UPTEC X 07 005
Examensarbete 20 p
Februari 2007

Proteochemometric modelling of protein microarray interactions

Markus Rasmussen

UPPSALA UNIVERSITET

Bioinformatics Program

Uppsala University School of Engineering

UPTEC X 07 005	**Date of issue 2007-02**

Author

Markus Rasmussen

Title (English)

Proteochemometric modelling of protein microarray interactions

Title (Swedish)

Proteokemometrisk modellering av proteinmikroarrayinteraktioner

Abstract

The interaction strength between two families of protein domains was modelled statiscally using proteochemometrics. Partial least squares regression modelled the relationship between the interaction data, originating from protein microarray experiments, and mathematical descriptions of the respective sequences involved.

Keywords

Proteochemometrics, partial least squares, protein interaction modelling

Supervisors

Jarl Wikberg och Martin Eklund
Uppsala University

Scientific reviewer

Mats Gustafsson
Uppsala University

Project name	Sponsors
Language **English**	Security
ISSN 1401-2138	Classification
Supplementary bibliographical information	Pages **26**

Biology Education Centre Biomedical Center Husargatan 3 Uppsala
Box 592 S-75124 Uppsala Tel +46 (0)18 4710000 Fax +46 (0)18 555217

Proteochemometric modelling of protein microarray interactions

Markus Rasmussen

Sammanfattning

Att kunna förutsäga hur biomolekyler interagerar med varandra blir en allt viktigare del av biologisk och medicinsk forskning. Proteokemometri är en metod där man utifrån kända värden på hur vissa molekyler binder till varandra försöker prediktera motsvarande egenskaper hos andra, liknande ämnen. Prediktionerna görs *in silico*, d.v.s. med datorer. Med hjälp av statistiska och matematiska metoder försöker man hitta en korrelation mellan de ingående molekylernas kemiska egenskaper och deras förmåga att interagera med varandra.

I det här examensarbetet gjordes en sådan modell över hur ErbB-receptorer, som är en familj av cellmembranbundna proteiner, interagerar med SH2-domäner på de intracellulära proteiner som aktiveras av receptorerna. Resultat från försök med proteinmikroarrayer användes för att beskriva hur ca 5000 kombinationer interagerade. Från proteinernas aminosyrasekvenser och aminosyrornas kemiska egenskaper skapades kvantitativa beskrivningar av varje proteinkombination, som tillsammans med respektive interaktionsvärde blev indata till modellen. Prediktionsförmågan blev ganska låg, främst p.g.a. den begränsade variationen på interaktionsdatat.

Examensarbete 20 p i Bioinformatikprogrammet

Uppsala universitet Februari 2007

Contents

Chapter 1

Introduction

1.1 Thesis outline

This Master's thesis describes an attempt to model, or explain, the interaction strengths between two families of proteins, based on data from a microarray experiment performed at the MacBeath Lab (1). This chapter provides some background information and motivation for the study. Chapter two describes the bioinformatics and statistics methods used to conduct the study. Chapter three presents the results and some comments on them. Chapter four discusses the results and possible sources of error. Chapter five concludes the study with lessons learned and suggests future work on the subject.

1.2 Motive

Finding out the functional properties of a newly discovered or designed protein by experimental methods is a costly and time consuming task. It involves not only having the protein, and possible binding ligands available, but also making sure all molecules involved are folded properly and in their correct chemical environment. Due to these difficulties, computational methods are persued. Several such methods exist, but many depend on having a well defined 3D structure available, at least of a reference protein as the function is assessed based on sequence similarity with previously known proteins. Determining the 3D structure is error prone, expensive and time consuming, and many proteins function only in very specific environments, e.g. across a cell membrane, that cannot be reproduced or investigated properly for a 3D structure. Since there is a strong correlation between the protein amino acid sequence and its 3D structure as well as between the 3D structure and it's function, there is much hope that methods which do not depend on a 3D structure will prove reliable (2). Here proteochemometrics is investigated as a tool to predict the interaction strength between

intracellular ErbB and SH2 domains.

1.3 Chemometric modelling and QSAR

The origin of proteochemometrics, chemometrics, is a broad field of science which generally involves relating some available measurements to a state of a chemical system. Chemometrics has evolved steadily for a long time and includes a wide range of mathematics and statistics methods and technologies. In quantitative structure-activity relationship (QSAR) modelling, the interaction between a specific target such as a protein, a cell or an organism and a series of ligands is considered. The physio-chemical properties of the ligands are collected into set of descriptors, which is a mathematical representation (typically a long vector) of experimentally and computationally derived properties. They range from simple, such as the molecular weight, to highly complex, such as molecular interaction fields derived using the 3D structure of the molecule (11). The descriptors are then fitted, using regression or machine learning techniques, towards the respective interaction strength (empirically derived) exhibited by the ligands, according to formula 1.1. The resulting regression coefficients can be used to to predict the interaction between the target and untested or hypothetical ligands that are sufficiently similar to those tested (2).

$$f(d_{ligand}) + \varepsilon = y \tag{1.1}$$

1.3.1 Proteochemometrics

Proteochemometrics differs from QSAR in that it covers a series of targets rather than one, and the data to be correlated to each interaction strength describes the chemical properties of both the current target and the current ligand as in formula 1.2. Instead of describing the "chemical space" of the ligands as in QSAR, the entire "interaction space" is thus covered by the descriptors. That way the complete set of properties considered is likely to include more properties of importance, and it is possible to use the model to discover active sites on both the target and the ligand. As the target and the ligand are not treated differently, either or both may be a protein (usually at least one of them is.) This project will concern proteochemometric modelling of protein-protein interactions (2).

$$f(d_{ligand}, d_{target}) + \varepsilon = y \tag{1.2}$$

Cross terms

The statistical methods, such as PLS and MLR, commonly used in QSAR and proteochemometrics usually reveal only linear covariances between the

molecule properties and the interaction strength. To expand them to cover some non-linear covariances as well, cross term expansion can be used. An example of a cross term expansion is taking products of descriptors and adding them as new descriptors. This approach makes nonlinear relationships available for the model and may reveal pairs of active residues or conformation determining sites. The biggest problem with cross terms is that the data matrix tends to grow very large, leading to computational problems and possible model overfitting. The number of cross terms can be limited to e.g. cross terms between areas of different molecules or expected active sites.

