

1 Introduction

This chapter describes the prediction of toxic effects with *data mining* techniques in a stepwise approach. Methods are characterized in terms of principle advantages, shortcomings and interpretability of the results. We seek to present techniques that are effective as well as universally applicable. We also give some software recommendations focusing on open-source software, which is not only free, but also transparent and extensible.

1.1 Problem description

Chemicals influence biological systems in a huge variety of biochemical interactions, mostly on the cellular and molecular level. In Toxicology, the aim is to understand the biochemical mechanisms involved and the degree to which chemicals induce toxicological activity in living organisms with respect to a well-defined endpoint.

In *predictive toxicology*, we exploit the toxicological knowledge about a set of chemical compounds in order to predict the degree of activity of other compounds. More specifically, we mathematically model the relationship between specific properties of training compounds (i.e. compounds for which the degree of activity is known) and their toxicological *activity* and apply the model to query compounds (i.e. compounds for which the degree of activity is not known) to obtain predicted activities.

The process of model-building is called (Quantitative) Structure Activity Relationship ((Q)SAR). SARs are models based on structural features, and QSARs rely on quantitative (frequently physico-chemical) properties. The most general mathematical form of a (Q)SAR is:

$$\text{Activity} = f(\text{physicochemical and/or structural features}) \quad (1)$$

The training compounds are stored in databases together with their activity values. Formally, we have observed data for n cases $((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n))$, where each $\mathbf{x}_i = (x_{i_1}, \dots, x_{i_m})^T$ is a feature vector of m input values and each y_i is the associated activity (dependent variable). The observations and corresponding activities can therefore be compactly represented in a *feature matrix* \mathbf{X} (sometimes also referred to as the set X) and a corresponding activity vector \mathbf{y} . Our primary interest is to predict the unknown activity value y_q for a query compound \mathbf{x}_q . A predicted value for \mathbf{x}_q is commonly referred to as $f(\mathbf{x}_q)$, associated with a confidence value c that is derived from certain properties of the model that describe the goodness of the fit. One of these properties is the chemical similarity between training compounds and the query compound, denoted as $\text{sim}(\mathbf{x}_i, \mathbf{x}_q)$.

For quantitative activities the prediction process is called *regression*; for qualitative activities (i.e. a finite set of activity classes) it is called *classification*.

1.2 Predictive toxicology approaches

According to [1, 2], predictive toxicology models can be classified as statistical and expert/rule-based approaches (see Figure 1). Statistical approaches use general toxic endpoints and activity values gathered for a wide range of structures and are primarily driven by information inherently present in the data, not from human expert knowledge. Expert/rule-based approaches build (Q)SAR generalizations from individual chemicals to chemical classes based on prior knowledge, heuristics, expert judgement and chemical and biological mechanism considerations.

For the purpose of this chapter we will focus on statistical (Q)SAR techniques and on the expert system aspects (e.g. categorization, feature selection) that are frequently used in (Q)SAR modelling.

1.2.1 Traditional (Q)SAR models

Traditional (Q)SAR methods use linear regression techniques to identify a relationship between chemical features and experimental activities. The classical approaches are

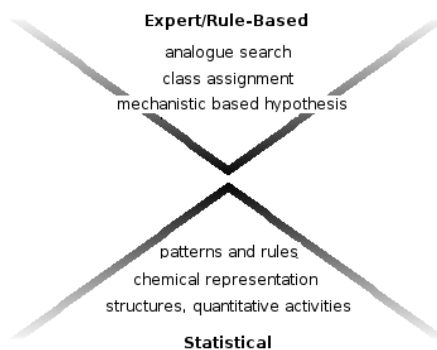


Figure 1: Types of (Q)SAR modelling: While Expert Systems make use of expert knowledge, specifically with feature selection and modelling, statistical approaches derive most things in an automated fashion.

Hansch-analysis

$$\log TA_{100} = 0.92(\pm 0.23)\log P + 1.17(\pm 0.83)HOMO - 1.18(\pm 0.44)LUMO + 7.35(\pm 6.9)$$

Descriptor values can be drawn from literature or calculated by computer programs.
and

Free-Wilson-analysis

$$\log TA_{100} = 0.92(\pm 0.23)\log P + 1.17(\pm 0.83)HOMO - 1.18(\pm 0.44)LUMO + 7.35(\pm 6.9)$$

The interpretation of linear (Q)SAR models is rather straightforward by inspecting the most important features (i.e. features with high coefficients). Overfitting is rarely a problem, because of the limited expressiveness of the model. For the same reason the applicability of linear models is restricted to congeneric series with similar modes of action. Another problem with traditional (Q)SAR techniques is the selection of features for endpoints, that are very complex and incorporate many different and potentially unknown biological mechanisms. In this case it is very likely to miss important features or to suffer from the curse of dimensionality, if too many features have been selected.

1.2.2 Constraints in predictive toxicology

Toxicological experiments are frequently expensive, time consuming and may require a large number of animal experiments. Therefore, it is usually impossible to create experimental data for congeneric series specifically for (Q)SAR modelling. For this reason most toxicological (Q)SARs have to rely on existing datasets, which are in many cases very diverse in respect to structure, biological mechanisms, data origin and quality.

Fortunately publicly available structural and biological databases (e.g. PubChem (<http://pubchem.ncbi.nlm.nih.gov/>), Toxnet (<http://toxnet.nlm.nih.gov/>), DSSTox (<http://www.epa.gov/nheerl/dsstox/>)) have grown substantially in recent years. Despite this wealth of information, databases are often characterized by the following properties that make modelling difficult:

- The chemicals are not *congeneric*, i.e. they do not share a common substructure and act by common mechanism.
- The activities are noisy with missing values.
- The activity distributions are skewed and/or have other non-normal properties.
- A substantial amount of toxicity data is confidential and not accessible to the general public.

With data mining techniques from artificial intelligence research, it is possible to use information from diverse databases much more efficiently than traditional (Q)SAR approaches, that rely on congeneric compounds. Many of these techniques can be seen as automation of various aspects from the (Q)SAR modelling process. They work similar to a human (Q)SAR expert, who separates the training set into subsets with similar mechanisms, selects and calculates chemical features, and builds (Q)SAR models for the individual subsets. Many of them can be also seen as an attempt to base scientific decisions on sound statistical criteria.

1.2.3 Data mining in (Q)SAR modelling

Data mining can be described as finding nontrivial, previously unknown and potentially useful information in large amounts of data. In predictive toxicology data mining techniques can be used for all model building tasks that will be described in the following sections. It is e.g. possible to create, aggregate and select relevant features, to group chemicals according to their similarity, or to create complex prediction models. In this context we see traditional (Q)SAR techniques also as data mining tools, that identify linear models in databases with chemical features and experimental toxicity data.

1.3 (Q)SAR model development

Independent of algorithmic and implementation details, the process of (Q)SAR modelling can be subdivided into four basic steps:

feature generation → feature selection → model learning → model validation → model interpretation

The following sections will be organized according to this sequence, but we can also refine the whole procedure into more detail:

1. Definition of the goal of the project and the purpose of the (Q)SAR models.
2. Creation or selection of the training set.
3. Checking the training set for mistakes and inconsistencies, and perform corrections.
4. Selection of the features relevant to the project (by expert knowledge or data mining).
5. Selection of the modelling technique.
6. Exploratory application and optimization of the modelling and feature selection techniques to see if it provides useful results.
7. Application of the selected and optimized techniques to the training set.
8. Interpretation of the derived model and evaluation of its performance.
9. Application of the derived model, e.g., to predict the activity of untested compounds or an external test set with known activity values.

