Selective Quantification of Co-Eluting Proteins by means of Partial Least Squares

Sofia Henryson 880504

12 november 2014

Abstract

The aim of this thesis is to create a model of the correlation between absorbance and concentration of co-eluting proteins. A separation using the chromatograph system ÄKTA from GE-Health care of BSA, Lysozyme and IgG was performed. Absorbance was measured at three different wave lengths. Since the mentioned proteins do not co-elute in a neutral pH the experiments were performed at a high pH-level which rendered the experiments unsuccessful. A PLS-function was written in Matlab. The attempt to create a model of the correlation was found futile since the data obtained from experiments did not contain enough information about the proteins. The written PLS-model was evaluated using an old data set and was found to be an useful and fast algorithm.

Sammanfattning

Arbetets syfte är att skapa en modell för sambandet mellan absorbans och koncentration av sam-eluerurande proteiner. En separation av BSA, lysozym och IgG med hjälp kromatografi systemet ÄKTA från GE-Heathcare utfördes. Absorbans mättes vid tre olika våglängder. Eftersom de nämnda proteinerna inte sameluerar i ett neuturalt pH-värde utfördes experimentet vid ett högt pH-värde vilket gjorde att försöken misslyckades. En PLS-function skrevs in Matlab och användes i ett försök att skapa en modell av korrelationen. Modellen bedömdes som oanvändbar eftersom de data som erhållits från försök inte innehöll tillräckligt med information om proteinerna.PLS-modellen utvärderades med hjälp av en gammal datauppsättning och vilket visade sig vara en användbar och snabb algoritm

Contents

1 Introduction

The aim of this thesis was to evaluate the possibility to describe the relation between concentrations of co-eluting proteins and the absorbances of the mixture using chemometric tools as Partial Least Squares method, PLS, also known as Projection to Latent Structures by the means of Partial Least Squares. Brestrich et al. (2014) and Hansen et al.(2011) showed that PLS have been used to quantify proteins.

2 Background

2.1 Co-eluting proteins and Real Time Pooling

PLS has been proven a successful tool for selective quantification of co-eluting proteins. The article "A tool for selective inline quantification of co-eluting proteins in chromatography using spectral analysis and partial least squares regression" describes how PLS is used for inline quantification of co-eluting proteins. Protein solutions consisted of Lysozyme, Ribonuclease A and Cytochrome C in buffer solutions with pH 7. The separation of proteins were preformed in the liquid chromatography system ÄKTA purifier on an SP Sepharose FF column and a Diode Array Detector, DAD, was used to detect the absorbance spectra. The PLS-model was based on a four-level D-optimal design consisting of 29 samples and the concentrations levels ranged from 0 to 0.7 g/L. The model was validated by using a three-level D-optimal design consisting of 25 samples with the same concentration span as calibration experiments. The absorbance was measured in the band of 240-300 nm with 1 nm resolution for each sample. The results showed that PLS-modeling was useful for inline quantification compared with offline analytical methods based on collected fractions. This showed that PLS is an useful tool for inline quantification and real time pooling decisions.(Brestrich et. al, 2014)

2.2 Principal Component Analysis- PCA

Principal Component Analysis uses the covariance to create a new base that describes the material better. The principal components are orthogonal and the first components holds the most information about the material. The principle of PCA is shown i equation (1)

$$X = T \cdot P + E \tag{1}$$

2.3 Partial least squares- PLS

The general idea of PLS is to find the latent structures describing the material in order to reduce the amount of dimensions by finding a new base, principal components, that describes the material better than the measured variables. PLS origins from Principal Component Analysis, PCA, developed by Pearson (1901) for applications in biomecric studies. PCA is based on eigenvalue decomposition of the covariance matrix or singular value decomposition of covariance matrix and is mainly used for exploratory data or prediction modeling. PLS was developed by Horst in 1961 and beares resemblance with PCA but PLS maximizes the covariance in the common structures of both the parameters and the responses. (Fonville et. al, 2010) Historically PLS was a method used for only one response but an extension of PLS was suggested often called PLS2. Further on in this thesis the concept PLS aims primary at PLS2 since several responses is used.

