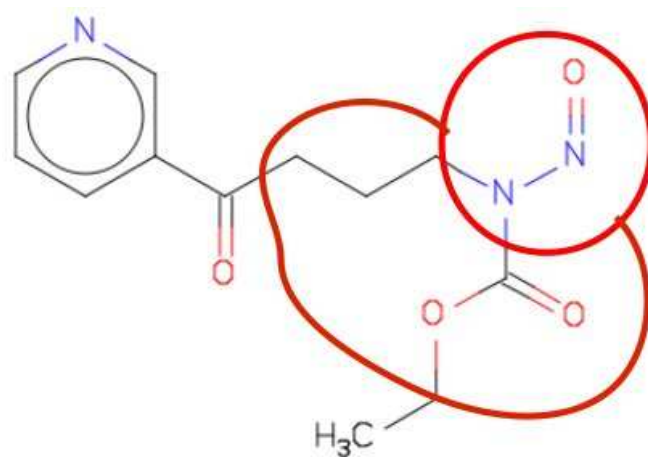


(b) RF approximation

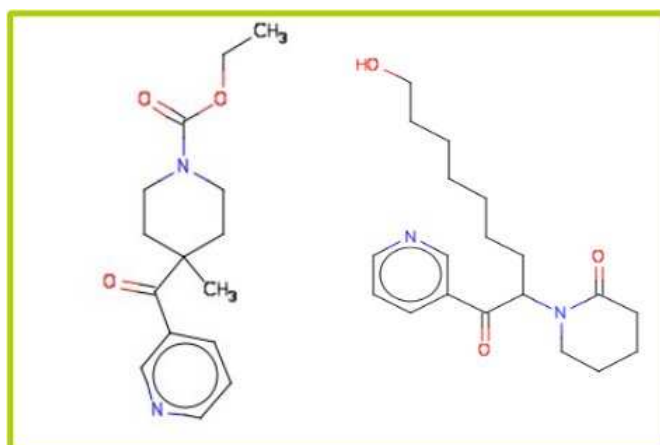


(c) SVM approximation

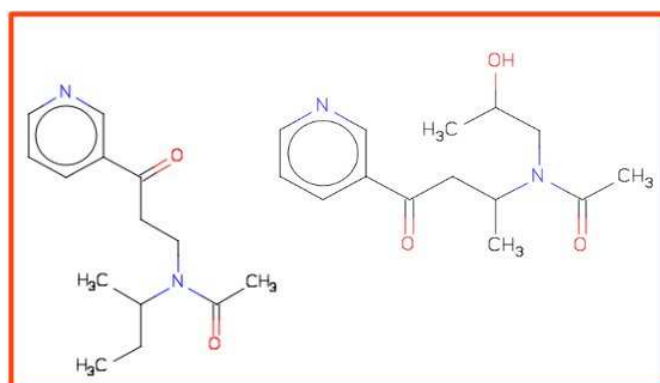
Figure 8: The most significant component of the gradient plotted at each test point. This was based on the gradient computed for the PLS, RF and SVM models with the first seed and a training set size of 800 data points. The underlying contour plot represents the correct areas of significance, yellow indicating component 1 and red indicating component 2.



(a) Compound and selected substructure, to be replaced



(b) Examples of generated compounds predicted negative



(c) Examples of generated compounds predicted positive

Figure 9: A compound with an unfavorable prediction and examples of the generated compounds.

collected by Kazius, *et al.*⁶⁵ These computations took roughly one month using six computing cores for building models and solving the system of Diophantine equations. The rebuilding process was distributed on a heterogeneous grid using at maximum 100 nodes for rebuilding compounds. The computational time for the individual compounds varied between a few minutes for aliphatic compounds to several days for some of the bicyclic aromatic compounds. Up to 15000 compounds have been generated for a single query, but in some cases only a handful of compounds have been generated. The amount of generated compounds affects the computational time needed. Out of the 303 compounds 181 were predicted to be positive by the model and the corresponding statistics (covering number of generated compounds, percentage generated compounds predicted positive, computational time, number of minimal solutions and the number of steps taken by the Diophantine equation solver) for those computations are provided in the supporting information of Paper **II**. The remaining compounds were predicted to be negative by the model and therefore not optimized. In the set of optimized compounds, 18 of the 28 approved toxicophores described by Kazius, *et al.* were covered (see supporting information of Paper **II** for details). The fact that 40% of the compounds were predicted negative was however a drawback. One reason for that may be that local QSAR models were used for retrieving the substructure instead of a global model. A global model would have been more accurate over the entire domain, which presumably would have affected the outcome. To thoroughly study local and global QSAR models an investigative study of their properties and risks was conducted, Paper **IV**.

These results show that the automated work-flow for molecular optimization is a useful tool in drug development. To enumerate and rebuild complete compounds with the algorithms proposed by Visco, *et al.*¹³ and Churchwell, *et al.*⁴⁹ is a costly procedure when it comes to drug like compounds containing multiple cycles. In drug discovery small changes in the chemical structure can have a considerable effect on the activity.^{67,68} By adding the identification of significant substructures and limiting the size of similar substructures, as proposed here, it was possible to regenerate compounds that are valid in a drug-design context.

The computational time needed for the method increases dramatically with the number of unique signatures due to the combinatorial explosion of possible compounds. The reason for this was that the method is a deterministic search method, meaning that all signature combinations that may result in new compounds will be found but it may take long time to find them. With a higher number of unique signatures the chemical diversity among the generated molecules will be higher and therefore a trade-off between computational time and chemical diversity has to be made.