It is usually impossible to use all features, because they are highly correlated and contain much noise. A high dimensional feature space is also sparsely populated and hardly interpretable. For this reason a thorough selection of features is extremely important in step 4. This can be achieved through a combination of objective feature selection and a further refinement step (projection-based or supervised method).

Steps 5-7 employ data mining techniques for distance weighting and distance measures, similarity measurements and regression.

A software package that implements a rather complete (Q)SAR solution using data mining methods is WEKA, (Waikato Environment for Knowledge Analysis) [3]. There are also several packages that make cheminformatics libraries written in other languages available in R [4, 5]. The OpenTox project (<http://www.opentox.org>) aims to build an open source framework for predictive toxicology. It will incorporate many the tools mentioned in this chapter together with automated validation routines and facilities to build graphical user interfaces.

1.3.1 Criteria for the selection and evaluation of data mining algorithms

The OECD (Organisation for Economic Co-operation and Development) has developed acceptance criteria for (Q)SARs for regulatory purposes [6, 7]. Specifically, these are

1. a defined endpoint,
2. an unambiguous algorithm, with a clear description of the mathematical procedure,
3. a defined applicability domain with descriptor and structure space definitions,
4. measures of goodness-of-fit (r), robustness (q^2) and predictivity (external prediction),
5. a mechanistic interpretation shall be given, if possible.

These rather broad criteria contain essential aspects of good practice in (Q)SAR modelling. However, for the purpose of data mining applications, these criteria are rather general and do not provide enough algorithmic details for their implementation. Within the following sections we will propose formal definitions and algorithms for OECD criteria, especially for the assessment of feature space properties, applicability domains and model validation (items 3. and 4.).

The following section will provide more detail about the individual steps that are involved in the development of predictive toxicology models.

2 Feature generation

The goal of feature generation is the description of chemical structures. There is no set of universal features that describe a compound equally well for all purposes.

The classical (Q)SAR methods (Hansch-Analysis and Free-Wilson) both employ multiple linear regression to build a model. Hansch-Analysis was historically used to derive a statistical relationship between measured quantities of chemicals and toxicological activities exhibited by those chemicals. The octanol-water partition coefficient ($\log P$) for example is closely related to lipophilicity and describes the ability of a chemical to pass membranes in the body. It is therefore correlated with many toxic effects and can be used to statistically model these endpoints.

Hansch analysis uses physico-chemical properties and substituent constants, while Free-Wilson uses chemical fragments derived from the 2D-structure. Such descriptors can be (among others)

- Structural properties (structural alerts / substructures from general feature mining),
 - structural alerts from experts (substituent constants)
 - hybrid (refinement of structural alerts by data mining techniques)
 - substructures derived by graph mining algorithms
 - spectroscopic data
- Experimental and calculated physico-chemical properties, quantum chemical parameters or graph-theoretical indices (electronic, hydrophobic or steric), e.g. $\log P$, pKa.

- Measured biological properties, e.g. from short term assays, high-throughput screening, -omics data

Structural properties can be obtained directly from chemical compounds and are called primary features, while experimental or calculated quantities are secondary features.

The selected feature type affects not only the predictive performance but also the biological rationale for the algorithm and the interpretation of individual predictions. The interpretability of models and predictions benefits from features that are well known to chemists and toxicologists and have a clear mechanistic relevance.

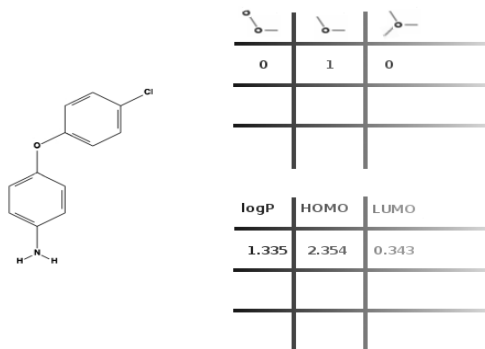


Figure 2: Features obtained from a chemical. Upper: qualitative primary features (SAR), below: quantitative secondary features (QSAR)

From a statistical point of view it is important to classify features as qualitative or quantitative. Qualitative features indicate the presence or absence of some feature, while quantitative features give a measured or calculated amount on some numerical scale. Structural features are frequently qualitative, e.g. they indicate the presence or absence of some substructure, and experimental features are frequently quantitative. Historically, both types were used and models are referred to as either SAR or QSAR for qualitative and quantitative features, respectively. Hansch analysis uses quantitative and Free-Wilson uses qualitative features.

Open-source software projects that provide chemical toolkits and libraries for feature generation and many other purposes are associated in the *blue obelisk group* (e.g. OpenBabel, CDK) [8].

3 Feature selection

Traditionally the (Q)SAR modeler has to use his/her knowledge about toxicological mechanisms to decide which features will be included in a (Q)SAR model. Especially with complex and poorly understood toxic effects, the selection of features is likely to be incomplete and error prone. With data mining we can use objective criteria to select relevant features automatically from a large set of features, in order to filter out noise and to find informative patterns within the data. Using a large feature space together with objective criteria for feature selection reduces the risk of ignoring important features and allows an automated detection of new structural alerts. A basic understanding of statistical tests is vital for the application of feature selection algorithms [9].

3.1 Unsupervised techniques

Methods that do not consider toxic activities (the dependent variable) are called unsupervised techniques. They remove redundant information and/or construct fewer, more informative features. Table 1 lists some popular unsupervised techniques for feature selection.

With *objective feature selection*, each pair of features is compared. This is usually implemented by iteratively adding features to the data matrix \mathbf{X} when they pass the tests. In SAR modelling, i.e. with qualitative features,

Name	Theory of operation	Retains features?
Objective Feature Selection	selects features iteratively	yes
Cluster Analysis	groups correlated features	yes
Principle Components Analysis	projects data to a lower dimension	no

Table 1: Some popular unsupervised techniques for feature selection

objective feature selection can contain identity, zero and singularity tests, checking for features that occur in the same structures, and for features that do not occur or occur only once in the training compounds. In QSAR modelling, i.e. with quantitative features, it is possible to check for standard deviation (a feature carries little information when it has a low standard deviation), singularity (where the values are the same for all compounds except one) and correlation.

Cluster analysis is a procedure for grouping together similar features in clusters, thus enabling the algorithm to pick one representative for each cluster. The problem is to decide *a priori* how many groups should be built as this depends to a large extent on the data. Most popular are techniques that recursively partition the features. A very advanced technique is known as self-organizing maps [10]. Computational complexity varies greatly for these approaches.

Principal components analysis is a projection of the data to a lower-dimensional vector space, thereby eliminating correlations between features. It works by finding the eigenvalues and eigenvectors of the covariance matrix of \mathbf{X} . A rotation matrix is created that projects the data into the vector space made up by the most influential eigenvectors (the principle components), accounting for most of the data’s variance. Usually, a decision is made beforehand for a specific percentage of variance and the algorithm uses only the most influential eigenvectors to reach this threshold. By not using all eigenvectors data compression through dimensionality reduction is achieved. The amount of compression depends on the correlation within the original data. principle components analysis is a frequently applied technique and well documented [11]. It is available as a function in R [5].

Using principle components analysis harms the interpretability of a model, as the original feature space gets lost. However, the *loadings* can be inspected to assess the influence of the original features present in the principal components. Objective feature selection and clustering techniques are well behaved in this respect. Unsupervised techniques are not prone to overfitting since only redundant information is removed.

3.2 Supervised techniques

Supervised feature selection tries to select features that correlate well with the dependent variable, i.e. the activities. In the SAR case (i.e. with qualitative features) it is possible to assign significance values to features which can be used as a preprocessing step to selection and is discussed first. Similar techniques are available for quantitative features. Significance values for features are also valuable when it comes to model building.