One of the advantages with the PLS model is the ability to predict with a smaller risk of random correlation between matrices. PLS also has the ability to handle several responses and describe more variables than observations which might lead to over prediction when using other linear regression methods. (Cramer, 1993)

2.3.1 Principle of PLS

The principle of PLS is shown in Equation (2) and (3) below. X is the measures parameters and the input parameters in the future model, C is the calculated responses and output in future model. The number of rows in C should be equal to the number of rows in X.

$$X = T \cdot P + E \tag{2}$$

$$C = T \cdot Q + F \tag{3}$$

T are called scores and P and Q are called loading even though the names are the same as in principal component analysis the scores and loading matrices in PLS differs from the one in PCA. The scores are orthogonal but the loading are neither orthogonal or normalized. The importance of an PLS component is based on both the scores and loading matrices and are defined as the product of the sum of squares of both scores and loadings.

The algorithm used in this thesis also calculates the weight matrix W, which is a normalized vector proportional to Q. Equation (4) shows the matrix W and the diagonal matrix b.

$$C = T \cdot b \cdot W + F \tag{4}$$

2.3.2 PLS-method used

The algorithm used in the created model is a straight forward iterative algorithm and follows the same method as in Brereton (Brereton, 2003).

The input matrices might be centered and standardized in advance depending on the material. If X is centered with regards to rows C needs to be centered too.

Calculate a vector h using Equation (5)

$$h = X' \cdot u \tag{5}$$

where u is a vector and initially a guess that can be chosen as one of the columns in the response matrix, C, to maximize the covariance between X and C. Scores are calculated as in Equation (6) using the initial guess. Loadings for both responses and parameters are calculated as in Equation (7) and (8) below.

$$\hat{\mathrm{t}} = \frac{X \cdot h}{\sqrt{\sum h^2}} \tag{6}$$

$$\hat{\mathrm{p}} = \frac{\hat{\mathrm{t}} \cdot X}{\sqrt{\sum t^2}} \tag{7}$$

$$\hat{\mathrm{q}} = \frac{C' \cdot t'}{\sqrt{\sum t^2}} \tag{8}$$

The sum of squares of the score vectors are calculated. For the first iteration or if the calculated score is unsatisfying the score will be discarded. A new score vector will then be calculated using the same approach as above using a new guess for u. Otherwise the calculated scores and loadings will be saved and the next components will be calculated.

In Equation (9) the effect of the new PLS-component is substracted from the data-matrix so that it is not calculated again. The residual data matrix is obtained

$$_{resid}X = X - t \cdot p \tag{9}$$

The estimated responeses are determined in Equation (10) and the residual data matrix for responses are calculated using Equation(11).

$$\hat{\mathrm{C}} = t \cdot q \tag{10}$$

$$_{resid}C =_{true} C - C \tag{11}$$

This is an iteration model and the next component is calculated using the same method with a new guess for u returning to Equation (5).

3 Materials and Methods

The aim of the experiments were to get a data set to be able to create a model of the relation between abs/s and concentration of specific protein/s. The input data is the absorbance measured at three different wavelengths and the output is the concentration of proteins.

Three rounds of experiments were performed, the first round using two proteins, the second round referred to as scouting experiments and the final round of experiments using a ternary proteins solution on which the data analysis is based upon.

3.1 Materials

The experiments were carried out on an ÄKTA purifier from GE Healthcare. Column used was an anion exchange column, HiTrap Q FF 1 ml from GE Healthcare.

The first round of experiments was performed using Bovine Serume Albumin and lysozyme. Loading buffer, Buffer A, 20mM sodium phosphate. Elution buffer, Buffer B, 20mM sodium phosphate and 500 mM sodium chloride

Scouting experiments were performed to find a pH-level and salt concentration in which the proteins would co-elute. A fullfactorial design of experiments was used, pH varied at three levels and the salt concentration of the elution buffer varied at two levels. The protein solution consisted of lysozyme, Bovine Serum Albumin and Immunoglobulin G. Loading buffer, Buffer A was 20mM potassium phosphate and elution buffer, Buffer B, 20 mM potassium phosphate and 500 or 100 mM potassium chloride. pH varied at three levels between 8-12.

The final experiments were executed with the same buffer solutions as scouting experiments using the ternary proteins solution at pH-level 11.5 and with a salt concentration of 500 mM in the elution buffer.

