Regression and classification of biochemical systems using the DemPRED library

Dissertation zur Erlangung des akademischen Grades des

Doktors der Naturwissenschaften (Dr. rer. nat.)

eingereicht im Fachbereich Biologie, Chemie, Pharmazie

der Freien Universität Berlin

vorgelegt in englischer Sprache von

Özgür Demir

aus Bielefeld

Dezember, 2011

Diese Dissertation wurde von der Deutschen Forschungsgemeinschaft im Rahmen des Internationale Research Training Group (IRTG) finanziert und am Institut für Chemie und Biochemie der Freien Universität Berlin in englischer Sprache verfasst.

1. Gutachter: Prof. Dr. Ernst-Walter Knapp

2. Gutachter: Prof Dr. Gerhard Wolber

Disputation am 04.04.2012

Preamble

This cumulative thesis is the sum of my research work in which a novel machine learning library has been created and used for various biological prediction tasks. This thesis is based on the following four peer-reviewed journal publications:

Demir-Kavuk, O., Bentzien, J., Muegge, I., Knapp, E.W.,
Predicting human volume of distribution and clearance of drugs using automated feature selection
J Comput Aided Mol Des., 25 (2011), Nr. 12, p.1121-1133,
http://dx.doi.org/DOI:10.1007/s10822-011-9496-z

Demir-Kavuk, O., Kamada, M., Akutsu, T., Knapp, E.W.,
Prediction using step-wise L1, L2 regularization and feature selection for small data sets with large number of features
BMC Bioinformatics, 12 (2011), p. 412, http://dx.doi.org/DOI:10.1186/1471-2105-12-412

Demir-Kavuk, O., Riedesel, H., and Knapp, E. W.,
Exploring classification strategies with the CoEPrA 2006 contest
Bioinformatics, 26 (2010), Nr. 5, p. 603-609,
http://dx.doi.org/DOI:10.1093/bioinformatics/btq021

Demir-Kavuk, O., Krull, F., Chae, M. H., and Knapp, E. W.,
Predicting Protein Complex Geometries with Linear Scoring Functions
Genome Informatics, 24 (2010), p. 21-30

During my PhD research, additionally the following paper has been published, which makes use of the DemPred library for empirical prediction of Pka values:

Gamiz-Hernandez, A. P., Kieseritzky, G., Galstyan, A. S., **Demir-Kavuk, O.**, and Knapp, E. W., *Understanding properties of cofactors in proteins:redox potentials of synthetic cytochromes b*
Chemphyschem., 11 (2010), Nr. 6, p. 1196-206

Acknowledgements

I would like to thank Prof. Dr. Ernst Walter Knapp for his valuable support. I would also like to thank my colleagues and friends of the AG Knapp for fruitful discussions. Furthermore, I would like to thank my family and especially Antonia Weßel for their encouragement and ongoing support. Last but not least, I would like to thank Ingo Mügge and Jörg Bentzien for their cooperation on the VD_{ss} publication. The following work was funded by the International Research Training Group (IRTG) on "Genomics and Systems Biology of Molecular Networks" (GRK1360, Deutsche Forschungsgemeinschaft (DFG)).

Table of Content

1 Introduction

Biological systems are very complex. It is therefore often not possible to predict a certain molecular property such as binding affinity just by examining the single relevant molecular system. Predicting properties of such a system in a wet lab on the other hand can be a very time consuming and laborious task. Hence, in order to save time and money machine learning methods can help to predict these molecular properties *in silico*. Nowadays, empirical methods of machine learning are widely used in life sciences and related sciences such as chemistry, biochemistry, pharmacy, and medicinal diagnostics. The advantages are many fold as computational methods are fast, cheap and therefore applicable as a large throughput method. At the moment of writing, most of the computational predictors available are not as accurate as direct measurements in a wet lab. Nevertheless, even if computational predictions are not of perfect quality they still can be very helpful. If for example an activity prediction is biased in some way it still can help to discriminate molecules with large from those with small activities. For such problems the computational predictor can serve as a pre-filter to reduce the number of candidate compounds to be tested in a wet lab. Machine learning methods may also be used as a reverse engineering approach. In order to do so, an initial model for a certain target property in focus is automatically generated. Most machine learning methods will automatically focus only on relevant aspects of the problem. Hence, examining the structure of the resulting computational model may give a detailed insight into the underlying biochemical process. With this advanced knowledge molecule structures may directly be modified in order to amplify or weaken a certain molecular property.

Developing computational prediction models from scratch requires good programming skills and may be a time consuming task. Hence, there is a demand for powerful yet easy to use libraries, which users can employ and extend to build their own models given a particular classification/regression task. During my PhD I developed such a library called DemPRED. DemPRED is a platform independent JAVA library which includes many routines which can be freely combined in order to generate prediction models.

I used the DemPRED library for various classification and regression tasks such as predicting major histocompatibility complex II (MHC II) epitopes, prediction of human volume of distribution and clearance as well as detecting protein interface regions. The predictive power of

all generated models was as good as or even better than other state of the art classification and regression techniques.

The following last part of the introduction briefly describes the core techniques of DemPRED. These techniques were used to build all of the above mentioned prediction models and are common for many other prediction tasks.

1.1 Datasets

In order to build computational models for objects (molecules) with a certain target property, training datasets of molecules with experimentally measured target values are needed. For a two class classification problem, e.g. discriminate between molecules that are binders or non-binders, the target values are set to +1 or -1, respectively. Ideally, the number of positive and negative training data should be of similar size, but in case they are not a suitable correction to balance the data can be applied (see “Classification of Unbalanced Data” on page 14). If the measured target values are continuous the correlation between the molecules and their target values are established by regression analysis [4]. In such a case the training set should ideally cover the whole range of possible target values. It is obvious that an ideal training set should not contain any measurement errors. However, due to the non-availability of pre-compiled training sets many published models are learned on self-compiled data sets extracted from the literature. Since experimental conditions and the laboratories, where target values were measured, vary for different publications, the resulting data sets may be biased and therefore yield biased prediction models.

1.2 Feature Extraction

The second step in machine learning consists of transforming the considered real world objects of the training set into a multidimensional computer-readable representation. This is done by extracting so-called feature vectors. A feature vector is a d-dimensional vector whose components contain numerical values of specific features (descriptors) of the considered objects (molecules). For an image for example the feature values might correspond to the RGB values of that image. For a molecule topological or physicochemical descriptors may be used. Thus, each molecule in the regarded dataset can be expressed by a point in a d-dimensional space called feature space.