3.3. Finding Significant Substructures

Today substructural alerts are applied in many areas of drug discovery to warn chemists of potential problems with functional groups. The tool that has been developed in the present work aids the modeler or chemist in finding the substructures that actually separates the data at hand. Similarly it is possible to evaluate existing substructural alerts. This method finds substructures that are overrepresented in the data. It will however not replace chemical knowledge, which guides the field today. Chemical knowledge is however subjective and if an alert based on chemical knowledge is not separating the data it is likely that the alert is not alone responsible for the outcome.

To depict the substructures Ogham⁶⁹ was used to create molecular visualizations of each substructure projected on a compound, like in Figure 5. For each significant signature the signature string together with the accuracy, *p*-value, signature similarity, positive and negative count was written to a spreadsheet table together with the visualization of the substructure on one of the molecules.

The work flow was demonstrated using AMES mutagenicity data from CCRIS⁶⁶ from which compounds and corresponding activity have been collected according to the conditions described by Kazius et al.⁶⁵ The data set has been divided into 10 subsets and evaluated using cross validation, where the model has been trained on the remains of the data and tested on the subset. An external validation set of 880 compounds reported by Young, *et al.*⁷⁰ has also been used. The method has been compared to *gaston*,⁷¹ *gSpan*⁷² and *PAFI*.⁷³ These three methods retrieves frequent subgraphs from the data. The frequent subgraphs have been converted to SMARTS⁴⁷ and based on the SMARTS significant substructures have been retrieved based on *p*-value, accuracy and occurrence. The data analysis have been performed according to the procedure described by Kazius, *et al.*⁶⁵ where compounds that contains no significant substructures have been classified as negative. In contrast to Kazius *et al.* the threshold for the accuracy has been 80% and the threshold for the *p*-value has been 0.05.

The method has been applied with different number of required hits (5, 10, 20, 50 and 100). The results have been visualized in Figures 10, 11, 12. Figure 10 shows the accuracies for the methods at the different numbers of required hits on the test data and the validation data respectively. It shows that the possibility to mine the data on a low occurrence threshold results in an increased accuracy. Figure 11 shows the computational effort for training and predicting using the different methods. It shows that the complexity of the significant signatures method is lower than the the complexity for the other methods. Finally Figure 12 shows how the number of generated substructures vary for the different methods with different number of required hits. For *PAFI*, *Gaston* and *gSpan* the training time and the prediction time are very similar. One reason for this could be that in the prediction all matches of a substructure on a compound was

located whereas in the training only the first match was sufficient for the method. The SMARTS matching was conducted using OEChem.⁷⁴ For 5 and 10 required hits Gaston and gSpan crashed on insufficient memory, and for PAFI this happened for 5 required occurrences.

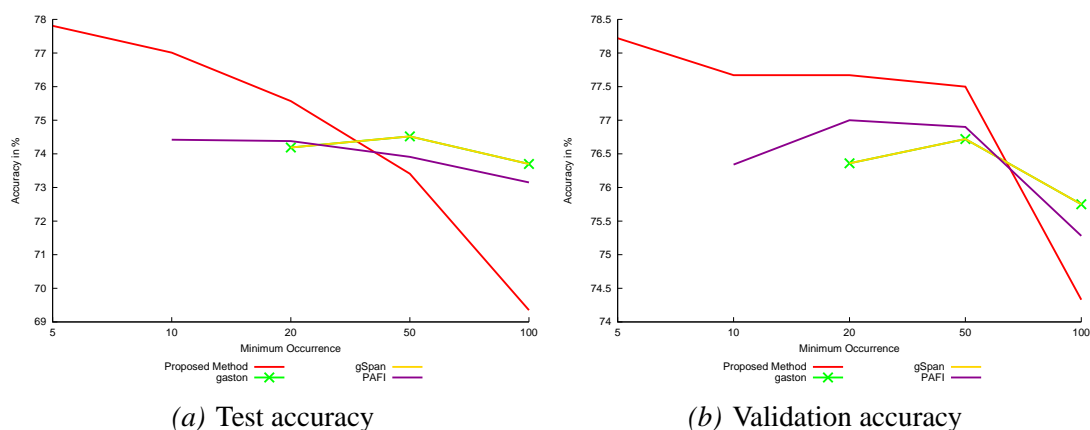


Figure 10: Average test and validation accuracy for the methods used on the AMES data

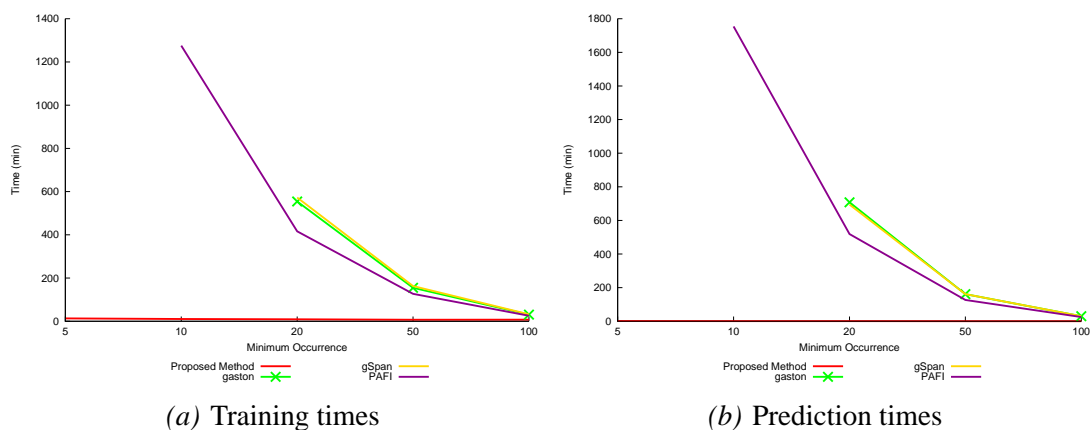
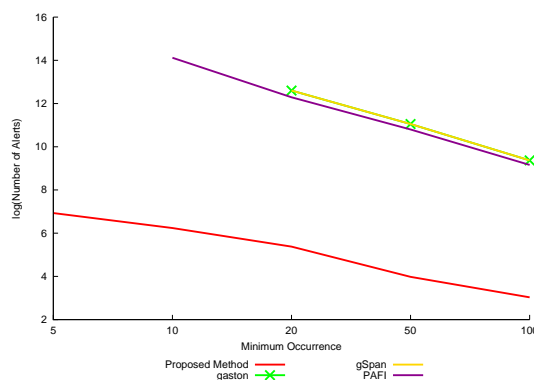