1.4 ErbB receptors and SH2 domains

Epidermal growth factor receptor (ErbB1) and its close relatives ErbB2, ErbB3 and ErbB4 are initiators of a very thoroughly studied set of signaling networks. Such networks are involved in a wide range of cellular activities, including apoptosis, growth, migration and adhesion. All ErbB receptors have three components: one intracellular component including tyrosin kinase, one transmembrane component and one extracellular domain which binds signaling compounds. As the receptors are activated, they phosphorylate each other on several tyrosine residues, which in turn serve as docking sites for downstream adaptor proteins and enzymes, illustrated in Figure 1.1. Such proteins will often depend on phosphotyrosine binding (PTB) or Src homology (SH2) domains for the interaction. The interaction between such intracellular docking sites and ligands with SH2 domains is what is investigated in this study (1).

1.5 Protein microarrays

Similar to DNA microarrays, protein microarrays are designed to experimentally test the interaction strengths between a set of peptides (representing protein domains) fused to a solid surface and a set of dissolved proteins, which have been altered to be fluorescent. After incubation, the fluorescence level of a specific microarray spot correlates to the amount of proteins bound to the peptides. Protein microarrays is a technology still under developement, and it is more difficult to achieve good results than with DNA microarrays. For this study, a protein microarray experiment from the MacBeath lab, in which an interaction network between ErbB phosphotyrosine sites and PTH and SH2 domains was built, has been chosen (1). Protein-protein interactions are known to be extremely difficult to model, but from a modelling point of view, it is interesting to develop an idea of how much can be accomplished with present tools and data sets such as this one. Also, developing a quantitative protein interaction network with

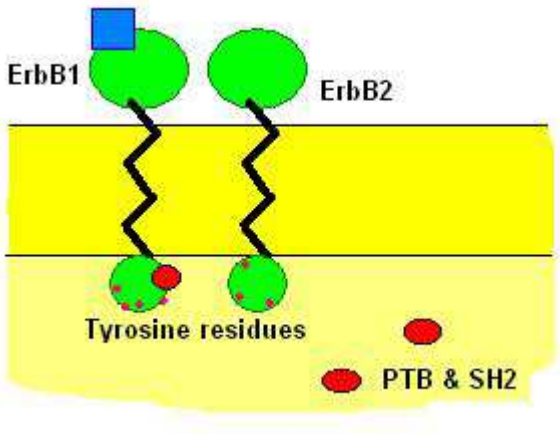

Figure 1.1: PTB and SH2 domains of intracellular proteins interact with tyrosine and phosphotyrosine residues on intracellular domains of ErbB proteins.

protein microarrays is an interesting approach to finding new knowledge and we may see many similar experiments in the future.

1.6 Software tools

Simca, a multivariate data analysis software developed by Umetrics Software, has been used for all statistical analysis and modelling in this project. Protein alignments were done with online alignment tool Muscle. Java code was written in Eclipse and this report was typeset with Texmaker and MikTex.

Chapter 2

Method

2.1 Bioinformatics methods

2.1.1 Data retrieval and parsing

For this study, protein microarray data was retreived from the MacBeath lab website (`http://www.sysbio.harvard.edu/csb/macbeath/`). The data consisted of measured interactions (dissociation constants) between a variety of epidermal growth factor receptors (ErbB) and most of the Src homology 2 (SH2) and phosphotyrosine binding (PTB) domains found in the human genome. (jones) From the data set describing 44 PTB and 109 SH2 domains versus 66 variations of ErbB1 through ErbB4 peptides, only the dissociation constants relating to SH2 domains were extracted for this study (as there were more SH2 domains, and they appeared to interact more frequently than the PTBs). An alignment of the SH2 proteins was found at the MacBeath lab, produced with the Muscle software (`www.drive5.com/muscle/`). Therefore, Muscle was used to produce an alignment of the ErbB peptides as well. The peptides turned out to align well, with rather few gaps in the multiple sequence alignment. A java parser was written to read, sort and fuse the corresponding (aligned) sequences, perform the translation into descriptors described below, and compose a data matrix where each column represented one property of a corresponding site in either the ErbB peptide or the SH2 domain and each row contained the descriptors from one particular combination. Most of the dissociation constants were ∞, which is not suitable for calculations. Therefore they were all inverted (thus most of them became zero) and then positioned in the last column of the "spreadsheet", to be imported into the Simca software.

2.1.2 Mathematical description of biomolecules

Relating the molecules to their function requires a quantitatively or qualitatively comparable description of each combination of ErbB peptide and

SH2 domain. There are many ways to produce such a description, and the more information that can be included, the better the basis for building a model. At the same time the information must be easy to acquire, and each sample should preferably not require additional laboratory work. One way of describing biopolymer sequences, is to use a set of descriptors for each of their amino acids. Instead of performing chemical analysis on proteins, one uses empirically and mathematically derived descriptions of amino acids, combined with the protein amino acid sequences. The most evident advantage of this approach is that it uses the polymeric nature of the proteins to avoid involving a 3D structure of the entire protein. The descriptors used in this study originate from Sandberg et. al 1998 where 26 measured and computed properties were reduced to five principal components covering about 95% of the total variance (3). These components, each representing a linear combination of amino acid properties, are called z-scales ($z_1 ... z_5$). All non-modified amino acids present in human proteins have a set of z-scale descriptors, but not the phosphrylated tyrosine found in many of the ErbB peptides. As that particular site was always either tyrosine or phosphotyrosine, a single binary (0/1) descriptor was used for it.