The descriptors (feature vectors) are the eyes with which the machine learning approach observes the objects. Practically all machine learning methods will achieve good prediction re-

sults, if the extracted features strongly correlate with the target property in focus. If on the other hand the features do not contain any information on the target property (e.g. numerical representation of the compound trade names) even the best machine learning technique will fail. The feature generation step is thus the most important part of model building and has the largest influence on the predictive power. In most cases, neither ideal nor totally irrelevant descriptors will be chosen. In such a case, the machine learning method has to detect only those descriptors, which are relevant for the predicted property and disregard the remaining ones. The choice of the machine learning approach may therefore also have a large influence on the resulting model.

1.3 Linear Scoring Function

Describing molecules by real valued descriptors collected in vectors $\vec{x}_i \in \mathbb{R}^d$ allows their representation as points in a d-dimensional feature space. In a general two-class classification approach, a hypersurface is defined that separates the data points into two half-spaces such that all positive data points (target value +1) are located on one side of the hypersurface whereas all negative data points (target value −1) are located on the other side. New data points are then classified according to the half-space they belong to (see Figure 1). In the simplest case the separating hypersurface is a hyperplane in the d-dimensional feature space. In a regression task a hyperplane is constructed where the distances of the data points to the hyperplane are proportional to their target property values. The target property of a new molecule can then be predicted by computing its proportional distance to the hyperplane (see Figure 1).

To determine such a hyperplane with normal vector $\vec{w}$ and offset *b* from the origin we set up an objective function *L* that is minimal for the optimal hyperplane:

$$L(\vec{w},b) = \underbrace{\frac{1-\sum_p \lambda_p}{N} \sum_{i=1}^{N} \left[\mu_i \, g\left(f\left(\vec{x}_i;\vec{w},b\right), m_i \right) \right]}_{\text{model terms}} + \underbrace{\lambda_p \, || \, \vec{w} \, ||_p^p}_{\text{regularization terms}} \quad , \tag{1}$$

where m_i are the target properties and $g(f, m_i)$ is a loss function that determines the stiffness of the correlation between the property values m_i and the linear scoring function $f(\vec{x}_i;\vec{w},b)$:

$$f(\vec{x}_i;\vec{w},b) = \vec{w}^t \bullet \vec{x}_i + b \, . \tag{2}$$

The linear scoring function is also used to predict the target values of new unseen data points once a hyperplane is defined. In a classification scenario, a data point is classified as positive

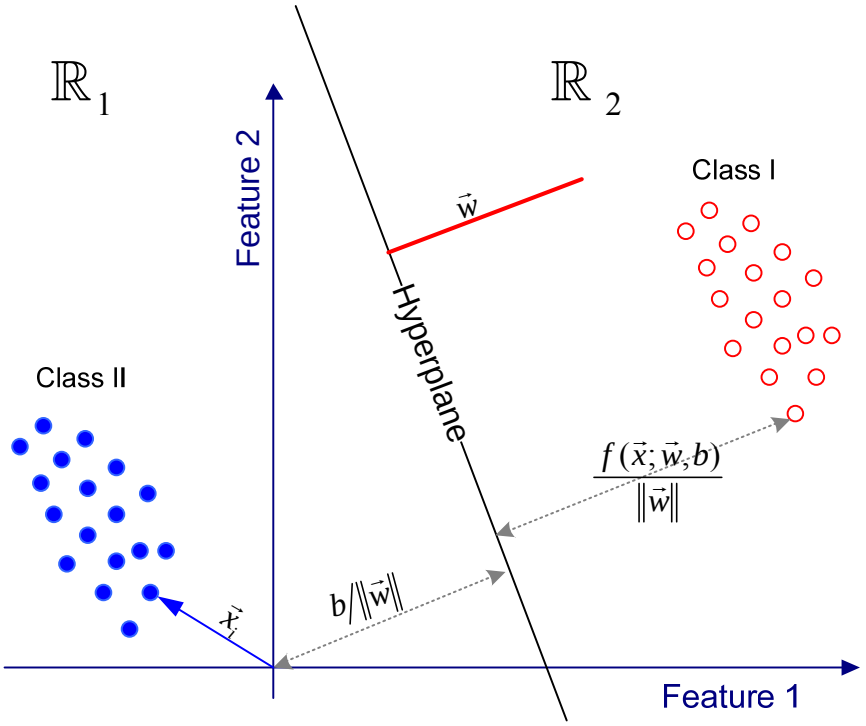

Figure 1: Hyperplane with normal vector $\vec{w}$ and offset b. The hyperplane separates the feature space into two half spaces $\mathbb{R}_1^d$ and $\mathbb{R}_2^d$. Each half space represents a class. The distance from the hyperplane to the origin is given by $b/\|\vec{w}\|$. In a regression analysis the distance from a sample vector $\vec{x}_i$ to the hyper plane is given by $f(\vec{x}_i;\vec{w},b)/\|\vec{w}\|$.

if the result of eq. (2) is larger than 0 otherwise it is classified as negative. In a regression task the result of eq. (2) directly represents the desired target value. The additional parameters μ_i in the objective function, eq. (1), can be used to weight the more reliable data points higher than others.

The objective function *L*, eq.(1), consists of two parts that compete with each other. The first part involves the so called 'model terms'. These terms optimize the prediction performance on the training set called recall performance: whenever the considered model returns a poor prediction on the training set the loss-function $g(f_i,m_i)$ invokes a penalty that depends on the error margin. Hence, during learning the hyperplane parameters $(\vec{w},b)$ will be chosen such that predictions on the training set are as close as possible to their experimentally measured property values.

For most prediction tasks there is little knowledge of the underlying biochemical process available. Hence, it is not clear which features to include into the model building process.