3.2.1 Significance tests

Given two different sets of compounds (e.g. compounds with/without a certain substructure) it is interesting to find out whether the two samples differ significantly in respect to their toxicological activities. The activity values form sample distributions and we can use statistical tests to find out if the distributions of both sets differ significantly (see Figure 3). If the difference is significant, it is possible that the investigated substructure contributes to the toxic activity. This association is of course purely statistical and human expert knowledge is still needed to determine the exact biological mechanisms.

A popular choice for the comparison of qualitative results (e.g. carcinogen/non-carcinogen classifications) is the χ^2 test and the *Kolmogorov-Smirnov test* can be used for the comparison of qualitative data (e.g. LD₅₀ values). The probability (*p*-value) that an observed difference is due to chance can be calculated from the test statistics ^a.

^aIf multiple tests are performed (e.g. for the evaluation of sets of substructures) the *p*-values have to be corrected. If *p* is the

A common significance threshold is 0.05 which means that one false positive difference is accepted in twenty cases.

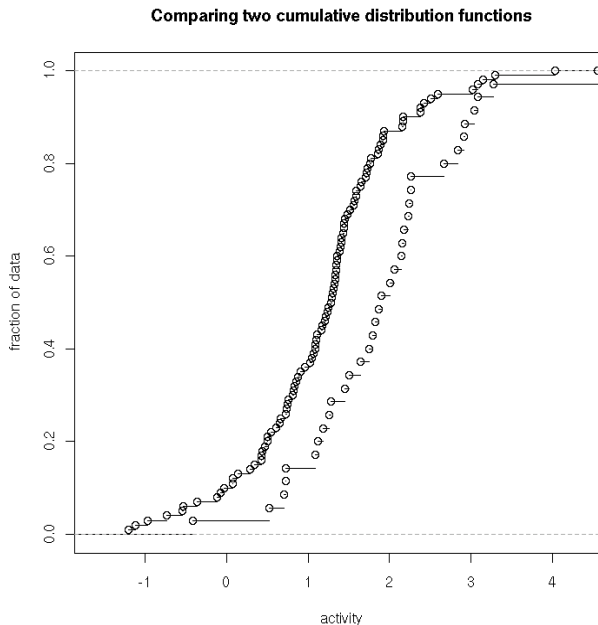


Figure 3: A comparison of the cumulative activity distributions of two sets of activity values x and y with sizes 100 and 35, respectively. The mean value of x is 1.0, the mean value of y is 2.0. It is highly unlikely (Kolmogorov-Smirnov test gives $p = 0.0001319$) that x and y have been drawn from the same data source.

The set size is very important for significance tests. A set size below 12 is usually considered as “very small” and below 30 as “small”. Mean values differ greatly for very small sets and are still unstable for small sets [9]. In other words, to avoid chance effects no significance tests should be performed for very small sets. For small sets, permutation tests can be helpful.

3.2.2 Supervised selection

In *supervised feature selection*, a particular selection of features is evaluated and assigned a score (reward signal). This process is iterated many times to identify an optimal feature set. This makes the method computationally expensive and bears the danger of overfitting the selection with respect to the training data, reducing the ability to predict external data. Significance tests for features can be used as a preliminary step for supervised feature selection which is a special case of *reinforcement learning* [14]. Table 2 lists some popular supervised techniques for supervised feature selection.

Name	Theory of operation	Retains features?
Forward Selection/ Backward Elimination	iterative (de)selection	yes
Simulated Annealing	probabilistic selection	yes
Genetic Algorithm Subset Selection	probabilistic selection	yes

Table 2: Some popular supervised techniques for feature selection

significance threshold for a specific test, then $1 - p$ is the probability of drawing a negative feature f_i . For n independent tests, the probability that no single test is positive for f_i is $(1 - p)^n$, which converges to 0. This increases the probability of Type I errors (*false discovery rate*). A simple correction is the *Bonferroni correction*, which simply divides each p -value by n . More sophisticated methods to control the false discovery rate exist [12]. In settings where the absolute values are less important than rankings corrections can be omitted. There exists an R package for multiple tests [13] that features also functions for *permutation tests*, *bootstrapping* and *jackknifing* procedures that increase the reliability of tests.

The naïve approach in supervised feature selection is to evaluate all possible subsets of features. However, most of the time, this is computationally too expensive. *Forward selection* starts with an empty set of features and successively adds features that increase the fit, starting with the most significant features. But forward selection has drawbacks, including the fact that each addition of a new feature may render one or more of the already included features non-significant. *Backward elimination* goes the other way round: it starts with all features and removes those that have little contribution to the model. This method has also limitations, sometimes features are dropped that would be significant when added to the final reduced model. Stepwise selection is a compromise between the two methods, allowing moves in either direction.

Simulated annealing switches in each iteration to a different selection of features with a probability that depends on the goodness of fit and a “temperature” parameter t . The lower the fit and the higher t , the greater the probability for switching. t is decreased with every iteration (therefore the name) until a certain threshold is reached. The idea is to overcome local maxima by “jumping”.

Genetic algorithm subset selection successively narrows down the feature space by evolutionary means. It recombines pairs of sets of features by mimicking crossover and mutation to obtain better features. In each “generation”, the remaining candidates are evaluated by a “fitness function” and the process repeats itself with the more successful ones.

4 Model learning

4.1 Data preprocessing

Most predictive toxicology techniques do not work directly on raw experimental measurements, but rely on some sort of preprocessing. This can involve statistical calculations (e.g. for the determination of TD_{50} or LD_{50} values) as well as expert knowledge (e.g. human carcinogenicity classifications) to aggregate replicates, doses or multiple experiments into a single value. It is important to understand the properties and limitations of these techniques before attempting to model a derived variable (e.g. are assumptions behind the procedures verified, are quantitative values good indicators of toxic potencies, are the results expressed in molar values). A description of the data aggregation procedures should be part of the documentation for the first OECD criteria (defined endpoint).

A common (Q)SAR practice is to log-transform quantitative variables to the range of values and to achieve a normally distributed dataset. It is still important to check the normality assumption for each dataset, before parametric methods are applied. If the normality assumption is not met, “binning” the data into discrete values might help. A more generally applicable solution is to use non-parametric methods, that make no assumptions about data distributions.

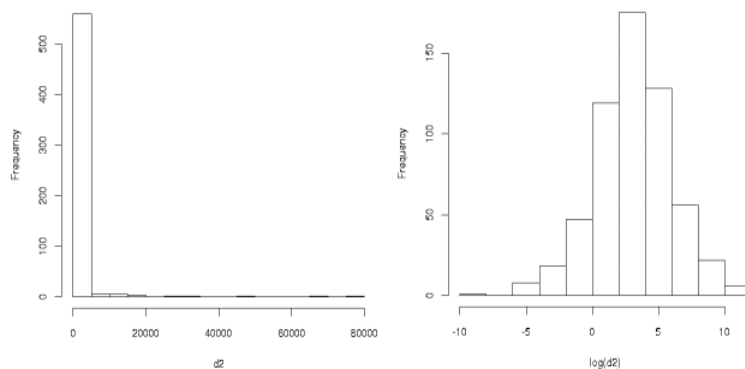


Figure 4: Histogram of original (left) and log-transformed database activities (right)

4.2 Modelling techniques

Table 3 lists popular modelling techniques for (Q)SAR regression and classification.