2.1.3 Alignment independent alternatives

As only completely comparable properties can be used in this kind of analysis, only areas not aligned towards a gap in any sequence can be used in the model unless further processing is performed. Such gap-free areas are called blocks. They can be translated into z-scale descriptor series and put directly into the X matrix. If the alignment of the sequences contains many gaps, significant information can be lost as all sequence stretches aligned towards gaps must be omitted. If so, it may be possible to process the complete sequences in such a way that they become independent of the alignment quality. One such approach is called auto and cross covariance (ACC) (4). The idea is that rather than finding covariance in every descriptor position, one looks along the sequences for covariance a certain number of descriptors apart (lag). While loosing the ability to identify specific important sites in the sequence, it is possible to find patterns correlating to the structure and function of the entire sequence, that do not depend on it's size or sequence length. Such covariances in a z-scale descriptor sequence can be obtained by repeatedly (for each lag) running a "sliding window" along the descriptor sequence, collecting a covariance of a specific lag each run (Formula 2.1, Figure 2.1). Auto covariance is obtained for comparisons between the same type of descriptor (e.g. z_1) and cross covariance for comparisons between different descriptors. Auto covariance and cross covariance have been treated equally in this study. An ACC descriptor is produced for each combination of z-scale descriptors j and k (such as z_1, z_4) and lag according to Formula 2.1.

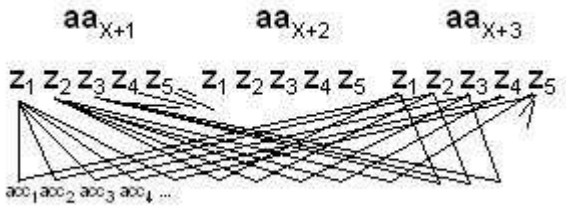

Figure 2.1: Each ACC entry requires sliding along the entire descriptor sequence, collecting the covariance between descriptors a certain distance apart.

$$ACC_{j,k,lag} = \sum_{i=1}^{n-lag} \frac{(z_{j,i} - \overline{z}_j) \times (z_{k,i+lag} - \overline{z}_k)}{n - lag} \tag{2.1}$$

2.2 Statistics methods

2.2.1 Data processing

Mean centering and scaling

The data set produced by the parser consisted of a large matrix with one row per "object" (combination of ErbB and SH2) and one column per variable (or descriptor). To make sure the variances among the descriptors were not dependent on differences in unit or scale, all variables were mean centered and scaled to unit variance. Mean centering means calculating the mean values of every column and subtracting it from every individual entry. Thus, all objects were scattered around the origin. Unit variance means that all columns are scaled to a variance of one. Where large blocks of cross terms or auto and cross covariance entries are present, they may be scaled down to a smaller variance. Dividing the variables into groups of different types of descriptors and scaling accordingly is called block scaling (not to be confused with alignment "blocks").

Log transformation

Some of the variables may not be evenly distributed around their mean value. Such variables, with a high skewness, may have a bad influence on the model as statistical methods often expect data to be normally distributed. This is typically the case for methods that depend on linear relationships. Skewness is usually measured with Formula 2.2. Skewed variables can often be

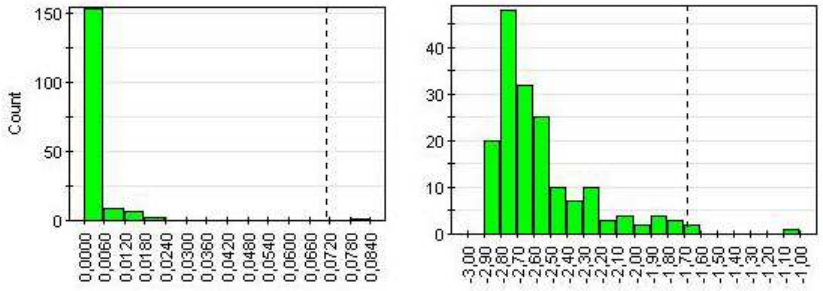

Figure 2.2: Histogram of skewed interaction data (inverted dissociation constants) before and after log transformation.

transformed to resemble a normal distribution by using a simple logarithmic, exponential or polynomial formula on the original values. Figure 2.2 shows the effect of log transforming the interactions values used in this study.

$$Skewness = \frac{\sum_{i=1}^{N} \left(Y_i - \overline{Y}\right)^3}{(N-1)\, s^3} \tag{2.2}$$

2.2.2 Principal components analysis

This study mainly relies on a regression technique called partial least squares (or sometimes projection to latent structures). PLS has much in common with a projection technique called principal components analysis (PCA). PCA is a common way to simplify, or compress, data of many dimensions by projecting it onto a subspace of a convenient number of dimensions. The new coordinate system is chosen as an orthogonal set of linear combinations of the original system such that the direction of greatest data variance (most principal component) becomes the first new coordinate axis, the direction of greatest remaining variance becomes the second coordinate axis and so on (Figure 2.3). In many data sets, only a handful of principal components may contain the majority of the total variance. PCA is an optimal method, with regards to minimizing the squared sum of residual variance (variance lost), to find the subspace of greatest variance. Performing PCA on a data set may provide important information on how its variance is distributed. The principal components will not, however, necessarily be the components along which the variance is of most importance regarding a specific problem (5) (6).

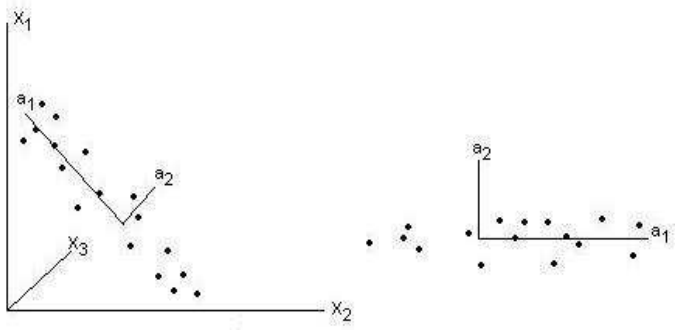

Figure 2.3: PCA can compress data into a new base with fewer dimensions with a minimum squared error.