With no further knowledge, generally all available descriptors are considered. This results in a high-dimensional feature space. On the other hand, for many problems, where molecular target values need to be predicted by empirical machine learning methods, the amount of data (number of compounds) is often scarce. Hence, in a typical prediction scenario the number of compounds with known target values can be very small (e.g. 100 or even smaller). The number of potentially relevant features on the contrary is often large (e.g. 1000 or even larger). Machine learning methods tend to over-fit the training data in such situations, i.e. the method adjusts to very specific features of the training data, which are not characteristic for the considered property. To control this effect, the objective function, eq.(1), is usually extended by a so called regularization term of positive weight:

$$\lambda_p \| \vec{w} \|_p = \lambda_p \left(\sum_{i=1}^{N} |w_i|^p \right)^{1/p} . \quad (3)$$

This regularization term adopts its minimum value, if all components of the parameter vector $\vec{w}$ vanish. This is in conflict with the 'model terms' of the objective function, which requires specific non-vanishing model parameters. The trade-off is that the model parameters governing the less important features are set to small or ideally vanishing values, while model parameters referring to features that exhibit strong correlations with the target values are kept. Hence, the regularization term penalizes model details of unnecessary complexity, focuses on the most relevant features, and thus avoids over-fitting of the data used for training [5]. The most commonly used regularization methods are L1 regularization (p = 1), also known as Lasso [6] and L2 regularization (p = 2) also known as ridge regression [7].

1.4 Feature Selection

Due to the form of the objective function, eq.(1), features whose corresponding model parameters vanish are completely ignored. The quadratic form of the L2 regularization term will not lead to model weights set exactly to zero during learning. On the other hand, L1 regularization leads to sparse models due to its linear form where the weights of many features are set to zero rigorously. L1 regularization therefore combines feature selection and model building into a single training round. Nevertheless, the quadratic L2 regularization can also be used for explicit feature selection. Assuming unimportant features will get smaller parameter weights than important ones, a backward selection can be performed by removing features with small weights after each training round. This approach is called RFE (recursive feature elimination) [8]. In spite of its simplicity, the RFE algorithm yields excellent results and has been success-

fully used in many classification and regression tasks [8-11]. Additional feature selection strategies include simple filters, e.g. correlation between target value and single features, and more complex strategies such as the sequential floating search method [12], genetic algorithms [13], and swarm intelligence algorithms such as ant colony optimization [14, 15].

1.5 Loss Functions

The loss function $g(f_i, m_i)$ determines the stiffness of the correlation between the measured and predicted target values. Figure 2 gives an overview of possible loss functions. For regression analysis the ideal model should predict the training data as accurate as possible. Hence, loss functions which punish deviations from the property value in both directions evenly are preferred. Possible loss functions include the squared and Lorentzian loss function (see Figure 2). Due to its quadratic form the squared loss function is best suited for datasets with unbiased measurements. Datasets, which may contain outliers, should be trained using a Lorentzian loss function as the impact of outliers is reduced. Both squared error and Lorentzian loss function can also be used for classification tasks. However, for classification, the exact result of the scoring function is not of relevance. It is only of interest that the result is above or below

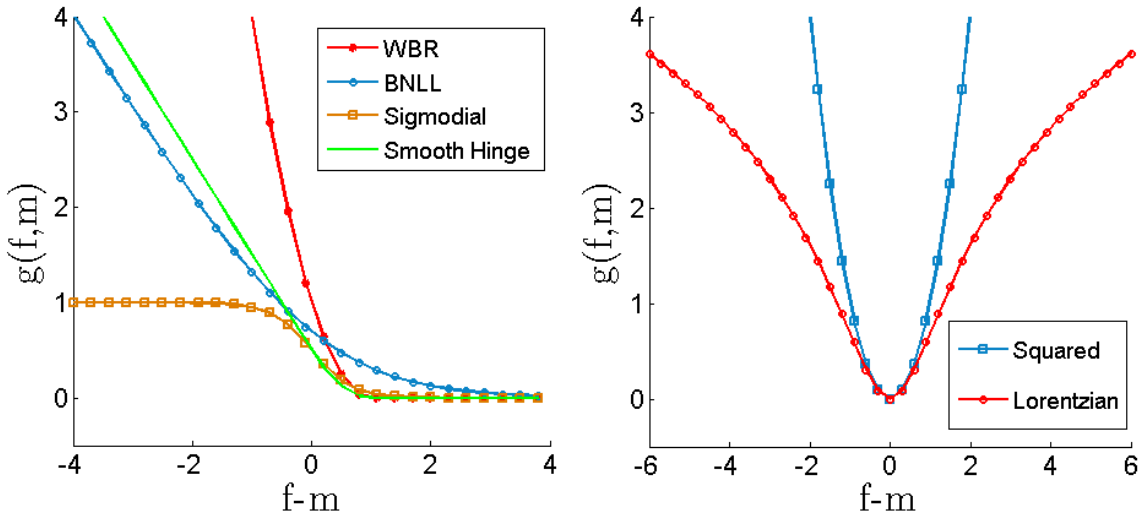

Figure 2: Different loss functions $g(f, m)$ as a function of the difference $f - m$ between the value of the scoring function f estimating the property value and the corresponding true property value *m*. Left side: one-sided loss functions, which can be used for two-class classification tasks. Right side: symmetric loss functions, which can be used for classification and regression tasks.

the threshold at zero. Data with positive target values classified as strongly positive and data with negative target values classified as strongly negative do not have to be punished. Hence, better results may be achieved with loss functions, which only punish deviations in one direction. For that purpose loss functions such as the Smooth Hinge, Sigmodial, Binomial Log Likelihood (BNLL) and Weighted Biased Regression (WBR) loss functions may be used, which mostly differ in the way they treat outliers (see Figure 2).

1.6 Non-linear Models

For most biochemical prediction tasks there are many more descriptors available than there are data points. Hence, a linear model will most often be the best choice. However, there may be problems where a linear model is not sufficiently flexible to describe the studied data. In such a case, a non-linear transformation of the original model data into a feature space of higher dimension may render the dataset more suitable for a linear separation. This corresponds to a non-linear separation in the original feature space using a more general hypersurface instead of a hyperplane. Nevertheless, an explicit transformation of the dataset of compounds may computationally be too expensive or even intractable. Instead of transforming the compound data explicitly. the kernel trick [16] transforms the data implicitly. For that purpose, the objective function *L*, eq.(1), is rewritten such that the parameter vector $\vec{w}$ can be expressed as a weighted sum of the training feature vectors $\vec{x}_i^t$.

$$\vec{w} = \sum_{i=1}^{N} \alpha_i \vec{x}_i^t \,, \tag{4}$$

with

$$\alpha_i = -\frac{(1-\lambda_2)\mu_i}{N2\lambda_2} \frac{\partial g(f(\vec{w},b;\vec{x}_i),m_i)}{\partial f} \,. \tag{5}$$