Figure 11: Average training and prediction times for the methods used on the AMES data

The method in Paper **III** illustrates a complete automated work flow for SMARTS like pattern generation. The method treats the data objectively and generates a set of significant substructures according to the user defined constraints. By displaying active and inactive substructures the algorithm can aid the user in changing the compounds to hit or avoid hitting the target of interest. The method is computationally inexpensive compared to existing methods and the results indicate that it can replace large parts of the analysis that is performed manually today.



(a) Generated substructures

Figure 12: The logarithm of the number of generated alerts for each method

3.4. Evaluation of QSAR Modeling Strategies

Local or Global modeling strategies has been investigated using simulated and real data, and the results clearly show that use of all available data is preferable. Here only the regression case for the simulated data is presented. Complete results for both the simulated and the real data can be found in Paper IV and its corresponding supporting information.

3.4.1. Experimental Setup

In the simulation studies all parameters for the underlying relationship were known, the answer to the problem was known and thereby it was possible to design responses based on different combinations of descriptors and to study the effect of *Local* and *Global* modeling strategies thoroughly.

In this case descriptors have been drawn from the gamma distribution function resulting in a descriptor set that will mimic the distribution of real chemical descriptors. The simulated descriptor space consists of three different descriptors, d_1 , d_2 and d_3 , drawn such that $d_1 \in \Gamma(4, 1)$, $d_2 \in \Gamma(9, 1)$ and $d_3 \in \Gamma(7.4, 1)$. The function determining the response is $f_j = \cos(d_{2j} - \bar{d}_2)/(1 + (d_{1j} - \bar{d}_1)^2) + 1.2 * \sin(1.3 * (d_{3j} - \bar{d}_3))$ where d_{ij} is the j th point drawn from the i th descriptor above and \bar{d}_i is the mean of the drawn points for that descriptor.

For each modeling strategy 10 seeds have been used and for each seed 1000 examples have been generated. The examples have been drawn uniformly into 10 bins and each bin has been used as a test set with the remaining data as a training set. This results in 100 *Global* models for each case and 10000 *Local* models since for each point in the respective test sets a *Local* model has been built. The *Local* modeling strategy has been tested using 10, 20, 50, 100, 200, 400, 600 and 800 near neighbors from the training set. The near neighbors have been selected using the Euclidean norm as a distance metric in descriptor space. The results of the simulations are presented as the averaged accuracy or Root-Mean Square Error (RMSE) for each case.

All Simulations were conducted in R⁶² using the machine-learning libraries `e1071`⁷⁵ for SVM, `randomForest`⁷⁶ for RF and `pls`⁷⁷ for PLS. The accuracy of SVM models is very sensitive to parameter optimization and therefore the SVM models have been optimized using a grid search over the γ parameter ($2^n, n = [-5 : 0]$) and the ε parameter ($2^n, n = [-5 : -1]$). The γ parameter is the exponent in the RBF kernel function and ε is the tolerance of the termination criterion, controlling the width of the loss-insensitive zone in the loss function.

These simulations have resulted in a vast amount of data. All the results can be obtained from the histograms in Paper IV and the corresponding supporting information. The essence of the results are presented here in a simplified manner.

Figure 13 display the averaged overall RMSE of the *Global* and the *Local* models using the different number of near neighbors, defined above, within the applicability domain for the RF algorithm. The first Figure, 13(a) shows the *RMRL* and the *RMIL* cases. From this it was possible to see that for *RMRL*, where no new information was added in the *Local* model, there will be no predictive gain, *i.e.* no change in RMSE. If however information was added, like in the *RMIL* case, the predictivity will increase, *i.e.* the error decreases for models with few near neighbors. In Figure 13(b) the *IMIL* case is added showing lower error which means that by adding the information to the *Global* model the result would be even better than the *Local RMIL* model. Finally, Figure 13(c) shows the full picture where the *IMRL* case shows that the errors increase if the *Local* neighborhood was retrieved without all necessary information with respect to the underlying relationship.