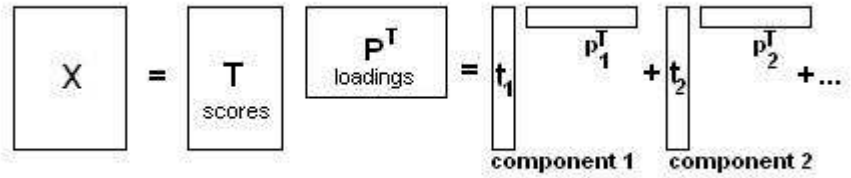

Figure 2.4: Matrix operations overview of PCA.

PCA scores and loadings

The transposing of the objects in X onto the new basis can be represented by a matrix operation $X = T \times P^T$ where T and P are called the scores matrix and the loadings matrix, respectively (Figure 2.4). The scores matrix desribes all objects from X projected onto the new basis. The loadings matrix describes how the variables in X are combined to form the new base. To display the projected data in the original coordinate system, simply multiply $T \times P^T$.

The scores and loadings can be used to investigate various properties of the data set. By plotting the scores of the first components against the scores of the second component it is possible to discern groups or classes of objects from each other, or find outliers capable of disturbing the model. Outliers are objects that do not appear to belong to the same probabilistic distribution as the rest of the data, and they may have appeared as a result of faulty measurements or other errors. Removing outliers can improve the model, as long as there is a reason to believe that they are there because

of an error. A corresponding plot of the loadings will reveal covariances among the variables, that is, if one object has a certain value of a particular variable, which other variables can be predicted. "Outliers" in the loadings scatter plot are variables that show an unexpected bahavior, perhaps due to a faulty measurement instrument.

2.2.3 Principal components regression

A regression model with hundreds or thousands of variables and thousands of objects is problematic for simple regression methods such as multiple linear regression. The system is likely to be underdetermined if descriptive methods such as cross terms are used, and even if it is not, a large number of variables compared to measurements makes overfitting easy. Therefore it is usually necessary to limit the regression to the parts of the data that exhibit the most variance, essentially compressing the data into fewer variables. With the principal components found in PCA, it is straightforward to imagine a regression model where the components of greatest variance are fitted towards the interaction values (or their principal components) so that $TB + E = Y$. T would be the (chosen components of the) descriptor matrix, Y the response matrix (interaction strength), B a regression coefficient matrix and E a noise term (with the same dimensions as Y). Such a model, called principal components regression (PCR), has the drawback that T, the components of greatest variance, might not be the components of greatest importance for predicting (Y) (8).

2.2.4 Partial least squares

A different approach called PLS (partial least squares or, sometimes, projection to latent structures) has the advantage of extracting components of the X data that have a high variance *and* great correlation with Y. Just like PCA, PLS essentially projects the objects in X onto a new base. However this new base is selected with respect also to the correlation with Y. Both regression models, principal components regression and partial least squares regression, produce linearly uncorrelated factor scores. That is, the new base describing the X components of choise is orthogonal. The difference between them is in how the scores T are extracted. In PCA the loadings matrix P^T represents the covariance structure of X, while it's PLS counterpart represents a combination of the covariance structure in X and the correlation between X and Y (7).

Algorithm

The most popular and the first partial least squares algotithm, nonlinear iterative partial least squares (NIPALS), has the advantage to calculate one regression component at the time. If matrix operations were used instead,

the entire set of components would have to be calculated at once. That would lead to unnecessary, potentially very time consuming computations as only a few components are likely to be needed. Between extracting components, the model validation described below is performed. The presently used NIPALS was developed by Wold et al. in 1987 (8).

Scores and loadings

The scores and loadings matrices produced by the PLS procedure can be examined in roughly the same way as PCA scores and loadings. However, objects are close together in the score scatter plot not only due to covariance between X variables, but also because of similar covariance between their X and Y variables. For example, two molecules with many similar chemical properties would be close together in a PCA score scatter plot, but for PLS they would also have to have those chemical properties correlated to their Y data with similar correlation coefficients. Variables or attributes close together in the PLS loadings scatter plot would not only be correlated in X but also show a similar impact on Y.

2.2.5 Model validation

To optimize the number of regression components included in a PLS model, validation of the model is carried out between calculating the components. Including too few components will allow part of the relationship between X and Y to go to waste, and including too many will cause an overfitted model. The measurements normally used to assess the model are R^2X, R^2Y and Q^2. R^2X represents what fraction of the sum of squares (complete variance) in X that resides in the components selected so far (remains after the projection). Similarly, R^2Y represents what fraction of the Y variance that is explained by the model. R^2 is displayed cumulatively (not for individual components, in which case it would represent the fraction of the total variance that is covered by that particular component). Therefore it increases with the number of PLS components included. For data scaled to unit variance, R^2X and R^2Y would be 1 if all components were included. However too many PLS components will make the model overfitted as it covers smaller, noise-like covariances between X and Y. To assess the predictive ability of a model, a cross validation loop is performed. The objects are randomly separated into groups that are repeadedly left out of the PLS regression. The model is then evaluated on those left-out objects. Q^2 (cumulative) represents the amount of Y variation that appear be possible to predict with the model, according to the cross validation. In Simca, Q^2 is calculated as $1 - \prod_A (PRESS/SS)$. The prediction error sum of squares ($PRESS$) is the squared differences between observed and predicted values for the data kept out of the model fitting, SS is the sum of squares (to-

tal variance) in Y and A is the set of components selected. Q^2 can be no higher than R^2Y and normally starts low for few components, increases to a plateau for the optimal number of components and then declinies as the model is more and more overfitted (9) (8).

Chapter 3

Result

3.1 Result summary

An effectively predictive model could not be built from the data set. However:

- The model was improved by log-transforming the much skewed Y vector
- The model improved slightly when adding SH2 domain auto and cross covariance descriptors.
- The model also improved when using cross terms and block scaling.
- It was indicated that a better variance in the Y vector would produce a much better model

For a complete table of R^2X, R^2Y and Q^2 results in the models tried, see the appendix.