Name	Theory of operation
Traditional QSARs (Hansch, Free-Wilson)[15]	Multilinear regression on physico-chemical properties/structural features.
Artificial Neural Networks [16, 17, 18]	Nonlinear multidimensional parameterized model mimicking the function of neurons.
Support Vector Machines [19]	Robust classification algorithm using hyperplanes to split the feature space into class regions.
Decision Trees and Rule Learners [20]	Hierarchical rules from recursive partitioning of the training data
k-Nearest Neighbor Techniques [20]	Derive the prediction from the activities of structurally similar compounds

Table 3: Popular modelling techniques

Multilinear models have been in use since a long time. As linear equations, they are easy to use and relatively straightforward to interpret. For n instances they are defined as the coefficients that minimize the error on a system of n linear equations:

$$y_i = b_1x_{i1} + \dots + b_mx_{im} + d \quad i \in \{1, \dots, n\}, \quad (2)$$

or in a more compact notation,

$$\mathbf{y} = (\langle \mathbf{X}, \mathbf{b} \rangle + d). \quad (3)$$

where $\langle \cdot, \cdot \rangle$ denotes the normal dot product and \mathbf{b} and d are the coefficients to learn. Multilinear models assume linear relationships between features and activities, therefore the expressiveness is limited and the model will perform poorly if these conditions are met. The prediction $f(\mathbf{x}_q)$ is obtained by

$$f(\mathbf{x}_q) = (\langle \mathbf{x}_q, \mathbf{b} \rangle + d). \quad (4)$$

The remaining (nonlinear) models are able to fit diverse datastructures (in fact many can fit arbitrary data), but are frequently too complex for interpretation or have a poor biological rationale. Learning can take very long and overfitting is more likely. For both types of neural networks, a lot of decisions have to be made about architecture, learning rate and activation functions.

Support vector machines are perhaps the most prominent approach that represent the family of kernel-based techniques (*kernel machines*). Another member of this family that is quite new to machine learning are *gaussian processes* [21]. Successful approaches have demonstrated that kernel machines are more solid than and can serve as a replacement for artificial neural networks in a wide variety of fields [22]. Kernel machines use a so-called kernel function to project the data to a feature space with higher dimensions where it is easier to separate classes linearly or to perform linear regression. The coefficients \mathbf{b} are a linear combination of training vectors for the coefficients $\{a_1, \dots, a_n\}$:

$$b_j = \sum_{i=1}^n a_i y_i \mathbf{x}_i, \quad (5)$$

which allows to rewrite equation (3) as an integration over the training data:

$$f(\mathbf{x}_q) = \sum_{i=1}^n a_i y_i \langle \mathbf{x}_q, \mathbf{x}_i \rangle + d \quad (6)$$

The dot-product $\langle \mathbf{x}_q, \mathbf{x}_i \rangle$ denotes the cosine of the angle between \mathbf{x}_q and \mathbf{x}_i (assuming unit length of the vectors). It can thus be seen as a similarity measure with geometric interpretation. The dot-product is the simplest instance of a kernel function. However, kernel machines usually don't perform learning in the original feature space. The key is to replace \mathbf{x}_q and \mathbf{x}_i in the right-hand side of equation (6) by higher-dimensional representations $\phi(\mathbf{x}_q)$ and $\phi(\mathbf{x}_i)$, where $\phi : \mathbb{R}^m \rightarrow \mathbb{R}^{m'}$ with $m' > m$ is called a *map*.

The expression $\phi(\mathbf{x})$ is not calculated directly in practice due to combinatorial explosion. Kernel Machines exploit the fact that it only occurs in dot-products in the algorithms. This allows to bypass direct calculation of the map. Instead, a so-called kernel function $k : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$ is used that calculates $\langle \phi(\mathbf{x}_q), \phi(\mathbf{x}_i) \rangle$ directly in the input space ("kernel trick"). In fact, k can be any positive definite function denoting similarity^b. The final predictive equation for kernel machines is then given by

$$f(\mathbf{x}_q) = \sum_{i=1}^n a_i y_i k(\mathbf{x}_q, \mathbf{x}_i) + d \quad (7)$$

For classification, i.e., when $\mathbf{y} = \{-1, +1\}^n$, the sign of the prediction can be used: $f(\mathbf{x}_q) = \text{sgn}(\sum_{i=1}^n a_i y_i k(\mathbf{x}_q, \mathbf{x}_i) + d)$

4.3 Global models

If a model is fitted to training data in advance, i.e. without knowing the query structure, then the model is called "global". At query time, global models simply evaluate the model function on the training instance to obtain a prediction. Therefore, global models require low memory and give fast predictions once the training phase is over. However, complex functions in a high dimensional feature space suffer from data sparseness and are easily overfitted, thereby destroying its predictive ability for new compounds.

Overfitting is the process of fitting a model with many parameters too accurately to the training data. Despite a perfect fit for the training data, the resulting model has poor generalization capabilities and is not predictive for unknown query instances. To avoid overfitting, it is necessary to use additional techniques (e.g. crossvalidation, Bayesian priors on parameters or model comparison, that can indicate when further training does not result in better generalization. The process of overfitting a neural network during training is also known as overtraining.

The effect of data sparseness in high dimensions is due to the so-called "curse of dimensionality" [23]. Roughly speaking, with increasing dimensions, subsets of the data span a growing subspace that approaches the whole feature space rapidly. In other words, with a high number of dimensions, the distance between compounds increases and the neighborhoods get sparse.

4.4 Instance-based techniques (local models)

It is frequently possible to identify congeneric subsets within diverse datasets. Such a group of structures can be said to represent a local (Q)SAR. Global (Q)SAR methods may not recognize such local relationships if they don't use very complex (nonlinear) functions and many features.

Local models obtain a prediction for a query structure using its "local neighborhood" rather than considering the whole dataset, i.e. they only use training compounds that are similar to the query structure with respect to some distance measure. They can also use fewer features than global models. Local models cannot be built before the query instance is known. Most local models not only defer model learning but also defer clustering the training compounds into neighborhoods until a query instance is to be predicted. Because of that they are also termed as "lazy".

With *lazy learning*, for each distinct query a new approximation to the target function is created. The approximations are local and differ from one another; therefore, for the whole feature space, many different approximations

^bIt has been shown that for any positive definite function k of the described kind there exists a (possibly infinite) expansion in terms of Basis Functions ϕ such that $\forall \mathbf{u}, \mathbf{v} \in \mathbf{X} : k(\mathbf{u}, \mathbf{v}) = \langle \phi(\mathbf{u}), \phi(\mathbf{v}) \rangle$

are used at different locations. The single approximations may be simple (e.g. linear), but seen as a whole they can approximate a complex function. They are also robust because they depend only on the data points close to the query instance. In contrast to eager learning, the computational burden for the prediction is higher, since all the training is done at query time.

4.4.1 Similarity measures

The idea is to cluster congeneric compounds by chemical similarity and to use only the nearest neighbors as training instances, and/or weight the contribution by distance. The similarities between the query compound and the training compounds are also useful for determining applicability domains and prediction confidences (see section 6).

Traditionally, chemical similarity is determined by expert knowledge to obtain clusters of congeneric chemicals (chemical classes). The assignment of chemical classes is however frequently ambiguous and does not necessarily reflect biological mechanisms. For fully automated data mining approaches, a wide variety of similarity indices have been proposed [24].

Willet et. al. [24] have reviewed 22 structural similarity indices by searching databases for chemical analogues. They showed that combinations of descriptors perform best, among them the Tanimoto-, the Russel/Rao-, the Simple Matching- and the Stiles-coefficient. They all work on 2D fragment bit-strings, indicating the presence or absence of structural features in a compound. The Tanimoto index, for example, calculates the ratio of common features between two compounds.