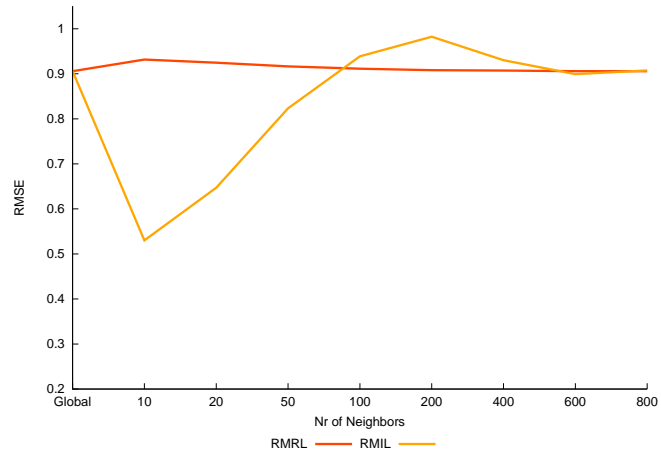
Figure 14(a) shows that the errors increase dramatically for *Local* models with few near neighbors for predictions outside of the intended applicability domain. This visualizes the risk of local models. That is followed by Figure 14(b) which shows that for most models the chance that the local models should predict better is below 50% within the applicability domain and outside the applicability domain the chances are even worse. This does not hold for the *RMIL* case but if that extra information was added to the *Global* model, the *IMG* case, then the effect vanishes.

In the paper RF, SVM and PLS are compared for both regression and classification models. The results for RF and SVM are very similar, but the PLS models differ. This is shown in Figure 15.

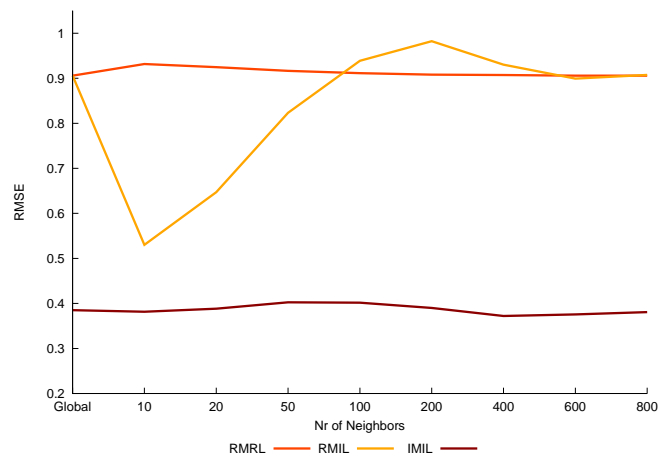
Compare the *Global* models in Figure 15(a) and 13(c). The *IMIL* cases for RF shows a low error but the PLS show a high error. In fact the error is the same for *IMIL* and *RMRL* which implies that PLS can not describe the non linearity in the data. Further more Figure 15(b) and 14(a) show that there is a substantial difference in error between PLS and RF outside of the applicability domain. This shows that there is a large risk associated with the use of *Local* models and in particular *Local* PLS models, note that the two Figures display different ranges

of RMSE values.

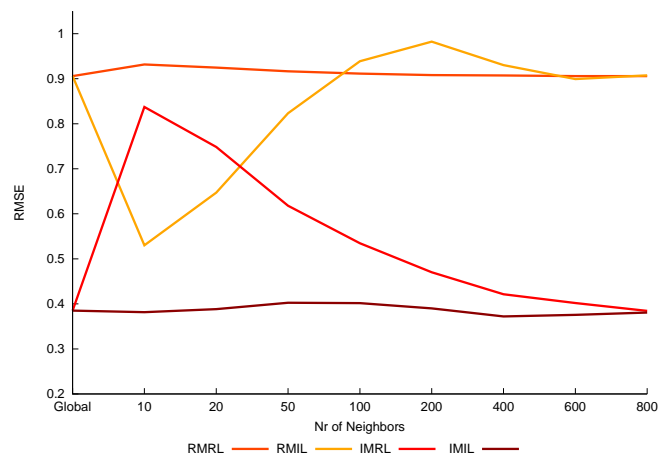
This shows that non-linear machine-learning methods are capable of handling *Global* data sets. Linear methods however fail to handle the non linearized in the data and can not utilize the extra information for this non-linear case. When the underlying relationship is linear, all machine-learning models give approximately the same errors, as can be seen in the Supporting Information of Paper **IV**.



(a) Part I

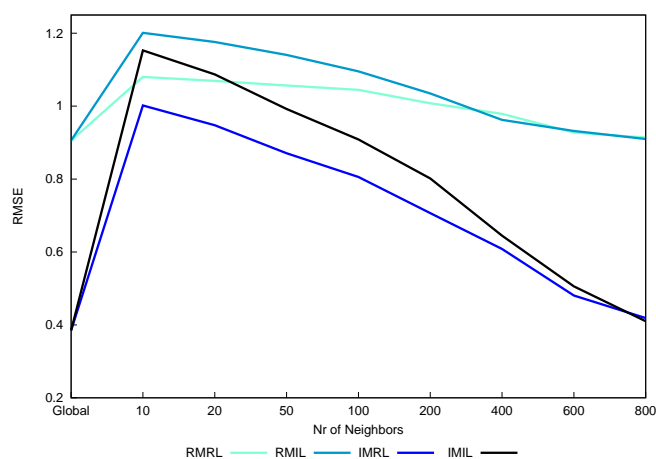


(b) Part II

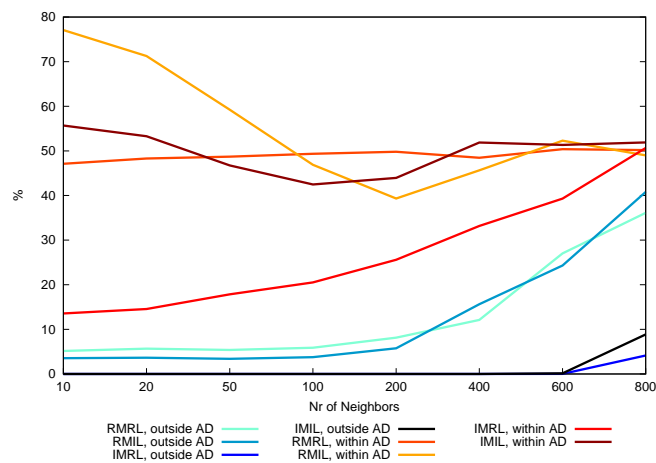


(c) Full Description

Figure 13: RMSE of the RF machine-learning algorithm for the different *Local* model cases, and their respective *Global* counterparts, within the applicability domain