3.2 Data matrix

The structure of the data set, which was the input for the model, is shown in Figure 3.1.

3.2.1 Objects

The basic data matrix produced by the java parser ended up covering interactions between all combinations of 52 ErbB peptides and 105 SH2 proteins. The X matrix and Y vector (produced concatenated by the parser) thus came to have 5460 rows. This is the number of objects in the model.

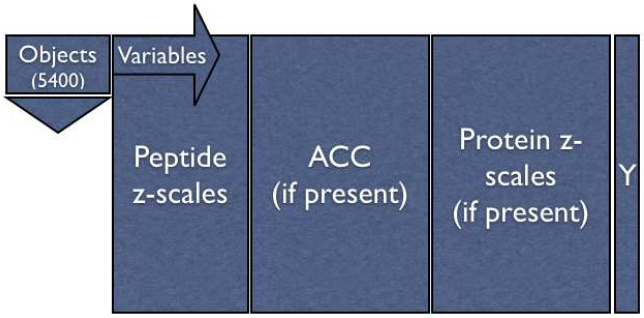

Figure 3.1: Structure of the data set imported to Simca

3.2.2 The X matrix

Each row in the X matrix constitutes one measurement, or object. Each column covers one variable, an object property of some sort which is comparable within the complete set of objects, such as the hydrophobicity at one particular position. As several methods of protein description were tried, the number of columns in the X matrix varied throughout the experiment. As the ordering of columns is of no importance in the model, blocks of protein and peptide Z-scales, protein auto and cross covariance data and numbers representing the lengths of unalignable protein areas were just concatenated into different combinations as needed. With five Z-scales per amino acid residue, the number of columns in the protein Z-scale blocks was 160. As one of the amino acids in the peptide sequences had a binary descriptor only, the number of columns in the peptide Z-scales was 76. The auto and cross covariance block had 500 columns (with a maximum lag of 20. No improvement could be seen for higher maximum lags).

3.2.3 The Y vector

With most combinations of peptide and protein not showing any interaction in the microarray experiment, the Y vector turned out highly skewed, as most of the inverted dissociation constants used were zero. The number of non-zero values were only 171 out of a total of 5460, producing a skewness of 43.4. The set of non-zero values was also skewed towards low values, having a skewness of 8.55. The Y entries ranged between 0 and 0.08.

3.3 Countering intraction value skew

The Y skewness of 43.4 was reduced to 9.27 by log-transforming the vector:

$$y' = lg(y + 0.001), \tag{3.1}$$

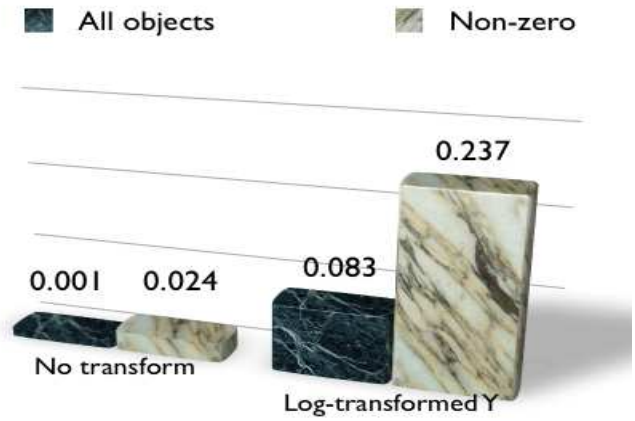

Figure 3.2: Q^2, the predictive power of the model, improves with a log transformed Y vector. The improvement is greater when excluding objects with a Y value of zero.

the small addition as $lg(0)$ is undefined). None of the X variables had an obvious skewness. The Y skewness may have been possible to reduce slightly further by adjusting the formula, but the underlying problem is the large number of zeros. After transforming, the skewness of the set of non-zero values fell to only 1.86, indicating a rather even distribution among them (Figure 2.2). The effect of log transforming Y on the predictive ability of the model is displayed in Figure 3.2.

3.4 Countering poor protein alignment

One possible source of poor results next to the Y skew is the rather low quality of the SH2 domain alignment. The alignment algorithm had not only aligned areas whose representation of the same function could be questioned, but also produced quite a few large gaps where stretches of sequences apparently showed no resemblance to each other. As only blocks, or areas completely without gaps, can be used in the PLS modelling, any information in such stretches was lost. Blocks that are of questionable quality may be an important source of noise data in the PLS model, as they might not represent comparable functions. Two different approaches were tried to combat this problem. The first calculated the lengths of stretches lost due to gaps and added those numbers to the end of the descriptor sequences. Although their content would still be lost, at least the lengths of the areas would survive into the PLS modelling.

The other approach implemented an alignment independent method of describing the z-scale descriptor sequences of the complete (not just blocks)

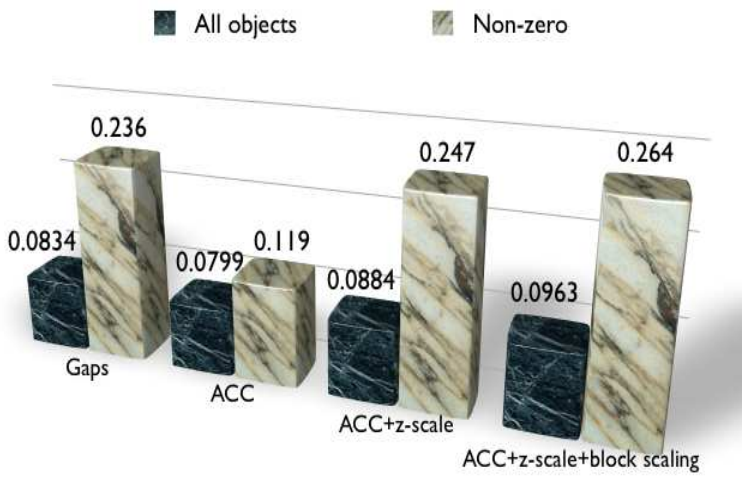

Figure 3.3: Q^2 did not improve when adding descriptors representing the lengths of sequence parts lost due to gaps in the alignment. Replacing the SH2 z-scale descriptors with AAC had a negative impact, but including both improved Q^2. Block scaling improved Q^2 further.