Hence, the linear scoring function, eq. (2), can be rewritten as

$$f(\vec{w},b;\vec{x}) = \sum_{i=1}^{N} \left(\alpha_i \vec{x}_i^t \bullet \vec{x}\right) + b = \sum_{i=1}^{N} \left(\alpha_i K\left(\vec{x}_i^t, \vec{x}\right)\right) + b \,. \tag{6}$$

Now, instead of finding the hyperplane normal $\vec{w}$, we determine the scalar multipliers α_i, eq. (5). Note that determining the multipliers as well as predicting new data points can be done solely using values of the dot products $\vec{x}_i^t \bullet \vec{x}$. These dot products may be replaced by a kernel

function $K(\vec{x}_i^t, \vec{x})$. The simplest kernel function consists of the dot product $K_{linear}(\vec{x}_i^t, \vec{x}) = \vec{x}_i^t \bullet \vec{x}$. However, using higher order kernels transforms the linear model into a non-linear model [17, 18]. The choice of kernel function strongly depends on the prediction task. There are numerous specialized kernel functions available, such as graph kernels, which are well suited for graph based problems (e.g. topological compound structures) [19] or String kernels which may be used for DNA or amino acid sequences [20]. Other widely used kernel functions include polynomial kernel, radial basis function (RBF), and sigmodial kernel [17, 18].

1.7 Classification of Unbalanced Data

For a classification task problems can arise if the sizes of the two classes available for learning are very different. To avoid false positives for the majority class, it can be advantageous to split the N^+ positive $\vec{x}_i^+$ from the N^- negative $\vec{x}_j^-$ data ($N = N^+ + N^-$) leading to the balanced objective function:

$$L_{balanced} = \left(1 - \sum\nolimits_p \lambda\right) \sum_{s=+,-} \left[\frac{\delta^s}{N^s} \sum_{i=1}^{N^s} \mu_i g\left(f\left(\vec{x}_i^s; \vec{w}, b\right), m_i^s \right) \right] + \lambda_p \| \vec{w} \|_p^p , \tag{7}$$

where N is the size of the data set and w^+ and w^- ($w^+ + w^- = 1$) are the weights for the positive and negative data, respectively.

2 Publications

2.1 Predicting human volume of distribution and clearance of drugs using automated feature selection

Authors **Demir-Kavuk, O.**, Bentzien, J., Muegge, I., Knapp, E.W.,

Bibliography J Comput Aided Mol Des., 25 (2011), Nr. 12, p. 1121-1133
http://dx.doi.org/DOI:10.1007/s10822-011-9496-z

Contribution

- Development of the research question
- Development of the required software
- Development of the webpage
- Generation and analysis of the results
- Manuscript preparation

In this paper, a specialized standalone version of *DemPRED* called *DemQSAR* has been published. *DemQSAR* combines feature generation, feature selection and model building into one platform independent JAVA application. In contrast to the *DemPRED* library, no additional programming is needed. The user just has to provide 2-D structures of compounds together with an experimentally measured target property in order to generate prediction models. The *DemQSAR* application has been developed in cooperation with Boehringer Ingelheim pharmaceuticals. Our cooperation partners as well as other pharmaceutical companies have the demand to build interpretable models using several thousand descriptors per compound. Hence, great focus has been led on easy to use feature selection. *DemQSAR* incorporates two state of the art feature selection strategies: embedded Lasso and RFE.

The performance of *DemQSAR* has been tested by building models for the prediction of human Volume of distribution (VD_{ss}) and Clearance (CL). Volume of distribution is a measure of how a drug is distributed between plasma and tissues. Clearance is a measure of the rate at which a drug is removed from the body. Both values together determine the half-life of a drug and thus the appropriate dose and frequency of drug application. A drug should be administered such that the free plasma concentration is large enough to obtain an effect throughout the dosing interval, while lessening the maximal concentration over time by clearance and thereby reducing the potential for side effects. The generated *DemQSAR* models are able to predict human VD_{ss} and CL of an independent test set with a **g**eometric **m**ean **f**old **e**rror (GMFE) of 2.0 and 2.4 respectively:

$$GMFE = \exp\left|\frac{1}{N}\sum_{m}\ln\left(\frac{\mathrm{m}_{pred}}{\mathrm{m}_{\exp}}\right)\right| \tag{8}$$

Both prediction models were generated using two sets of features each containing around 4000 descriptors per compound. The first feature set was generated using commercial software packages such as ACD/Labs, ClogP, Volsurf, MolConn-Z, Scitegic, MOE, Pipeline Pilot. The second feature set was generated using the open-source Chemistry Development Kit (CDK) [21]. The implemented feature selection strategy was able to pick up 8 descriptors out of each feature set to build reliable models for the prediction of human $VD_{ss.}$ Figure 3 illustrates these selected features ordered by their absolute parameter value. Relative parameter values reflect the importance of a particular feature. I.e., as larger the absolute value of a parameter is compared to others the more important is its influence. For the first feature set strong positive correlations were observed for: “fraction cationic at pH 7”, “pKa MB (mostly

basic)" and "VS-descriptor LgD 10/9/8 (logarithm of the partition coefficient between 1-octanol and water)". Strong negative correlations were observed for the following interpretable feature: "fraction anionic at pH 7". Hence, the human VD_{ss} of a compound could be enhanced by increasing its cationic fraction at pH 7 and decreasing its anionic fraction at pH 7 or by increasing its octanol-water distribution coefficient, logD (hydrophobic drugs). For the second open-source feature set, strong positive correlations were observed for various substructures: 83 (Carboxylic ester), 22 (Primary aliph amine) and 25 (Quaternary aliph ammonium), while strong negative correlations were observed for TopoPSA and Khs.do. TopoPSA computes the topological polar surface area based on fragment contributions. Khs.do counts the number of double bonded oxygen atoms: O=*.

The final models for the prediction of human VD_{ss} and human CL were made accessible through an easy to use web interface [22]. In addition to the predicted VD_{ss} and CL values, 2-dimensional images, smiles codes, molecular formula and molecular weights are computed for the uploaded compounds. All results can be exported in pre-formatted Excel, text and XML files. At the moment of writing, the provided web server is to our knowledge the only publicly available resource to predict human VD_{ss} and CL.