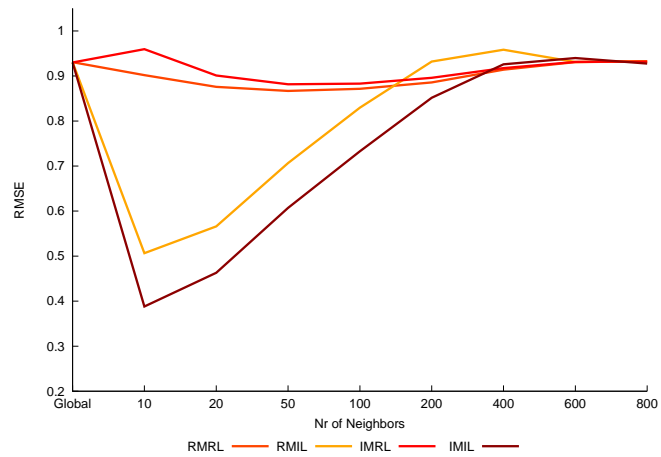


(a) RMSE of the RF machine-learning algorithm for the different *Local* model cases, and their respective *Global* counterparts, outside of the applicability domain.

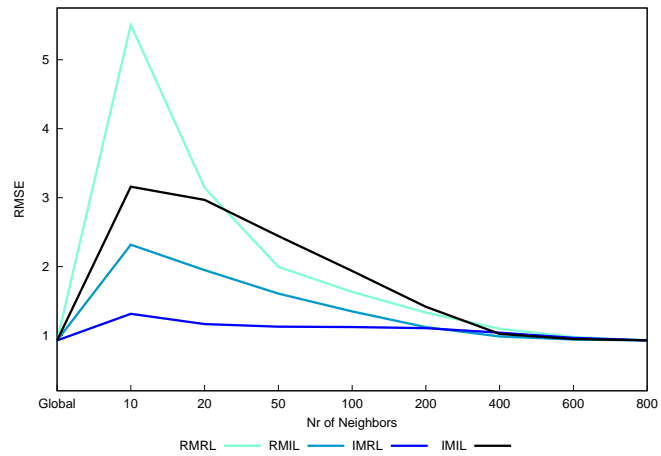


(b) The percentage of cases where the *Local* model is more accurate than the *Global* one for predictions within and outside of the applicability domain, AD, of the *Local* models.

Figure 14



(a) Within the applicability domain

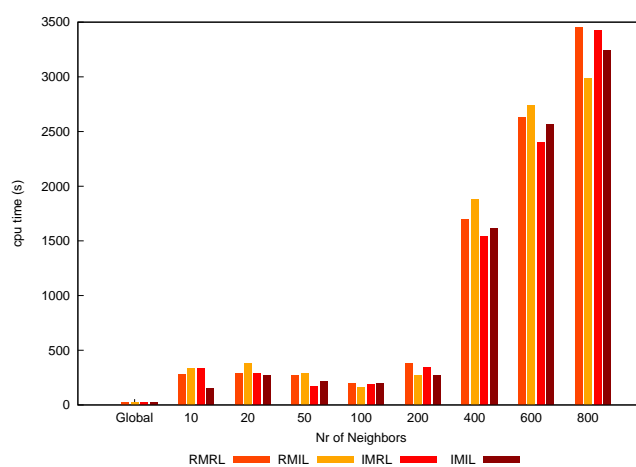


(b) Outside of the applicability domain.

Figure 15: RMSE of the PLS machine-learning algorithm for the different *Local* model cases, and their respective *Global* counterparts

3.4.2. Computational Costs

The computational effort for neighbor extraction, building and predicting the *Local* models is shown in Figure 16. The figure indicates that there is a substantial growth in CPU time needed as the number of neighbors increases, which indicates that building *Local* models is time consuming. Here it is important to remember that a *Local* model is built for each query, thus building local models using 800 near neighbors almost amounts to building as many global models as there are queries.



(a) Regression models

Figure 16: CPU time needed to train the local models with respect to the number of neighbors. The computations were run on a heterogeneous grid.

3.4.3. Discussion

For a predictive modeling system there is an interest in being able to predict all incoming compounds. When doing predictive modeling the model with the highest overall accuracy is most commonly the best and preferred model. Sometimes this approach does not lead to an accurate enough model and in an attempt to overcome this problem models based on a sub-domain of the data are built.⁷⁸ The sub-domain should then capture the problem in a more accurate way. Paper IV questions the use of sub-domain models for predictive modeling on three major points:

1. There is no statistically validated improvement in accuracy for *Local* models
2. The risk of falling outside of the applicability domain of the *Local* model is high. Additionally, outside the applicability domain the accuracy of the *Local* model is very poor compared to the accuracy of the *Global* model

3. There appears to be a substantial increase in computational cost associated with the *Local* models

The obvious questions are: How can the modeler be sure that future compounds will fall within the model space and that this sub space really is relevant for the issue of interest? If several sub-domain models are built, which one is to be trusted?