SH2 domains, auto and cross covariance (ACC). The ACC columns cover covariances a certain number of descriptors apart in a sequence, up to a maximum lag equal to the length of the shortest sequence -1. The number of covariances becomes the same for all sequences and no parts are left out of the model. However, ACC cannot be used to find active sites in the proteins. The results displayed in Figure 3.3 are for ACC with a *lag* of up to 20. Including longer lags did not appear to improve the result. Q^2 values are shown for models with and without the protein z-scale blocks, and also with and without excluding rows where $y = 0$.

3.5 Block scaling

When several different types of descriptors are used, and their numbers are very different, some descriptors may be too few to impact the model according to their importance. Therefore, blocks of protein z-scales, peptide z-scales and ACC descriptors were rescaled such that each block had the same total variance. That was done after the first scaling to unit variance. Rescaling of blocks with certain weights might further improve the predictive power of this kind of model, but was not tried due to time restriction.

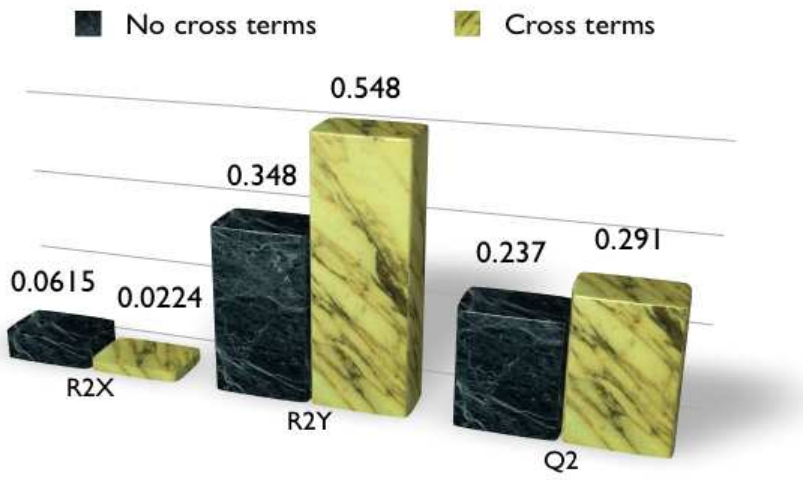

Figure 3.4: PLS on the reduced data set with non-zero Y values. Cross terms reduced R^2X, the ratio of X variance covered, probably becauce the amount of X variance increased greatly with cross terms. The increased descriptive power of cross terms improved R^2Y, the ratio of Y variance explained, and Q^2, the ratio of Y variance possible to predict.

3.6 Cross terms for non linear components

As the basic PLS regression only discovers linear components of variance, any important nonlinear relationships between combinations of descriptors and Y values will be lost. However, it is possible to convert some nonlinear combinations to linear by adding cross terms of descriptors to the model. As the data set grows rapidly with cross terms, for computational reasons this was only tried for a reduced data set where nonbinding ($Y = 0$) objects are omitted. Therefore the result may be seen as a clue to how cross terms would affect the model had the data set been less skewed. The R^2X, R^2Y and Q^2 values of Figure 3.4 represent models without ACC, and with and without cross terms of all z-scale descriptors.

3.7 Scores and loadings scatter plots

Figures 3.5 and 3.6 show examples of scores scatter plots from PLS experiments without and with ACC descriptors. Two or three groups of objects can be seen, but they proved only to be groups objects with similar SH2 sequences. With a bigger data set, such groups might have been suitable for separate modelling. When ACC descriptors are used, that separation is mush less visible. Red dots are objects where $Y > 0$. Although a clear

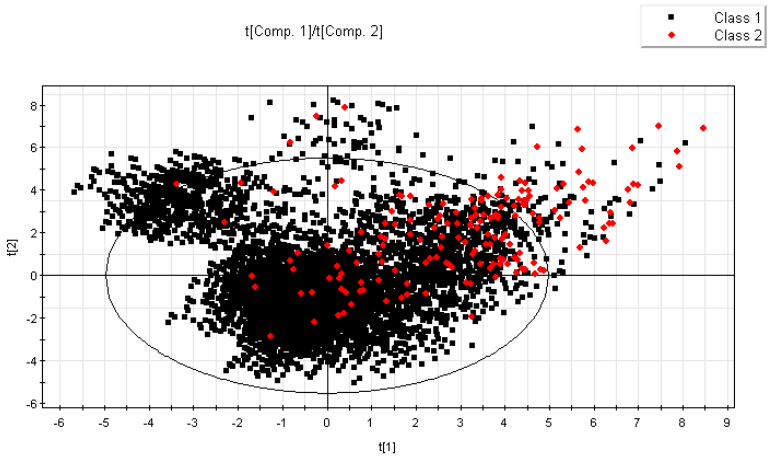

Figure 3.5: Scores scatter plot (first 2 PLS components) of all objects, using z-scale descriptors only.

separation between red and black dots cannot be made, it appears that the red are differently distributed. Figure 3.7 is a loadings scatter plot from the same model as Figure 3.5. For a model with a high predictive ability, the covariances between variables/descriptors and their impact on Y can be used to find sites of special importance for the interaction.