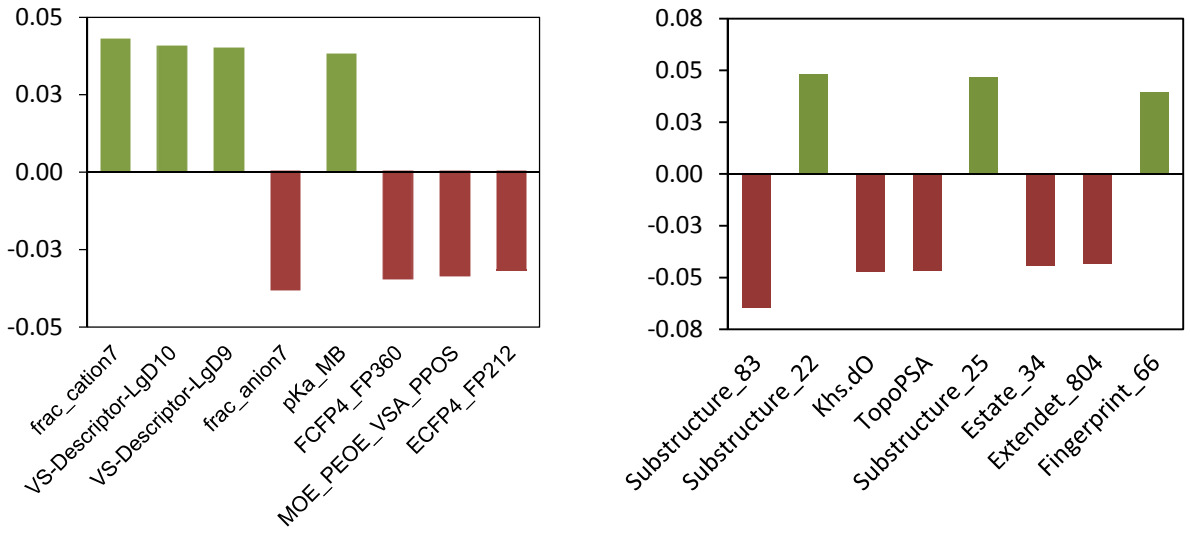

Figure 3: Parameter weights for the human VD_{ss} prediction model where eight features were selected for each feature set. As larger the absolute values of the weights are, as more important are the corresponding features. Left side: 1st feature set generated using commercial software packages. Right side: 2nd feature set generated using open-source CDK package [21].

2.2 Prediction using step-wise L1, L2 regularization and feature selection for small data sets with large number of features

Authors	**Demir-Kavuk, O.**, Kamada, M., Akutsu, T., Knapp, E.W.,
Bibliography	BMC Bioinformatics, 12 (2011), p. 412 http://dx.doi.org/DOI:10.1186/1471-2105-12-412
Contribution	• Development of the research question • Development of the required software • Generation and analysis of the results • Manuscript preparation

In recent years, considerable advancements were made in high throughput techniques to measure particular molecular properties of large compound data bases. Nevertheless, for many problems, where molecular target values need to be predicted by empirical machine learning methods, the amount of data is often scarce. This leads to machine learning problems were the number of compounds with known target values can be very small compared to the number of potentially relevant features. Under such circumstances, overtraining would be unavoidable unless specific precautions are applied to control and reduce the number of features. Various regularization and feature selection techniques have thus been presented in the past and successfully applied to numerous prediction tasks [23].

This publication proposes a new two-step learning scheme which is especially well suited for a prediction task were few data are described by many features. In the first stage L1 regularization is used to filter out redundant and irrelevant features. The remaining features are used in a second stage of model building in conjunction with L2 regularization. The proposed method has been used for the regression tasks of CoEPrA (Comparative Evaluation of Prediction Algorithms) modeling competition of 2006 [24]. The data sets of CoEPrA 2006 contain octo- and nona-peptides relevant to MHC class I binding which play an important role in the immune response of mammals. These data sets are characterized by few data (~80) described by a much larger number of features (~5000). The data sets of the CoEPrA contest are particularly valuable, since they offer the possibility to compare the own approach with a larger number of alternative approaches from different groups on equal footing.

The proposed two step learning scheme has been compared to the top performing participants of the CoEPrA contest. Table 1 shows the prediction results on the test sets for all four CoEPrA tasks in terms of coefficient of determination (q^2):

$$q^2 = 1 - \frac{\sum_m \left(m_{\exp} - m_{pred} \right)^2}{\sum_m \left(m_{\exp} - average(m_{\exp}) \right)^2} \tag{9}$$

For all four regression tasks, the number of features could be reduced drastically to about one hundredth (~50) of the initial number of features or even less. Except for one task (IV) using the features selected with L1 regularization in a subsequent second training step with L2 regularization shows better prediction performance. The prediction results of this study surpass the best performing participants of the CoEPrA contest adopting first rank for task I and second rank for task II and III. As one can see from the very low q^2 values for task III and even lower

Table 1: Prediction results of q^2 values for all four CoEPrA regression tasks using a two-step optimization procedure. First three lines display the results of the three best predictions for the different CoEPrA tasks. Stage 1: only L1 regularization is used to filter irrelevant features. Stage 2: L2 regularization is used with all features remaining after stage 1.

Rank	Task I	Task II	Task III	Task IV [a]
First	0.677	0.735	0.237	-2.578
Second	0.627	0.612	0.201	-2.560
Third	0.615	0.455	0.154	-2.561
Stage 1				
Predict	0.667	0.642	0.205	-2.573
Features [b]	50	43	56	41
Stage 2				
Predict	0.691	0.668	0.131	-2.574

[b] number of features after L1 regularization.

values for task IV, the prediction results are not significant. This is due to a lack of overlap between the target values of the training and the test set. This is an ill-defined task where machine learning methods are bound to fail. It is part of the dataset compilation step to avoid such extreme cases where training and test sets differ. These tasks therefore do not represent a real case scenario.

Our proposed method achieved good prediction results for the four CoEPrA regression tasks. Furthermore, the number of molecular descriptors has been reduced drastically for the final prediction models. The CoEPrA data sets are representative for many biological classification and regression problems where small data sets of less than hundred are described by thousands of descriptors. Hence, we expect the proposed method to be applicable for many other machine learning tasks having similar conditions.