The results show that a *Local* modeling strategy is only better than a *Global* strategy if additional information, which is relevant for the underlying relationship, is added in the neighbor search. If a *Local* model according to the definition used here, performs better than a *Global* model it is advisable to add that additional information, used to retrieve near neighbors, to the *Global* model which will then be even better. This is exemplified in Figure 13(c) by comparing *RMG* and *RMIL*. Here information that is important for the underlying relationship has been added in the neighbor search for the *Local* model resulting in a lower RMSE for the models that use few near neighbors. The *Global* model updated with the same information is *IMG* and that model is more accurate than the best *RMIL* model. Figure 15 also shows a substantial difference in RMSE for the different PLS models which is due to the fact that the PLS *IMG* cannot utilize the extra information and performs approximately as its *RMG* counterpart, which indicates that the additional information in the *Ideal* compared to the *Restricted* case is of non-linear nature with respect to the response.

The series *IMRL* show a case where the *Local* neighborhood is known not to be *Local* with respect to the underlying relationship. Here the *Local* modeling strategy will give less accurate models compared to the *Global* modeling strategy.

Figure 16 show that the computational cost for predicting data sets using *Local* models, as they have been defined in this study, is generally higher compared to *Global* models. Thus it appears that to maintain a prediction system using *Local* models a large computational resource needs to be dedicated for these computations.

4. Concluding Remarks and Future Perspective

Paper **I** – **III** utilizes substructures in compounds to visualize chemical liabilities, retrieve and replace fragments with undesired properties. The results indicate that substructures are important in describing compound properties. Compound properties such as clogP are in most cases calculated from structures and substructural properties so when substructures, like signatures, are used as descriptors for QSAR modeling molecular properties are implicitly accounted for.⁴⁸ Using substructures can reduce modeling errors that arise from multiple mappings, *i.e.* from substructures via for example clogP and polar surface to the biological response of interest. In Paper **I** the most important component of the decision function was retrieved by gradient computations. Further analysis of what the gradients can reveal could be a topic for future studies. An analysis of more than the single most important component could possibly improve the method as well as a more thorough investigation of the effects of smoothing for the discrete gradient computations or some estimate on step size.

The local optimization method presented in Paper **II** spans the molecular space around the compound of interest in a good way but is however still time consuming. The most time consuming part is the Diophantine equation solver, and a parallel implementation of that step could potentially improve the method. In the method description compounds with high fingerprint similarity to the query substructure are used. It could be interesting to study the behavior of the method if the compounds for setting up the system of constraints equations were selected in other manners. For example compounds could be selected at random or using similarity but in a range that is moderately similar to the query substructure.

Paper **III** shows that it is sufficient to search a subspace to find common features that can separate data in a good way. This use of the signatures is interesting since the complexity of the signature generation algorithm is polynomial whereas the complexity of a general substructure generation algorithm is exponential. This property could perhaps be utilized in maximum common substructure searches, where it could potentially reduce the computational time significantly for large data sets.

The study of the performance and properties of local and global modeling strategies in Paper **IV** shows that the local modeling strategies is associated with relatively high computational costs, high risks in using *Local* and the *Local* models give no reliable increased predictive performance. If there is interest in studying local properties, then a global model could be applied and using the gradient computations outlined in Paper **I**, local behavior could be studied.

The results from Paper **IV** suggests that building *Global* models and keeping them updated with new information that might affect the underlying relationship is the best way to consistently assure accurate models.

REFERENCES

- [1] Nikolova, N.; Jaworska, J. *QSAR & Comb. Sci.* **2004**, *22*, 9–10.
- [2] Grover, M.; Singh, B.; Bakshi, M.; Singh, S. *Pharm. Sci. & Tech. Today* **2000**, *3*, 28–35.
- [3] Perkins, R.; Fang, H.; Tong, W.; Welsh, W. *Env. Tox. & Chem.* **2003**, *22*, 1666–1679.
- [4] Sabljic, A. *Chemosphere* **2001**, *43*, 363–375.
- [5] Fox, T.; Kriegl, J. M. *Current Topics in Medicinal Chemistry* **2006**, *6*, 1579–1591.
- [6] Helma, C.; Kazius, J. *Current Computer-Aided Drug Design* **2006**, *2*, 123–133.
- [7] Borman, S. *Chem. Eng. News* **1990**, *68*, 20–23.
- [8] Lipnick, R. L. *Trends Pharmacol. Sci.* **1986**, *7*, 161–164.
- [9] Hansch, C.; Leo, A.; Taft, R. W. *Chem. Rev.* **1991**, *91*, 165–195.
- [10] Lantican, B. P.; Muir, R. M. *Plant Physiol.* **1967**, *42*, 1158–1160.
- [11] Hansch, C. *Acct. Chem. Res.* **1969**, *2*, 232–239.
- [12] Todeschini, R.; Consonni, V. *Molecular Descriptors for Chemoinformatics*; Wiley, 2009.
- [13] Visco, D. P. J.; Pophale, R. S.; Rintoul, M. D.; Faulon, J.-L. *J. Mol. Graphics Model.* **2002**, *20*, 429–438.
- [14] Skvortsova, M. I.; Baskin, I. I.; Slovokhotova, O. L.; Palyulin, V. A.; Zefirov, N. S. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 630–634.
- [15] Churchwell, C.; Rintoul, M. D.; Martin, S.; Visco, D. P.; Kotu, R.; Larson, R. S.; Sillerud, L. O.; Brown, D. C.; Faulon, J. L. *J. Mol. Graph. Model.* **2004**, *22*, 263–273.
- [16] Guha, R.; Dutta, D.; Jurs, P. C.; Chen, T. *J. Chem. Inf. Model.* **2006**, *46*, 1836–1847.
- [17] Zhang, S.; Golbraikh, A.; Oloff, S.; Kohn, H.; Tropsha, A. *J. Chem. Inf. Model.* **2006**, *46*, 1984–1995.
- [18] Martin, Y. C.; Kofron, J. L.; Traphagen, L. M. *J. Med. Chem.* **2002**, *45*, 4350–4358.
- [19] Boström, J.; Hogner, A.; Schmitt, S. *J. Med. Chem.* **2006**, *49*, 6716–6725.
- [20] Yuan, H.; Wang, Y.; Cheng, Y. *J. Chem. Inf. Model.* **2007**, *47*, 159–169.
- [21] Gavaghan, C. L.; Hasselgren, C.; Blomberg, N.; Gert, S.; Boyer, S. *Journal of Computer-Aided Molecular Design* **2006**, *21*, 189–206.
- [22] Breiman, L. *Machine Learning* **2001**, *45*, 5–32.
- [23] Vapnik, V. *The nature of statistical learning theory*; Springer Verlag, New York, 1995.
- [24] Wold, S.; Ruhe, A.; Wold, H.; Dunn III, W. J. *SIAM J. Sci. Stat. Comput.* **1984**, *5*, 735–743.
- [25] Breiman, L. *Machine Learning* **2001**, *45*, 5–32.
- [26] Christianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines and other kernel-based learning methods*; Cambridge University Press, 2000.
- [27] Ho, T. K. Random Decision Forest. *3rd Int Conf. on Document Analysis and Recognition*, 1995; pp 278–282.
- [28] Breiman, L. *Machine Learning* **1996**, *24*, 123–140.