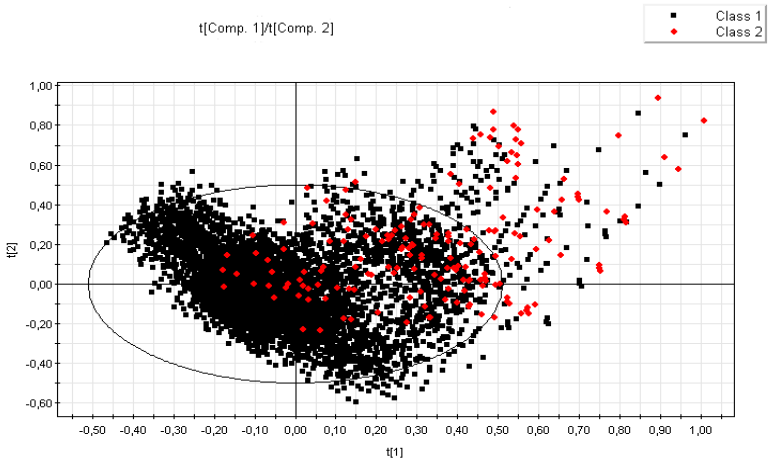

Figure 3.6: Scores scatter plot (first 2 PLS components) of all objects, using z-scale and ACC descriptors but not cross terms.

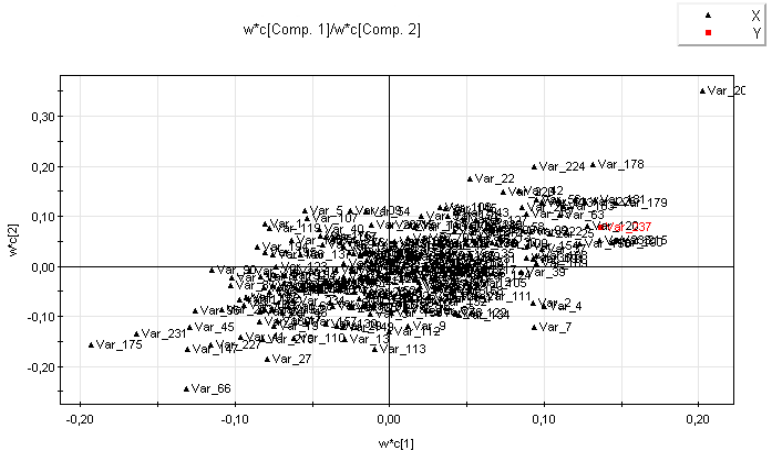

Figure 3.7: Loadings scatter plot (first 2 PLS components) of a model using z-scale descriptors only.

Chapter 4

Discussion

With fewer than 200 out of 5000 interaction strengths being above zero, a highly satisfactory result was not expected from the model. However, although it turned out difficult to make the data work well in a PLS model, it must be assumed that data such as this is common in this sort of microarray experiments. Protein microarrays is a new technology, different in many ways from DNA microarrays, and making it reliable can take some time. Protein-protein interactions are also known to be difficult to model. All in all, the resulting PLS model in this project was not very successful.

4.1 Data set

The skewness of the data set was the most problematic issue encountered during the project, and no method capable of dealing with it properly was found. The reason why this occured is mainly that methods such as PCA and PLS require the input data to be approximately normally distributed. Log transformation would have helped more against the skewness if not the majority of Y values had been exactly the same. Excluding objects that do not interact or certain SH2 or ErbB sequences entirely may improve the apparent result to a degree, but is contraproductive for predicting unknown data. Not only is the number of interacting objects rather few, excluding unwanted objects changes the distribution on which the model is built, reducing it's real predictive power. Naturally, the space of effectiveness of the model will be limited to sequences similar (i.e. particular descriptors represent the same function or feature) to those on which the model is being built, so the sequence feature space covered needs to be kept as large as possible. The addition of auto and cross covariance data to the model improved the result only slightly, but there may be other alignment independent ways to improve the mathematical description of objects and cover features not reached by the model in this study. At the same time, in order to use the full potential of cross terms, the total size of the data set must not be allowed

to grow out of hand. A statistical analysis of the data prior to modelling may have predicted the low likelyhood of success, but there was not a many other such data sets to choose from.

4.2 PLS results

A predictive model could not be produced for the complete data set. Omitting all objects with a Y value of zero provides a hint of what could be done with less skewed data. It is possible that the result of a more evenly distributed (more interacting objects) data set could be much better than seen here as the total number of interacting objects was rather low, only about 170. Log transformation has a good effect on skewed data provided that it is not concentrated to one value as was the case here. Cross terms also has a positive effect, and more work could have been done to evaluate their effect. For example, in Simca it was not possible to produce cross terms only between separate blocks of variables without tedious manual work. As cross terms of all possible variables cause polynomial growth of the variable matrix, it was not attempted on the complete data set. Nothing indicated that the results would have come close to those achieved with interacting ($Y > 0$) objects only, and that produced a Q^2 of only about 0.3. As most models tried became overfitted after the first component, it must also be concluded that there was little evidence of a strong correlation between X and Y values.

4.2.1 Sources of uncertainty

It has been shown that the model improves with a reduced skewness. But it remains a question whether there were other problems with the data that showed no interaction. There may have been shortcomings in the microarray experiment that prevented pairs of ErbB peptide and SH2 domain from interacting as they should. Some peptides are also much more promiscous than others. Perhaps there are differences in the 3D structure between groups of sequences, despite a high sequence similarity. Such differences could make comparisons of certain sites useless as they would differ in function. There is also a chance that the validation method used may not be optimal to the problem. However, it is unlikely that the model quality is underestimated. Freyhult et al. (10) showed that the formula used by the Simca software to calculate Q^2 might produce a slight overestimation under certain circomstances and there is evidence that a PLS model can become overfitted unless a double cross validation loop is used (9). In this particular study, every SH2 and ErbB descriptor sequence is found in a large number of rows in the X matrix. That might cause a problem as all sequences will probable always be represented inside the cross validation loop. Finally there is some risk of human errors during data handling and programming.