2.3 Exploring classification strategies with the CoEPrA 2006 contest

Authors **Demir-Kavuk, O.**, Riedesel, H., and Knapp, E. W.,

Bibliography Bioinformatics, 26 (2010), Nr. 5, p. 603-609
http://dx.doi.org/DOI:10.1093/bioinformatics/btq021

Contribution

- Development of the research question
- Development of the required software
- Generation and analysis of the results
- Manuscript preparation

In silico methods to classify compounds as potential drugs that bind to a specific target become increasingly important for drug design. To build classification devices, training sets of drugs with known activities are needed. For many such classification problems not only qualitative but also quantitative information of a specific property (e.g. binding affinity) is available. The latter can be used to build a regression scheme to predict this property for new compounds. However, predicting a compound property explicitly is generally more difficult than classifying that the property lies below or above a given threshold value. Hence, the outcome of a prediction based on regression is expected to introduce larger uncertainties than solving the classification problem directly. In fact, initially researchers are only interested in classifying compounds as potential drugs. The activities of these compounds are subsequently measured in wet lab. Nevertheless, the binding affinity contains additional information, which could be of use to solve the classification problem more reliably. In this paper, we thus proposed a novel approach that uses available quantitative information directly for classification. In order to do this a new loss function called **W**eighted **B**iased **R**egression (WBR) loss function is introduced.

This new classification scheme has been tested on the classification tasks of the Comparative Evaluation of Prediction Algorithms (CoEPrA) 2006 competition [25]. Table 2 shows the prediction results using the proposed WBR loss function together with the best CoEPrA competitors. These results clearly show that the WBR classification method outperforms the best CoEPrA competitors in all three tasks. The results indicate that the proposed WBR loss function can outperform simple classification methods that do not make use of the additional quantitative information. Hence, whenever a classification is demanded, but additional quantitative information is available, we propose to use a classification scheme similar to WBR.

Table 2: MCC prediction results of the CoEPrA classification tasks 1 to 3 using the pIC50 values of the binding affinities in a classifier with Weighted Biased Regression (WBR) loss function. All values denote the MCC of the prediction set. Best results per task are printed in bold digits. The two best results of the CoEPrA contest are given in the last two columns.

Task	WBR classifier	CoEPrA first	CoEPrA second
1	**0.7759**	0.7303	0.7273
2	**0.7410**	0.7108	0.7108
3	**0.3985**	0.3560	0.3188

2.4 Predicting protein complex geometries with linear scoring functions

Authors **Demir-Kavuk, O.**, Krull, F., Chae, M. H., and Knapp, E. W.,

Bibliography Genome Informatics, 24 (2010), p. 21-30

Contribution

- Development of the research question
- Development of the required software
- Generation and analysis of the results
- Manuscript preparation

It is widely known that protein-protein interactions play a crucial role in many cellular processes. To know how and if two given proteins interact is of great interest, as this information may help to detect cellular networks or identify new drug targets. Currently, it is believed that a large number of protein complexes form only transiently. Experimental detection of the geometry of such an unstable complex is very demanding. Nevertheless, three dimensional structures of the unbound proteins are often available from crystallography or NMR experiments. Hence, various computational methods have been proposed to predict protein interface regions in atomic detail given the structures of the participating proteins [26-28]. These so called docking algorithms consist of two consecutive steps. In the first step, random complex geometries (decoys) are generated considering primarily shape complementarity of the two individual proteins [29-31]. These decoys are then evaluated by a scoring function, which discriminates approximately near-native decoys from decoys, which are far from the native complex geometry. Ideally the structures of decoys with high scores are similar to the native three dimensional structure of the formed complex.

In this publication the *DemPRED* library has been used to build such a protein interface scoring function. In order to do so, a training set consisting of 191 protein complex structures has been compiled from literature (48 from Benchmark 3.0 [32] and 143 from Huang et. al. [33]). The test set consisted of 65 protein complexes, which all were taken from literature as well (Benchmark 3.0 [32]). For all protein complexes of the training set we generated near-native decoys with a maximum interface RMSD (*iRMSD*) of $d_{max} =$ 6.0 Å, by applying random translations and rotations to one of the two proteins in the complex. Decoys of the test set were generated in a similar manner but without any *iRMSD* limitation.

Features were generated using atom-pair distance distributions, i.e. each position of the feature vector quantifies a particular atom-pair interaction within a given distance. A total of 20 different heavy-atom types [33] and two polar hydrogen atom types (hydrogen atoms making H-bonds with sulfur and oxygen or alternatively with nitrogen) have been defined [3]. Non-polar hydrogen atoms were ignored. This resulted in a total of 253 = (22*23)/2 different types of atom-pairs (features).

The predictive power of the linear scoring function approach has been compared to ZRANK [2], ZDOCK 3.0 [1] and a previously developed neural network scoring function [3]. Plots of the success rates (fraction of protein complexes with at least one *HIT* within the given number of predictions) averaged over all 65 complexes of the prediction set are shown in Figure 4. A

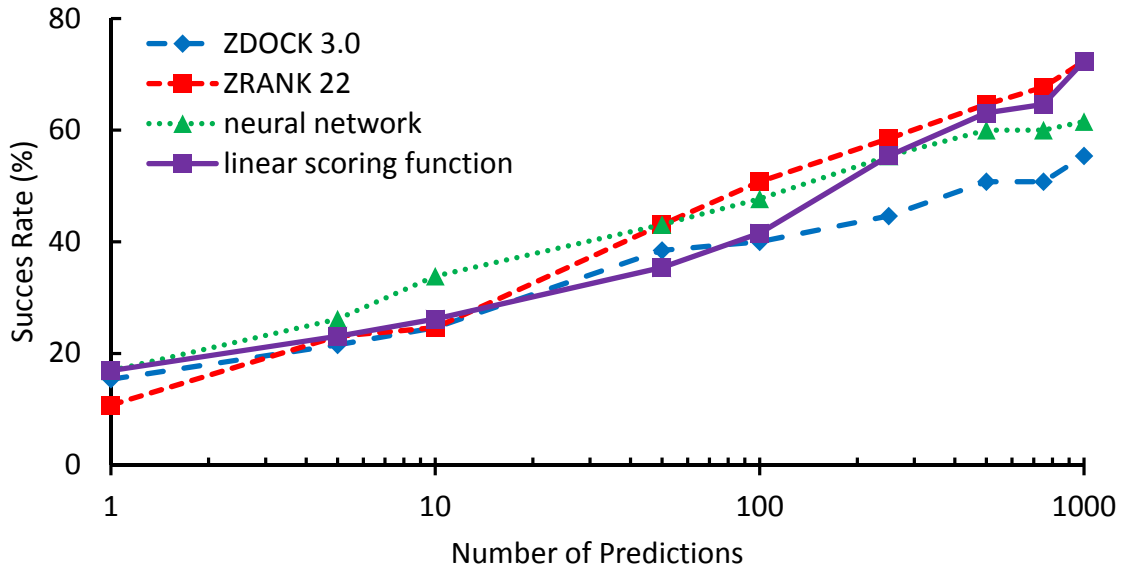

Figure 4: Comparison of the success rate versus the number of highest ranked decoys (number of predictions per protein complex) for all 65 protein complexes of the prediction set (the higher the line the better). We compare ZDOCK 3.0 [1], ZRANK [2], a neural network [3] and the linear scoring function of the present study.