For quantitative features, distance-based indices are also well suited (Euclidean or Mahalanobis distance, see also 6). A data structure that can be used for an efficient calculation are kd-trees (`libkdtree++`, available from <http://libkdtree.alioth.debian.org/>).

Chemical similarity can also be assessed by supervised techniques (i.e. by taking the training activities into account). The contribution of each feature to the Tanimoto index can be weighted for example with the p -values of statistical significance tests [25].

4.4.2 Prediction from neighbors and distance weighting

Having determined the similarities between the query structure and each training structure, these values can be used to select a local neighborhood to the query structure and to train the model on these compounds only. Different methods are available for neighbor selection:

- Counting cutoff: use the k nearest neighbors, where k is a fixed number.
- Similarity cutoff: use the neighbors that are more similar than some fixed similarity threshold.
- Soft selection: use all compounds and weight their contribution to the model by their similarity values, where more similar compounds get higher weights. Doing so is no harm to model precision, because distant training points will have little effect on the approximation. The only drawback is that model building takes longer.

Of course, distance weighting can also be applied in the cutoff approaches. In dense populations, a kernel function is frequently used to additionally smooth the similarity. A variety of smoothing functions has been reviewed in [26]. Most widely used are gaussian kernels of the squared-exponential form

$$sim_g(\mathbf{x}_i, \mathbf{x}_q) = \exp\left(-\frac{1}{2}sim(\mathbf{x}_i, \mathbf{x}_q)^2\right). \quad (8)$$

This kernel creates a progression phase in the neighborhood and generally ameliorates conditions. It can also be stretched by using the general gaussian probability distribution function with adjustable width.

The actual prediction can then be obtained by rather simple models, e.g. with distance weighted majority votes for classification problems and multilinear regression for regression problems. More complex models can be tried, if the simple approaches do not give satisfactory results.

5 Combination of (Q)SAR steps

Efficient *graph mining* techniques are currently a strong research focus of the data mining community. As chemicals can be represented as graphs many of these techniques can also be used for chemoinformatics and (Q)SAR problems. Most of them focus on the efficient identification of relevant substructures (combining the feature generation and selection steps) or on using graph structures directly for classification/regression.

5.1 Constraint based feature selection

Complete feature sets can be built by decomposing the structures of the training set into all subgraphs of a certain type (e.g. paths, trees, graphs). As this process is computationally very expensive various techniques to reduce the search space have been developed. Traditionally size limits have been used, but this can lead to the loss of large significant fragments. More recently frequency based constraints have been introduced (e.g. in MolFea [27], FreeTreeMiner [28], gSpan [29], Gaston [30]). The idea is to restrict the search space by stating the minimal and/or maximal frequencies in two classes of compounds (e.g. carcinogens/non-carcinogens) and the algorithm finds efficiently all subgraphs that fulfil these constraints.

Although restricting the search for substructures by min/max constraints is intriguing at a first glance, there are several problems associated with this approach:

- The goal of feature selection is to find fragments that are *significantly* correlated with an toxicological outcome. Most graph mining algorithms support only monotonic constraints (e.g. minimum, maximum frequencies), but test statistics are usually convex. Although extensions for convex functions (e.g. χ^2) exist, they prune the search space rather inefficiently in our experience.
- As frequency based searches use activity information it is important to repeat the search whenever the training set changes (e.g. if a query compound has been identified and removed from the database and for each fold during cross-validation) (see section 7). Having to repeat the fragment search frequently (e.g. for model development or cross-validation runs) may render the initial performance advantage as useless. Storing the complete fragment search and repeating only the selection process can be a more efficient alternative.

5.2 Graph kernels

Graph kernels have been developed to incorporate graph structures into support vector predictions (see section 4.2). The crucial part is to define a kernel, that indicates the chemical similarity of two compounds (see also section 4.4.1). An example that uses substructure fingerprints is the Tanimoto kernel. For two compounds \mathbf{x}_i and \mathbf{x}_j the kernel function is the proportion of features f that are shared between \mathbf{x}_i and \mathbf{x}_j :

$$k_{\mathbf{x}_i, \mathbf{x}_j}^t = \frac{|\{f | f \subseteq \mathbf{x}_i \wedge f \subseteq \mathbf{x}_j\}|}{|\{f | f \subseteq \mathbf{x}_i \vee f \subseteq \mathbf{x}_j\}|}, \quad (9)$$

Different techniques have been proposed that work on the adjacency matrix of graphs and derive different features (directed or undirected, labeled or unlabeled subgraphs, etc.) as well as marginalized graph kernels that obtain features from Markov random walks. In practise support vector machines with graph kernels can perform remarkably well, for an extended discussion of see e.g. [31].

6 Applicability domain

6.1 Definition and purpose of applicability domains

Jaworska et.al. define the *applicability domain* of a (Q)SAR as “the physico-chemical, structural or biological space, knowledge or information on which the training set of the model has been developed, and for which it is applicable to make predictions for new compounds.” [32]. A critical assessment of the applicability domain is important to distinguish between reliable and unreliable predictions.

The purpose of applicability domains is to tell whether the modeling assumptions are met. With data mining methods, this is a two-fold task: (i) are training compounds similar enough to the query instance, and (ii), how is the descriptor space populated (e.g. how dense are the training compounds, is the query compound within the subspace spanned by the training compounds).

6.2 Determination of applicability domains

In traditional (Q)SAR approaches, the applicability domain is determined by the modelled endpoint and the selection of compounds and features. In Hansch-Analysis for example, features triggering the endpoint are selected and consequently the applicability domain consists of compounds that contain those features, or whose features lie in the respective range, i.e. that belong to a certain chemical class.

With data mining methods, the practical application of the applicability domain concept requires an operational definition that permits the design of an automatic (computerized), quantitative procedure to determine a model’s applicability domain. Although up to now there is no single generally accepted algorithm for determining the applicability domain, there exists a rather systematic approach for defining interpolation regions [33]. The process involves the removal of outliers with the help of a probability density estimation using different distance measures. When using distance metrics care should be taken to use an orthogonal and significant feature space. This can be achieved by different means of feature selection and successive principle components analysis.

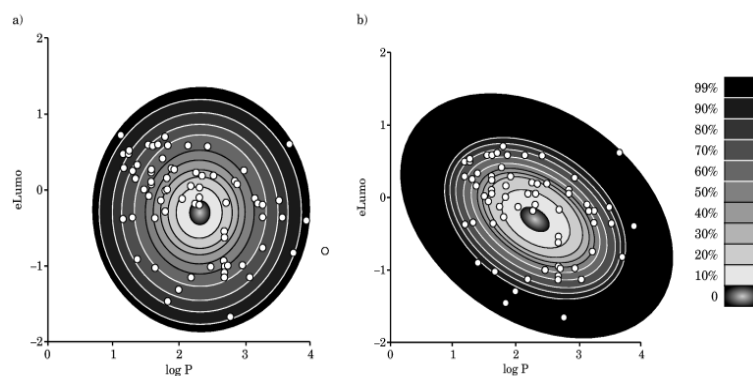


Figure 5: Probability density estimation using euclidean distance (left) and mahalanobis distance (right) (taken from [33] with permission).

For practical applications a viable approach consists of two steps: (i) transform the training data so that the feature space has acceptable properties (low dimensionality and orthogonality) and (ii) generate a probability density allowing to assess important aspects of the distribution. More specifically, the following steps can serve as a guide towards a reliable confidence index:

- Create a low-dimensional, orthogonal feature space and prune redundant information with objective feature selection followed by principle components analysis with a threshold for variance loss. The optimal threshold can be estimated by crossvalidation.