4.3 Improving the model

4.3.1 Binary classification

The PLS results improve radically when non interacting combinations are left out of the model. If there was a method capable of distinguishing combinations likely to interact from those that are unlikely, the skewness of the data set might be possible to reduce before running the PLS model. There are many mathematical and statistical clustering and classification methods available that might be useful. One classification method, Rough Sets, was tried briefly during this project. The 5400 objects were designated "interacting" or "non interacting" depending on whether their Y value was above zero. Then a reducing algorithm was used to produce a set of logic rules for automatically separating them. Cross validation was used to validate the classifier, which turned out very ineffective. Although a set of rules was produced successfully, they turned out classifying almost all objects as "non interacting", failing to discriminate significantly between the two classes. One reason was probably that the large number of non interacting objects compared to interacting caused the classifier to consider itself rather successful with a 90% correct classification, and therefore failed to discriminate between good and useless rules. A thorough analysis of the rough set method was omitted due to lack of time.

Additional information

It might have been possible to improve the model further by adding more information on the proteins involved. For example, sites known to be active in the interaction could have been weighted to impact the model more. This was not considered since the whole idea of proteochemometrics is to enable prediction of function without relying on the support of a 3D structure or other advanced additional research. This study aimed only to find a relationship between the sequences and their interaction strengths.

4.3.2 Using the model

Had the model turned out to have a good predictive ability, a discriminant analysis might have been able to reveal sites of importance for the interaction. By looking at score scatter plots, it would be possible to find out which PLS components separate interacting objects from non interacting ones. Corresponding loading scatter plots would then indicate which ones of the original variables influence the interaction the most. Such knowledge could be used to engineer new receptors or ligands. A discriminant analysis has not been made in this project due to the low predictive ability of the model.

Chapter 5

Conclusion

The ErbB-SH2 interaction data set recovered from Jones et al turned out to be too skewed towards zero to make a good model. However, some techniques for improving and processing data prior to PLS modelling appeared to have a positive effect on the model's predictive ability. Excellent results were not expected, but it might have been a good idea to try to estimate what could be done with this data set, and others, before deciding to use it. It would be interesting to compare the result of this study with those of other proteochemometric models and look for data set properties necessary for successful modelling. Perhaps in the future this kind of modelling can be kept in mind when designing microarray experiments in order to secure statistically viable data. During parsing of this data set, it was also evident that standardization of microarray result formatting would reduce the amount of sorting and programming required to parse the data, and reduce the likelyhood of errors making their way into the model.

Acknowledgements

I would like to thank my supervisors Martin Eklund and Jarl Wikberg, my scientific reviewer Mats Gustafsson and all the helpful staff at Pharmaceutical Pharmacology and at the Biology Education Centre.

Bibliography

[1] Jones R.B, Gordus A, Krall J.A, MacBeath G: "A quantitative protein interaction networl for the ErbB receptors using protein microarrays" *Nature 439* (2006): 168–174.

[2] Wikberg J.E.S, Lapins M, Prusis P: "Proteochemometrics: A tool for modeling the molecular interaction space", *H. Kubinyi, G. Mller, eds. Chemogenomics in Drug Discovery - A Medicinal Chemistry Perspective 22* (Weinheim, Wiley-VCH, 2004): 289–309.

[3] Sandberg M, Eriksson L, Jonsson, J, Sjstrm m, Wold S; "New chemical descriptors relevant for the design of biologically active peptides. A multivariate characterization of 87 amino acids." *JJ Med Chem 41* (1998): 2481–2491.

[4] Edman M: "Detection of sequence patternsin membrane proteins" *Phd Thesis,* (Umea University, Umea 2001)

[5] Webb A: *Statistical Pattern Recognition 2nd ed.* (John Wiley and Sons Ltd, Chichester 2002): 319–329.

[6] Wold S., Esbensen K, Geladi P: "Principal Components Analysis" *Chemom Intell Lab 44* (1987): 37–52.

[7] Wold S, Jonsson J, Sjostrom M, Sandberg M, Rannar S: "DAN and Peptide sequences and chemical processes multivariately modelled by principal components analysis and partial least squares projections to latent structures" *Ana. Chim Acta. 277* (1993): 239–253.

[8] Lapins M: "Developement of Proteochemometrics - A New Approach for Analysis of Protein-Ligand Interactions" *Phd Thesis* (Uppsala University, Uppsala 2006): 22–29.

[9] Gustafsson M. G: "A Probabilistic Derivation of the Partial Least-Squares Algorithm" *J. Chem. Inf. Comp. Sci. 41* (2001): 288–294.

[10] Freyhult E, Peteris P, Lapinsh M, Wikberg J.E.S, Moulton V, Gustafsson M. G: "Unbiased descriptor and parameter selection confirms the potential of proteochemometric modelling" *BMC Bioinformatics 6* (2005).

[11] Goodford P.J: "A Computational Procedure for Determining Energetically Favourable Binding Sites of Biologically Important Macromolecules" *J. Med. Chem. 28* (1985): 849–857.

Appendix: Table of PLS model results.

A = *Number of components selected*

Experiment configuration		**A**	**R^2X**	**R^2Y**	**Q^2**
z-scales	all objects	1	0,0227	0,0341	0,00992
only	non-zero	1	0,0579	0,219	0,024
	all, log(Y)	3	0,0805	0,127	0,0831
	log(Y), non-zero	1	0,0615	0,348	0,237
	log(Y), non-zero, cross terms	1	0,0224	0,548	0,291
	log(all), non-zero, cross terms	1	0,0224	0,55	0,293
with descriptors	all objects	1	0,0229	0,0337	0,00968
representing	all, log(Y)	3	0,0809	0,127	0,0834
lengths of areas	log(Y), non-zero	1	0,0604	0,349	0,236
lost due to gaps	log(Y), non-zero, cross terms	1	0,025	0,538	0,274
acc instead of	all, log(Y)	2	0,0385	0,124	0,0799
SH2 z-scales	log(Y), non-zero	1	0,0788	0,269	0,119
acc and SH2	all, log(Y)	2	0,041	0,128	0,0884
z-scales	log(Y), block scaling	2	0,0396	0,136	0,0963
	log(Y), non-zero	1	0,0714	0,365	0,247
	log(Y), non-zero, block scaling	1	0,0574	0,391	0,264