HIT is thereby defined as a predicted near native decoy with an $iRMSD \leq 2.5$ Å relative to the corresponding native complex geometry. That is, the higher the line in Figure 4, the better is the result. The success rate of the linear scoring function is comparable to the three other methods. For up to ten predicted protein complexes the linear scoring function is slightly better than ZDOCK 3.0 and ZRANK but worse than the neural network. If more than 200 predictions are considered, the linear scoring function of the present study and ZRANK are the best performing models. These results indicate that the linear scoring function is comparable to other state of the art protein decoy scoring methods.

3 Discussion

During my PhD I developed a classification and regression library called *DemPRED*. *DemPRED* includes numerous routines needed for model building of any prediction problem. These include various feature generation methods for amino acid sequences such as Sparse, BLOSUM, Physicochemical and BLAST Profiles. Generation of various fingerprints as well as topological and molecular descriptors for compounds using the open source CDK software.

At the heart is a linear classification scheme, which can be combined with various loss functions and transformed into a non-linear model using kernels. Various optimization techniques have been implemented in order to detect the minimum of the objective function: IRprop +/- and Rprop +/-, LBFGS and OWLQN.

DemPRED furthermore incorporates various state of the art feature selection strategies such as embedded Lasso, RFE, simple filters like PCC, Sequential floating forward selection, a newly developed combinatorial approach and a newly developed RFE632+ selection method.

In order to interpret predictions, various quality measurements may be generated such as: Mathews Correlation Coefficient (MCC), Area under Receiver Operating Characteristic (AuROC), RMSD, linear correlation coefficient (R), the coefficient of determination (R^2) and the geometric mean fold error (GMFE). Plotting routines will automatically generate ROC curves and correlation plots of the predictions made as well as bar plots of the model coefficients.

In order to optimize model parameters and score individual feature subsets various resampling techniques are included such as Cross Validation, Bootstrapping, Bootstrapping 632 and Bootstrapping 632+.

Last but not least, many helper functions for splitting, merging, filtering, reading, writing and normalizing data sets as well as loading and saving generated prediction models are available.

The above mentioned routines have been successfully used on various prediction tasks such as to predict major histocompatibility complex II (MHC II) epitopes, human volume of distribution and clearance as well as to detect protein interface regions. The achieved results were as good as or comparable to other state of the art prediction methods. The proposed *DemPRED* library may therefore be useful for many other biological prediction tasks.

4 Summary

In silico predictions of particular properties of biological active molecules can dramatically reduce time and costs needed to measure these properties in a wet lab. Nevertheless, the implementation of state of the art prediction techniques needs expert knowledge of machine learning methods and distinctive programming skills if starting from scratch. Hence, there is a demand for powerful yet easy to use libraries, which users can employ and extend to build their own models given a particular prediction task. During my PhD I developed such a library called *DemPRED*. The core of *DemPRED* consists of a linear scoring function. This scoring function can be combined with various loss functions, which makes *DemPRED* suitable for classification and regression. In cases were a linear model is not flexible enough *DemPRED* makes use of the kernel trick to transform the linear core into a non linear one. *DemPRED* contains many additional routines, which help users to generate reliable prediction models. These include various quality measurements as well as re-sampling strategies and routines for saving and loading of generated models. *DemPRED* includes various regularization and feature selection strategies, which make this library especially suitable for prediction tasks where few observations are described by thousands of descriptors. The object oriented implementation of *DemPRED* allows users to extend and modify the build in routines by their own ones. During my PhD I successfully used *DemPRED* on various classification and regression problems such as predicting major histocompatibility complex II (MHC II) epitopes, prediction of human volume of distribution and clearance as well as detecting protein interface regions. The predictive power of all generated models was as good as or even better than other state of the art classification and regression techniques.

5 Summary in German

Trotz fortgeschrittener Messtechniken kann das Erfassen molekularer Eigenschaften für die meisten biochemischen Prozesse sehr zeitaufwändig und teuer sein. Dies gilt insbesondere dann, wenn Eigenschaften umfangreicher Moleküldatenbanken untersucht werden sollen. Um den Prozess der Messung zu beschleunigen, werden Laborexperimente heutzutage immer häufiger durch Computer gestützte Vorhersagemethoden ergänzt. Somit können selbst große Datenbanken in einem Bruchteil der sonst dafür im Labor benötigten Zeit untersucht werden. Ohne geeignete Werkzeuge kann die Generierung eines aussagekräftigen, computergestützten Vorhersagemodels jedoch ebenfalls kompliziert und zeitaufwändig sein. Aus diesem Grund besteht die Nachfrage nach einfach zu bedienenden und erweiterbaren Programmbibliotheken, welche die Grundfunktionen für die Generierung von Vorhersagemodellen zur Verfügung stellen. Während meiner Promotion habe ich eine solche Bibliothek namens *DemPRED* entwickelt. *DemPRED* basiert im Kern auf einem linearen Model, welches mit verschiedenen Verlustfunktionen kombiniert werden kann. In Fällen, in denen ein lineares Model nicht die nötige Flexibilität liefert, kann *DemPRED* mit Hilfe des Kernel Tricks zu einem nichtlinearen Model erweitert werden. Die *DemPRED* Bibliothek bietet zudem etliche zusätzliche Funktionen an, die dem Benutzer helfen, gute Vorhersagemodelle zu generieren. Während meiner Promotion habe ich *DemPRED* dazu genutzt, unterschiedlichste biochemische Prozesse vorherzusagen. Unter anderem habe ich Modelle für die Vorhersage der MHC II bindenden Epitope, humanen Verteilungs- und Ausscheidungskoeffizienten und Protein Interaktionsflächen entwickelt. Die Qualität der generierten Vorhersagemodelle war hierbei meist besser oder aber mindestens vergleichbar zu anderen bisher verwendeten Techniken.