- For normally distributed training compounds create a probability density distribution estimation taking into account the data’s “shape” using Mahalanobis distance D_M , defined as

$$D_M(\mathbf{x}_i) = \sqrt{(\mathbf{x}_i - \mu_{\mathbf{X}})^T R_{\mathbf{X}}^{-1} (\mathbf{x}_i - \mu_{\mathbf{X}})} \quad (10)$$

for any datapoint \mathbf{x}_i , where $\mu_{\mathbf{X}}$ is the center of the distribution \mathbf{X} and $R_{\mathbf{X}}$ is the covariance matrix of the data. Leverage h , which is directly related to Mahalanobis distance, is defined as $h(\mathbf{x}_i) = D_M(\mathbf{x}_i)/n - 1$ (see Figure 5). For non-normally distributed data, non-parametric methods have to be applied [33].

- Identify if the query compound is an *outlier* with the estimated density distribution. As a rule of thumb, a compound \mathbf{x}_i is an outlier if $h(\mathbf{x}_i) > 2p/n$, where p is the number of parameters in the model [9].
- Create a confidence value c for every prediction by combining density distribution estimates and a global chemical similarity index (e.g. Tanimoto index). Typically, a well-spread neighborhood in feature space combined with high chemical similarity should give high quality predictions. Typical implementations use a fixed ratio of chemical similarity and density distribution estimation to determine the confidence. By convention c ranges between 0 (lowest confidence) and 1 (highest confidence).

A recent approach in this direction, termed “Automated Lazy Learning QSAR” achieved high accuracy using an automatically calculated applicability domain from distributional properties of the training set [34]. Specifically the applicability domain incorporated the average (euclidean) distance and standard deviation of distances to the center of the distribution. To account for chemical similarity, the lazy learning approach used similarity weighting based on gaussian kernels.

Suitable open-source software for these purposes is available from the R project [5].

7 Model validation

The goal of *model validation* is to evaluate the performance for untested compounds, i.e. the predictive power of the model. This step is often interleaved with applicability domain estimation: by predicting compounds it can be assessed how well the applicability domain discriminates between good and bad predictions.

7.1 Validation procedures

7.1.1 Retrofitting the training set

Especially with multilinear (Q)SAR models predicting the compounds in the training set is still a popular “validation” method, although this technique does not evaluate the performance for unseen instances. The problem is less obvious for multilinear regression, because it cannot fit the training data exactly, but many data mining techniques that can accommodate to any data distribution (e.g. neural networks). If no precautions against overfitting are taken they achieve 100% accuracy on the training set, but the overfitted function performs poorly for new predictions. For this reason it is crucial to test every model performance with structure, that have not been used for model building.

7.1.2 Artificial validation sets

As it is usually impossible to create experimental data for validation purposes it is common practice is to split the available data into training and test sets prior to modelling. The model is developed with the training set and the test set is used to validate the model prediction. Although the procedure may seem to be simple and straightforward, there are several possible pitfalls:

- *All* test set information has to be excluded from the training set. This means that all supervised feature selection methods have to be performed only with training set information.
- The composition of the test set has a huge impact on validation results. If the test set has many compounds within the applicability domain prediction accuracies will increase, test sets that are very dissimilar to the training set will achieve low accuracies.
- As validation results depend strongly on the test set composition, it would be ideal to validate with a test set that has been drawn randomly from future prediction instances - unfortunately these are rarely known to the model developer.
- If the training and test set are drawn from the same source they still share common information, e.g. about activity distributions. This will lead to overly optimistic results for techniques, that derive *a priori* probabilities from training set distributions (e.g. naive Bayes).
- If the same test set is used repeatedly for model development and parameter optimization it is likely that the resulting model is overfitted for a particular test set and will perform poorly for other instances.
- There is a tradeoff between training and test set sizes: Large training sets improve the model performance, but large test sets improve the accuracy of validation results.

We will argue later in section 7.2.3 that the inclusion of applicability domains in validation results will resolve some of these problems. To enable accurate performance indicators for smaller datasets *cross-validation* techniques have been developed. The complete dataset is divided into n folds. Each fold serves once as test set for a model based on the remaining $n - 1$ folds. With this procedure it is possible to obtain unbiased predictions for all compounds of the original dataset. It is however important to repeat feature selection and parameter optimisations within each cross-validation fold.

7.1.3 External validation sets

The “gold standard” to evaluate model performance is to determine the endpoint experimentally and compare the results with predictions. In this case it is impossible to cheat voluntarily or involuntarily or to use information about the test set distribution for model development. External validation sets share however two important limitations with other test sets

- The validation results depend to a large extent on the test set composition and on the fraction of compounds within the applicability domain of the model
- Validation results with large test set are more reliable than results from small test sets. As a rule of thumb, test sets should contain at least 30 compounds.

7.2 Performance measures

The following discussion of performance measures assumes that a validation set \mathbf{X} of size n has been predicted and the goal is to assess the predictive power of the model.

7.2.1 Classification

We assume two-fold classification, i.e. $f(\mathbf{x}_i)$ and y_i can only take two possible values, e.g. *active* and *inactive* for all $\mathbf{x}_i \in \mathbf{X}$. The simplest measure for the potential of the model to differentiate between right and wrong predictions is Precision. It is defined as the ratio of correct predictions with respect to a certain confidence threshold ad as

$$prec(ad) = \frac{|\{\mathbf{x}_i \mid f(\mathbf{x}_i) = y_i \wedge c_i > ad\}|}{|\{\mathbf{x}_i \mid c_i > ad\}|}, \quad (11)$$

where c_i is the confidence for prediction i .

A counting statistic can be obtained in a contingency table that counts classifications based on the predicted and database activity. When this data is combined with the confidence values c_i obtained from applicability domain estimation (see section 6), a ROC (Receiver-Operating Characteristic) curve [35] can be generated. For every confidence threshold ad it is possible to calculate the True Positive Ratio and False Positive Ratio as

$$\begin{aligned} tpr(ad) &= \frac{|\{\mathbf{x}_i \mid f(\mathbf{x}_i) = \text{active} \wedge y_i = \text{active} \wedge c_i > ad\}|}{|\{\mathbf{x}_i \mid y_i = \text{active} \wedge c_i > ad\}|}, \quad \text{and} \\ fpr(ad) &= \frac{|\{\mathbf{x}_i \mid f(\mathbf{x}_i) = \text{active} \wedge y_i = \text{inactive} \wedge c_i > ad\}|}{|\{\mathbf{x}_i \mid y_i = \text{inactive} \wedge c_i > ad\}|}. \end{aligned} \quad (12)$$

The true positive rate tpr indicates the *sensitivity* or *recall* of the model, i.e. how easy the model recognizes actives, and $1 - fpr$ indicates the *specificity* of the model, i.e. how robust it is against false alarms at a confidence level of ad . Plotting tpr against fpr for many possible values of ad between 0 and 1 gives the ROC curve (see Figure 6). A ROC curve shows several things. First, it demonstrates that any increase in sensitivity will be

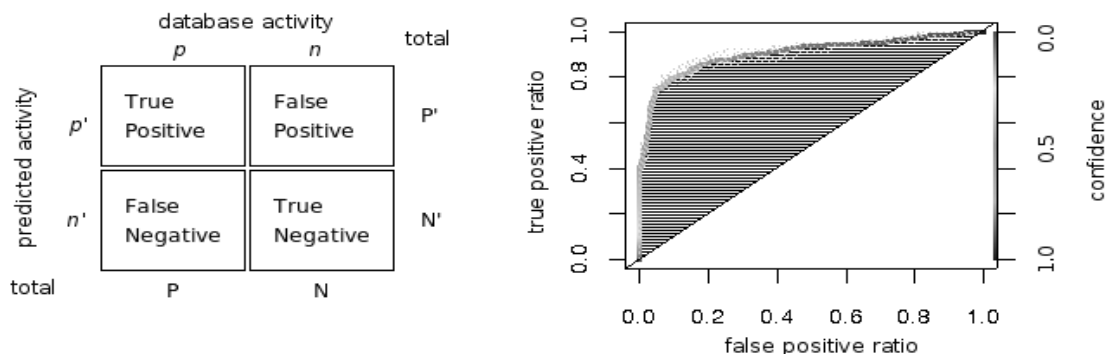


Figure 6: An example contingency table and ROC curve

accompanied by a decrease in specificity, i.e. there is a tradeoff between the two. Second, the closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the model, and the closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the model. Furthermore, the slope of the tangent line at a specific confidence threshold gives the likelihood ratio for that confidence value of the model. Finally, the area between the curve and the diagonal is a measure of model accuracy. This is a very valuable and usable parameter because it is non-parametric, i.e. assumes no specific data distribution.