- [29]Ho, T. K. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **1998**, *20*, 832–844.
- [30]Amit, Y.; Geman, D. *Neural Computation* **1997**, *9*, 1545–1588.
- [31]Boser, B.; Guyon, I.; Vapnik, V. *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory* **1992**, 144–152.
- [32]Cortes, C.; Vapnik, V. *J. Machine Learning* **1995**, *20*, 273–297.
- [33]Mercer, J. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* **1909**, *209*, 415–446.
- [34]Karush, W. M.Sc. thesis, Univ. of Chicago, 1939.
- [35]Kuhn, H. W.; W., T. A. Nonlinear Programming. *Proc. Second Berkeley Symp. on Math. Statist. and Prob.*, 1951; pp 481–492, <http://projecteuclid.org/euclid.bsmsp/1200500249>.
- [36]Valiant, L. G. *Commun. ACM* **1984**, *27*, 1134–1142.
- [37]Vapnik, V. N.; Chervonenkis, A. Y. *Theory of Probability and its Applications* **1971**, *16*, 264–280.
- [38]Wold, S.; Ruhe, A.; Wold, H.; Dunn, W. J.; III, *SIAM Journal on Scientific and Statistical Computing* **1984**, *5*, 735–743.
- [39]Wold, H. *J. Multivariate Analysis* **1966**, 391–420.
- [40]Franke, L.; Byvatov, E.; Werz, O.; Steinhilber, D.; Schneider, P.; Schneider, G. *J. Med. Chem.* **2005**, *48*, 6997–7004.
- [41]Liaw, A.; Wiener, M. *Breiman and Cutler's random forests for classification and regression, version 4.5-18*; 2006, Fortran original by Leo Breiman and Adele Cutler, R port by Andy Liaw and Matthew Wiener, <http://cran.r-project.org/doc/packages/randomForest.pdf>.
- [42]Henshaw, W. D. *J. Comput. Phys.* **1994**, *113*, 13–25.
- [43]Gutman, I.; Polansky, O. E. *Mathematical Concepts in Organic Chemistry*; Springer-Verlag, 1986.
- [44]*Handbook of Graph Theory*; Gross, J. L., Yellen, J., Eds.; CRC Press, 2004.
- [45]Weininger, D. *JCICS* **1988**, *28*, 31–36.
- [46]SMILES; Accessed Jan 03 2010, <http://www.daylight.com/dayhtml/doc/theory/theory.smiles.html>.
- [47]SMARTS; Accessed Aug 20 2009, <http://www.daylight.com/dayhtml/doc/theory/theory.smarts.html>.
- [48]Faulon, J.-L.; Visco, D. P. J.; Pophale, R. S. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 707–720.
- [49]Churchwell, C. J.; Rintoul, M. D.; Shawn, M.; Visco, D. P. J.; Kotu, A.; Larson, R. S.; Sillerud, L. O.; Brown, D. C.; Faulon, J.-L. *J. Mol. Graphics Model.* **2004**, *22*, 263–273.
- [50]Contejean, E.; Devie, H. *Inf. Comp.* **1994**, *113*, 143–172.
- [51]Johnson, D. E.; Blower, P. E.; Myatt, G. J.; Wolfgang, G. H. I. *Curr. Opin. Drug Discovery*