Statutory Declaration

Hereby, I testify that this thesis is the result of my own work and research, except of references given in the bibliography. This work contains material that is the copyright property of others, which cannot be reproduced without the permission of the copyright owner. Such material is clearly identified in the text.

Özgür Demir

References

1. Mintseris, J., et al., *Integrating statistical pair potentials into protein complex prediction.* Proteins, 2007. **69**(3): p. 511-20.

2. Pierce, B. and Z. Weng, *ZRANK: reranking protein docking predictions with an optimized energy function.* Proteins, 2007. **67**(4): p. 1078-86.

3. Chae, M.-H., et al., *Predicting protein complex geometries with a neural network.* Proteins: Structure, Function, and Bioinformatics, 2010. **78**(4): p. 1026-1039.

4. Galton, F., *Presidential address, Section H, Anthropology.* Report of the British Association for the Advancement of Science, 1885. **55**: p. 1206-14.

5. Demir-Kavuk, O., H. Riedesel, and E.W. Knapp, *Exploring classification strategies with the CoEPrA 2006 contest.* Bioinformatics. **26**(5): p. 603-9.

6. Tibshirani, R., *Regression shrinkage and selection via the Lasso.* Journal of the Royal Statistical Society Series B-Methodological, 1996. **58**(1): p. 267-288.

7. Hoerl, A.E. and R.W. Kennard, *Ridge Regression - Biased Estimation For Nonorthogonal Problems.* Technometrics, 1970. **12**(1): p. 55-&.

8. Guyon, I., et al., *Gene Selection for Cancer Classification using Support Vector Machines.* Machine Learning, 2002. **46**(1-3): p. 389-422.

9. Yu, H., et al., *Discovering compact and highly discriminative features or feature combinations of drug activities using support vector machines.* Proc IEEE Comput Soc Bioinform Conf, 2003. **2**: p. 220-8.

10. Li, H., et al., *Effect of selection of molecular descriptors on the prediction of blood-brain barrier penetrating and nonpenetrating agents by statistical learning methods.* J Chem Inf Model, 2005. **45**(5): p. 1376-84.

11. Zhu, J. and T. Hastie, *Classification of gene microarrays by penalized logistic regression.* Biostatistics, 2004. **5**(3): p. 427-43.

12. Pudil, P., J. NovoviÄ□ovÃ¡, and J. Kittler, *Floating search methods in feature selection.* Pattern Recogn. Lett., 1994. **15**(11): p. 1119-1125.

13. Vafaie, H. and K.D. Jong, *Genetic Algorithms as a Tool for Feature Selection in Machine Learning*, in *Proceedings of the 1992 IEEE Int. Conf. on Tools with AI*1992, Society Press. p. 200-204.

14. Al-Ani, A., *Ant Colony Optimization for Feature Subset Selection.* World Academy of Science, Engineering and Technology, 2005. **4**: p. 35-38.

15. Patil, D., et al., *Feature selection and classification employing hybrid ant colony optimization/random forest methodology.* Comb Chem High Throughput Screen, 2009. **12**(5): p. 507-13.

16. Aizerman, A., E.M. Braverman, and L.I. Rozoner, *Theoretical foundations of the potential function method in pattern recognition learning.* Automation and Remote Control, 1964. **25**: p. 821-837.

17. Williamson, M.G.G.a.N.C.a.J.S.-t.a.R., *Classes of kernels for machine learning: a statistics perspective.* Journal of Machine Learning Research, 2001. **2**: p. 299--312.

18. Shawe-Taylor, J. and N. Cristianini, *Kernel Methods for Pattern Analysis*2004: Cambridge University Press.

19. Rupp M, S.G., *Graph kernels for molecular similarity.* Molecular Informatics, 2010. **29**(4): p. 266-273.

20. Toussaint, N.C., et al., *Exploiting physico-chemical properties in string kernels.* BMC Bioinformatics. **11 Suppl 8**: p. S7.

21. Steinbeck, C., et al., *Recent developments of the chemistry development kit (CDK) - an open-source java library for chemo- and bioinformatics.* Curr Pharm Des, 2006. **12**(17): p. 2111-20.

22. Demir-Kavuk, Ö. *DemQSAR Webpage*. 2011; Available from: http://agknapp.chemie.fu-berlin.de/dempred/demqsar.

23. Guyon, I. and A. Elisseeff, *An introduction to variable and feature selection.* The Journal of Machine Learning Research archive, 2003. **3**: p. 1157 - 1182.

24. Goodman, J. *Exponential Priors for Maximum Entropy Models*. in *Proceedings of HLTNAACL 2004*. 2003.

25. *Comparative Evaluation of Prediction Algorithms (CoEPrA)*. 2006; Available from: http://www.coepra.org/.

26. Vajda, S. and D. Kozakov, *Convergence and combination of methods in protein-protein docking.* Curr Opin Struct Biol, 2009. **19**(2): p. 164-70.

27. Smith, G.R. and M.J. Sternberg, *Prediction of protein-protein interactions by docking methods.* Curr Opin Struct Biol, 2002. **12**(1): p. 28-35.

28. Halperin, I., et al., *Principles of docking: An overview of search algorithms and a guide to scoring functions.* Proteins, 2002. **47**(4): p. 409-43.

29. Schneidman-Duhovny, D., et al., *Geometry-based flexible and symmetric protein docking.* Proteins, 2005. **60**(2): p. 224-31.

30. Geppert, T., E. Proschak, and G. Schneider, *Protein-protein docking by shape-complementarity and property matching.* Journal of Computational Chemistry, 2010. **31**(9): p. 1919-1928.

31. Chen, R. and Z. Weng, *Docking unbound proteins using shape complementarity, desolvation, and electrostatics.* Proteins, 2002. **47**(3): p. 281-94.

32. Hwang, H., et al., *Protein-protein docking benchmark version 3.0.* Proteins, 2008. **73**(3): p. 705-9.

33. Huang, S.Y. and X. Zou, *An iterative knowledge-based scoring function for protein-protein recognition.* Proteins, 2008. **72**(2): p. 557-79.