ROCR, a rather powerful library for ROC analysis which is able to generate a wealth of performance measures for classification is available for R [36].

7.2.2 Regression

Choosing a performance measure for regression, i.e., when predicting quantitative values, is not so easy because a counting statistic is not available. A straightforward and non-parametric measure is the mean squared error. It is defined as

$$mse = \frac{1}{n} \sum_{i=1}^n (y_i - f(\mathbf{x}_i))^2.$$

The mean squared error should always be calculated as an unambiguous performance measure. This quantity is however sensitive to the overall scale of the target values, and it makes sense to normalize by the variance of the training activities to obtain the standardized mean squared error ($smse$).

If the sample size is not small (i.e. > 30) and the data is normally distributed the degree of correlation between predicted and database activities can be measured with r^2 , the squared correlation coefficient. For two normally

distributed variables F and Y the correlation coefficient is defined as

$$r(F, Y) = \frac{\text{cov}(F, Y)}{\sqrt{\sigma_F^2 \sigma_Y^2}}, \quad (13)$$

where σ_F^2 is the variance of F and σ_Y^2 is the variance of Y . More common than r is r^2 , the square of r . It can be interpreted as the proportion of the variance explained by the model.

Generally, the higher r^2 the better the fit of the model, because r^2 describes how well a linear approximation would fit the plot of pairs of Y and F . However, r^2 can only be applied if the two variables are normally distributed [37, 9], and this has to be verified in every case unless non parametric alternatives are used. Acceptable values for (Q)SAR models are $r^2 \geq 0.64$ ($r \geq 0.8$) [38].

Because of the variability of experimental results it has been argued that fraction of predictions within one log unit of error (assuming that the data is log-transformed) is "acceptable and closer to regulatory needs" than correlation coefficients [6]. This way the ease of counting statistics is regained and it is possible to perform ROC analysis. Correlation coefficients may be also difficult to interpret for non-statisticians (see Figure 7).

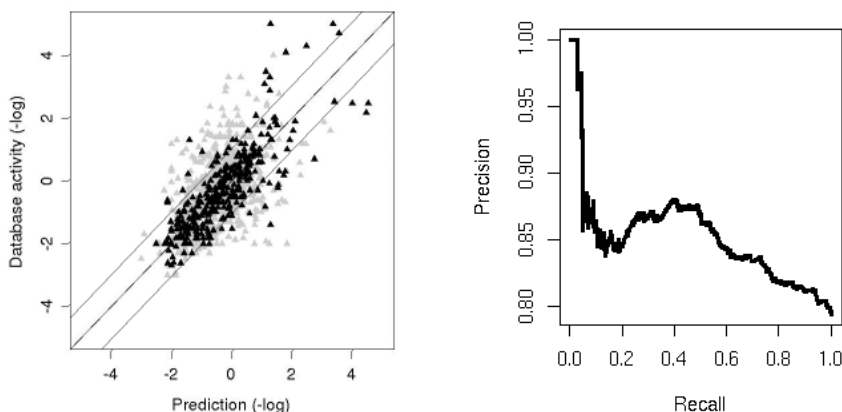


Figure 7: Left: predicted vs. database activities for the FDAMDD dataset [39] obtained by leave-one-out crossvalidation. Compounds within the applicability domain are drawn black, the rest gray. The error limit of one log unit is depicted as parallels to the diagonal. Right: Precision vs. Recall with regard to the error limit (unpublished material by the authors).

7.2.3 Impact of applicability domains on validation results

The purpose of applicability domains is to discriminate reliable from unreliable predictions. For individual predictions a confidence value can indicate the distance to the applicability domain and the expected quality of the prediction. The actual confidence values depend on the composition of the training set and on the query structure. For an easier interpretation of results a cutoff for acceptable confidence indices (and thus an applicability domain with fixed borders) can be introduced.

Validation results depend to a large extend on the test set composition and on the fraction of test compounds within the applicability domain. To compensate for this effect it is advisable to use only the test set compounds within the applicability domain for model validation, which gives more consistent validation results [40]. Another alternative would be to weight individual predictions with their associated confidence index.

If a counting statistic is available, i.e. a classification of predictions into correct and wrong, a very simple tool related to ROC analysis is Cumulative Accuracy (ca). For the k predictions with the highest confidence, calculate:

$$ca = \frac{\sum_{i=1}^n (c_i) * \delta_i}{\sum_{i=1}^n c_i} \quad (14)$$

where $\delta_i = 1$ if prediction i is correct and $\delta_i = 0$ else, and c_i is the confidence of prediction i . This calculates the confidence-weighted correct prediction ratio and removes the bias induced by high confidence values from precision (see section 7.2.1).

7.3 Mechanistic Interpretation

Many (Q)SAR and data mining techniques can be used to derive hypothesis about biological mechanisms. It is however important to remember that most of these techniques have no knowledge about chemical and biological processes. Thus they cannot reason about mechanisms, but they can provide information that is relevant for a mechanistic assessment (e.g. structural alerts, compounds with similar modes of action). This means that a toxicological researcher has to evaluate only a limited number of possible hypothesis, but expert knowledge is still needed for the identification of mechanisms.

The interpretability of models and individual predictions may depend on several factors:

Model complexity Interpretability decreases with model complexity and abstraction level, but complex models are frequently needed to accommodate for real world situations. It is however not always necessary to interpret complete models. The extraction of specific information (e.g. relevant substructures/properties) and the inspection of rationales for individual predictions may provide more information for toxicologists than complete models.

Biological rationale for the algorithm Most scientists find it easier to interpret models that have a biological rationale and/or resembles their way of thinking about toxicological phenomena. Techniques based on chemical similarities are very useful in this respect, because they support the search for analogs and chemical classes. Mechanistic hypothesis (and a critical evaluation of individual predictions) can be obtained from the inspection of relevant features and from the mechanisms of structurally similar compounds.

Visual presentation of the results Most data mining programs are hard to use for non-data mining experts and have great shortcomings in the visual presentation of their results. End-users with a toxicological background should not be confused with data mining/(Q)SAR terminology and detailed options for algorithms and parameter settings. The interface should provide instead an intuitive and traceable presentation of the rationales for a prediction together with links for the access of supporting information (e.g. original data, results in other assays, literature).

8 Conclusion

The most frequent application of data mining in predictive toxicology is the development of (Quantitative)Structure-Activity Relationship ((Q)SAR) models. The development of (Q)SAR models requires (i) the generation of features that represent chemical structures, (ii) the selection of features for a particular endpoint, (iii) the development of a (Q)SAR model, (iv) the validation of the model and (v) the interpretation of the model and of individual predictions.