- Dev* **2001**, *18*, 76 – 79.
- [52]DEREK; lhasa LTD, 2009, http://www.lhasalimited.org/index.php?cat=2&sub_cat=2#.
- [53]Multicase; MultiCASE Inc., 2009, <http://multicase.com/>.
- [54]Nicolaou, C. A.; Pattichis, C. S. Molecular Substructure Mining Approaches for Computer-Aided Drug Discovery: A Review. 2006.
- [55]Deshpande, M.; Kuramochi, M.; Wale, N.; Karypis, G. *IEEE Transactions on Knowledge and Data Engineering* **2005**, *17*, 1036–1050.
- [56]Nicolaou, C. A.; Tamura, S. Y.; Kelly, B. P.; Bassett, S. I.; Nutt, R. F. *J. Chem. Inf. Comp. Sci.* **2002**, *42*, 1069–1079.
- [57]Faulon, J.-L.; Churchwell, C. J. *J. Chem. Inf. Comp. Sci.* **2003**, *43*, 721–734.
- [58]Chang, C.-C.; Lin, C.-J. *LIBSVM: a library for support vector machines*; 2001, Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [59]Ahlberg Helgee, E. M.Sc. thesis, Chalmers University of Technology, 2005.
- [60]SYBYL Atom Types; Tripos, 2009, http://www.optive.com/mol2/atom_types.html.
- [61]Wold, S. *Quantitative Structure-Activity Relationships* **1991**, *10*, 191–193.
- [62]R Development Core Team, *R: A Language and Environment for Statistical Computing, version 2.7.2*; R Foundation for Statistical Computing: Vienna, Austria, 2008, ISBN 3-900051-07-0.
- [63]Wehrens, R.; Mevik, B.-H. *Partial Least Squares Regression (PLSR) and Principal Component Regression (PCR), version 2.1.0*; 2007, <http://cran.r-project.org/doc/packages/pls.pdf>.
- [64]Dimitriadou, E.; Hornik, K.; Leisch, F.; Meyer, D.; Weingessel, A. *Misc Functions of the Department of Statistics (e1071), TU Wien, version 1.5-16*; 2006, <http://cran.r-project.org/doc/packages/e1071.pdf>.
- [65]Kazius, J.; McGuire, R.; Bursi, R. *J. Med. Chem.* **2005**, *48*, 312–320.
- [66]TOXNET - Chemical Carcinogenesis Research Information System; Accessed Nov 22, 2006, <http://toxnet.nlm.nih.gov>.
- [67]Guha, R.; Van Drie, J. H. *J. Chem. Inf. Model.* **2008**, *48*, 646–658.
- [68]Sisay, M. T.; Peltason, L.; Bajorath, J. *J. Chem. Inf. Model.* **2009**, *49*, 2179–2189.
- [69]Ogham 2D Chemical Structure Layout and Rendering; Accessed on Jan 3 2010, <http://www.eyesopen.com/docs/ogham/1.7.0/html/index.html>.
- [70]Young, S.; Gombar, V.; Emptage, M.; Cariello, N.; Lambert, C. *Chemometrics and Intelligent Laboratory Systems* **2002**, *60*, 5–11.
- [71]Nijssen, S.; Kok, J. N. *Electronic Notes in Theoretical Computer Science* **2005**, *127*, 77 – 87, Proceedings of the International Workshop on Graph-Based Tools (GraBaTs 2004).
- [72]Yan, X.; Han, J. gSpan: Graph-Based Substructure Pattern Mining. *ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining*, Washington, DC, USA, 2002;

p 721.

- [73]Kuramochi, M.; Karypis, G. *IEEE Trans. on Knowl. and Data Eng.* **2004**, *16*, 1038–1051.
- [74]*Openeye Scientific Software*; accessed on Jan 3 2010, <http://www.eyesopen.com>.
- [75]Dimitriadou, E.; Hornik, K.; Leisch, F.; Meyer, D.; ; Weingessel, A. *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*; 2006, R package version 1.5-16.
- [76]Liaw, A.; Wiener, M. *R News* **2002**, *2*, 18–22.
- [77]Wehrens, R.; Mevik, B.-H. *pls: Partial Least Squares Regression (PLSR) and Principal Component Regression (PCR)*; 2007, R package version 2.0-1.
- [78]Penzotti, J. E.; Landrum, G. A.; Putta, S. *Current Opinion in Drug Discovery and Development* **2004**, *7*, 49–61.

Acknowledgments

There are always new challenges and ideas that one wants to try, which sometimes results in work done here and there but not properly written up. Someone said that “the work ain’t finished until the paperwork is done” and that is very true. Working with research can be stimulating, fun and great. It can also be really tough and time consuming and in the toughest times one needs most support and help. I would like to thank all friends and colleagues for your support and opinions on both vital and unimportant matters. However, I would like to specifically thank the following:

Dr. Scott Boyer, my supervisor, for supporting the work and for reading papers and manuscripts.

Dr. Lars Carlsson, my assistant-supervisor, for support, discussions and help with implementations and formulations. You have been my support in computational matters and have been able to twist and turn complex problems with me.

Senior lecturer Morten Grøtli and Prof Kristina Luthman from the University of Gothenburg for the introduction to some real chemistry and making my dissertation possible.

The Linux support at AstraZeneca, Mölndal, especially Andreas Loong and Martin Budsjö for help, support and patience when I happened to generate a few terabytes of data over night...

My parents, Elisabet Ahlberg and Bertil Helgee, my sister Edit and grandmother May for always being there for me and for all discussions regarding everything from house construction issues via psychology and philosophy to applied math, physics and chemistry.

Hanna Petersson, my lovely cohabitant, for all the things we do together, and for not giving up on me. I love You!

Lovisa Olsson, for our common interest in cooking, baking and for being a great friend and listener.

My friends at West Coast Jitterbugs and Chalmers Sångkör for temporarily taking my mind off work.