For each of these tasks a large number of data mining techniques are available. Selecting and combining suitable algorithms for the individual steps allows us to develop problem specific solutions with capabilities that go beyond standardized solutions. It is however important to understand the properties and limitations of the applied techniques and to communicate them clearly to the model users.

References

- [1] Richard AM: **Commercial toxicology prediction systems: a regulatory perspective.** *Toxicol Lett* 1998, **102-103**:611-616.

-
- [2] Richard AM, Williams CR: *Quantitative Structure-Activity Relationship (QSAR) Models of Mutagens and Carcinogens*, CRC Press 2003 chap. 5.
- [3] Witten IH, Frank E: *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco 2005, [<http://www.cs.waikato.ac.nz/ml/weka/>].
- [4] Guha R: **Chemical Informatics Functionality in R**. *Journal of Statistical Software* 2007, **18**(5), [<http://www.jstatsoft.org/v18/i05>].
- [5] Team RDC: *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria 2007, [<http://www.R-project.org>]. [ISBN 3-900051-07-0].
- [6] Benigni R, Bossa C, Netzeva T, Worth A: *Collection and Evaluation of (Q)SAR Models for Mutagenicity and Carcinogenicity*, European Commission Joint Research Centre 2007 chap. 4.1.
- [7] Pavan M, Netzeva T, Worth A: **Validation of a QSAR model for acute toxicity**. *SAR and QSAR in Environmental Research* 2006, :147–171.
- [8] Guha R, Howard MT, Hutchison GR, Murray-Rust P, Rzepa H, Steinbeck C, Wegner JK, Willighagen EL: **The Blue Obelisk—Interoperability in Chemical Informatics**. *Journal of Chemical Information and Modeling* 2006, **46**:991–998.
- [9] Crawley MJ: *Statistics: An Introduction using R*. Wiley 2005.
- [10] Guha R, Serra JR, Jurs PC: **Generation of QSAR sets with a self-organizing map**. *J Mol Graph Model* 2004, **23**:1–14.
- [11] Jolliffe IT: *Principal Components Analysis*, Springer 2002 .
- [12] Yoav B, Yekutieli D: **The Control of the False Discovery Rate in Multiple Testing Under Uncertainty**. *The Annals of Statistics* 2001, **29**:1165–1188.
- [13] Pollard KS, Dudoit S, van der Laan MJ: **Multiple Testing Procedures: R multtest Package and Applications to Genomics**. *U.C. Berkeley Division of Biostatistics Working Paper Series* 2004, **164**.
- [14] Russell SJ, Norvig P: *Artificial Intelligence: A Modern Approach (2nd Edition)*. Prentice Hall 2002.
- [15] Franke R, Gruska A: *Quantitative Structure-Activity Relationship (QSAR) Models of Mutagens and Carcinogens*, CRC Press 2003 chap. 1.
- [16] Papa E, Villa F, Gramatica P: **Statistically Validated QSARs, Based on Theoretical Descriptors, for Modeling Aquatic Toxicity of Organic Chemicals in Pimephales promelas (Fathead Minnow)**. *J. Chem. Inf. Model.* 2005, :1256–1266.
- [17] Eldred DV, Weikel CL, Jurs PC, Kaiser KL: **Prediction of Fathead Minnow Acute Toxicity of Organic Compounds from Molecular Structure**. *Chem. Res. Toxicol.* 1999, :670–678.
- [18] Serra JR, Jurs PC, Kaiser KL: **Linear regression and computational neural network prediction of tetrahymena acute toxicity for aromatic compounds from molecular structure**. *Chem Res Toxicol* 2001, **14**(11):1535–1545.
- [19] Chang CC, Lin CJ: *LIBSVM: a library for support vector machines* 2001. [Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>].
- [20] Mitchell TM: *Machine Learning*. Columbus, OH: The McGraw-Hill Companies, Inc. 1997.
- [21] Rasmussen CE, Williams CKI: *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press 2005.
- [22] Schölkopf B, Smola AJ: *Learning with Kernels*. MIT Press 2002.
- [23] Bellman R: *Dynamic Programming*. Princeton University Press, Princeton, NJ 1957.

- [24] Holliday J, Hu C, Willett P: **Grouping of coefficients for the calculation of inter-molecular similarity and dissimilarity using 2D fragment bit-strings.** *Comb Chem High Throughput Screen* 2002, **5**(2):155–66.
- [25] Helma C: **Lazy Structure-Activity Relationships (lazar) for the Prediction of Rodent Carcinogenicity and Salmonella Mutagenicity.** *Molecular Diversity* 2006, :147–158.
- [26] Atkeson CG, Moore AW, Schaal S: **Locally Weighted Learning.** *Artificial Intelligence Review* 1997, **11**(1-5):11–73.
- [27] Helma C, Kramer S, De Raedt L: **The Molecular Feature Miner MolFea.** *Proceedings of the Beilstein-Institut Workshop* 2002.
- [28] Rückert U, Kramer S: **Frequent Free Tree Discovery in Graph Data.** *Proc. of the ACM Symposium on Applied Computing (SAC 2004)* 2004, **164**:1165–1188.
- [29] Yan X, Han J: **gSpan: Graph-based substructure pattern mining** 2002, :721–724.
- [30] Nijssen S, Kok JN: **The Gaston tool for Frequent Subgraph Mining.** *Electronic Notes in Theoretical Computer Science* 2004.
- [31] Ralaivola L, Swamidass SJ, Saigo H, Baldi P: **Graph kernels for chemical informatics.** *Neural Netw.* 2005, **18**(8):1093–1110.
- [32] Jaworska JS, Comber M, Auer C, Van Leeuwen CJ: **Summary of a workshop on regulatory acceptance of (Q)SARs for human health and environmental endpoints.** *Environ Health Perspect* 2003, **111**(10):1358–1360. [Congresses].
- [33] Jaworska J, Nikolova-Jeliazkova N, Aldenberg T: **QSAR applicability domain estimation by projection of the training set descriptor space: a review.** *Altern Lab Anim* 2005, **33**(5):445–459.
- [34] Zhang S, Golbraikh A, Oloff S, Kohn H, Tropsha A: **A Novel Automated Lazy Learning QSAR (ALL-QSAR) Approach: Method Development, Applications, and Virtual Screening of Chemical Databases Using Validated ALL-QSAR Models.** *J. Chem. Inf. Model.* 2006, **46**:1984–1995.
- [35] Egan JP: *Signal Detection Theory and ROC Analysis.* New York: Academic Press 1975.
- [36] Sing T, Sander O, Beerenwinkel N, Lengauer T: **ROCR: visualizing classifier performance in R.** *Bioinformatics* 2005, **21**(20):3940–3941.
- [37] Anscombe FJ: **Graphs in statistical analysis.** *American Statistician* 1973, **27**:17–21.
- [38] Cronin MT, Livingstone DJ: *Predicting Chemical Toxicity and Fate.* CRC Press 2004.
- [39] Matthews EJ, Kruhlak NL, Benz RD, Contrera JF: **Assessment of the health effects of chemicals in humans: I. QSAR estimation of the maximum recommended therapeutic dose (MRTD) and no effect level (NOEL) of organic chemicals based on clinical trial data.** *Curr Drug Discov Technol* 2004, **1**:61–76.
- [40] Benigni R, Netzeva TI, Benfenati E, and R Franke CB, Helma C, Hulzebos E, Marchant C, Richard A, Woo YT, Yang C: **The expanding role of predictive toxicology: an update on the (Q)SAR models for mutagens and carcinogens.** *J Environ Sci Health C Environ Carcinog Ecotoxicol Rev.* 2007, **25**:53–97.