3.2 Methods

3.2.1 Experimental work

For the first round of experiments a 2-factor full factorial design was used using two center points. 5 different ratios and the maximum concentration were 0.9 g/L. There were no mixing constraints. Absorbance was measured at wavelengths 562 nm, 280 nm 527 nm. Although six experiments were

planned only three was executed using the center points and the maximum concentration of lysozyme, and max of BSA.

The scouting experiments were planned to find an optimal pH and salt concentration so that the proteins would co-elute. A full factorial design with two factors, pH and salt concentration, pH was varied at 3 levels and the salt concentration at 2 levels. Absorbance was measured at wavelengths 280 nm, 254 nm 320 nm since no response was registred at wavelengths around 500 nm. Due to the fact that the proteins are not co-eluting in a neutral pH the experiments was designed so that the proteins would co-elute based on the isoelectric points of Lysozyme, BSA and IgG the pH-levels was 8, 10 and 11.5 which did not seam to be outside of the pH constraints of the column.

A three-level D-optimal design with three center points was used for the final experiments. Based on scouting experiments the pH of buffer solutions was 11.5 and the salt concentration in elution buffer was 500 mM. The proteins were weighed and mixed into the buffer solution and then mixed in different ratios for the samples. The total protein concentration was constant at 1.8 g/L and the ratios varied. The experimental design is shown in Appendix A.1. The experiments were performed in a randomized order during the course of three days. In order to divide the test into two groups a dummy factor was used.

3.2.2 Data analysis

The experimental data matrix was constructed so that each row represented a run and the columns consisted of the chromatograms for each measured wavelength. The response matrix consisted of corresponding rows and the protein concentrations in columns.

The data set was analyzed using principal component analysis. The principal components was calculated using singular value decomposition and the covariance matrix, the two algorithms used are shown in Appendices A.2 and A.3. The raw data was centered prior to analysis. The RMS-error and the degree of explanation was calculated in order to establish the validity of the principal component analysis.

A new PLS algorithm was written, shown in Appendix A.4 based on the method described in background. The written algoritm uses the input values X, the parameters, and C, the responses to calculate the scores (T), loadings (P and Q) and the weight matrix W. The parameters are the absorbance chromatogram and the responses are the concetrations of proteins. The Root Mean Square-errors were calculated and the predicted values of concentrations was compared to the calculated values.

The execution of the experimental plan was tested using the high heel

factor to test for difference between groups.

The data was analyzed again using the chromatograms at wavelengths 254 nm and 280 nm and the concentration for BSA and Lysozyme.

4 Results and Discussion

4.1 Experimental work

The experimental work was over all failed, mainly due to poor experimental planning. The first round of experiments using a binary protein solution was discarded since the proteins did not co-elute and just went right through the separation column. When a new experimental plan was suggested it was changed to fit a ternary protein solution instead.

The scouting experiments and the theory behind isoelectric points appeared to show that the optimal pH level for this separation was between 11-12. Since the columns could handle pH-levels up to 12 the high pH was assessed to not be a problem. The injection of the protein solutions in scouting experiments was preformed inaccurately since the syringe was removed from the inlet resulting in air pumping thruogh the column hence results could not be trusted. Even though the constraints of the columns were in the pH range of 2-12 this is not equivalent with that the ligand in column is operational at high pH. And therefore no separation of the protein solution was obtained. For the first round of final experiments using BSA, lysozyme and IgG the injections were made in the same manner as for the scouting experiments resulting in chromatograms with only air spikes. Using the same experimental planning the experiments was done again with proper injection. However the experiments were not executed so that the proteins were able to bind to the ligand due to the high pH resulting in that the proteins eluted before gradient started and no separation was possible.In constrast to proteins eluting before gradient the pressure alarm went off for the final four experiments, number 14,7,8 and 3 in A.1, which raised the question whether one of the proteins did not elute at all and accumulated i column. When the column was rinsed through with sodium hydroxide no proteins were detected.

4.2 Data analysis

The data analysis was found unsuccessful since the absorbance measurements made at 320 nm did not contain any information which is visible in Figure 1 and Figure 2. Figure 1 shows the first run which in the experimental design, Figure A.1, is equivalent to experiment number 4. Figure 2 shows the final

run which in the experimental design, Figure A.1, is eqvivalent to experiment number 3.

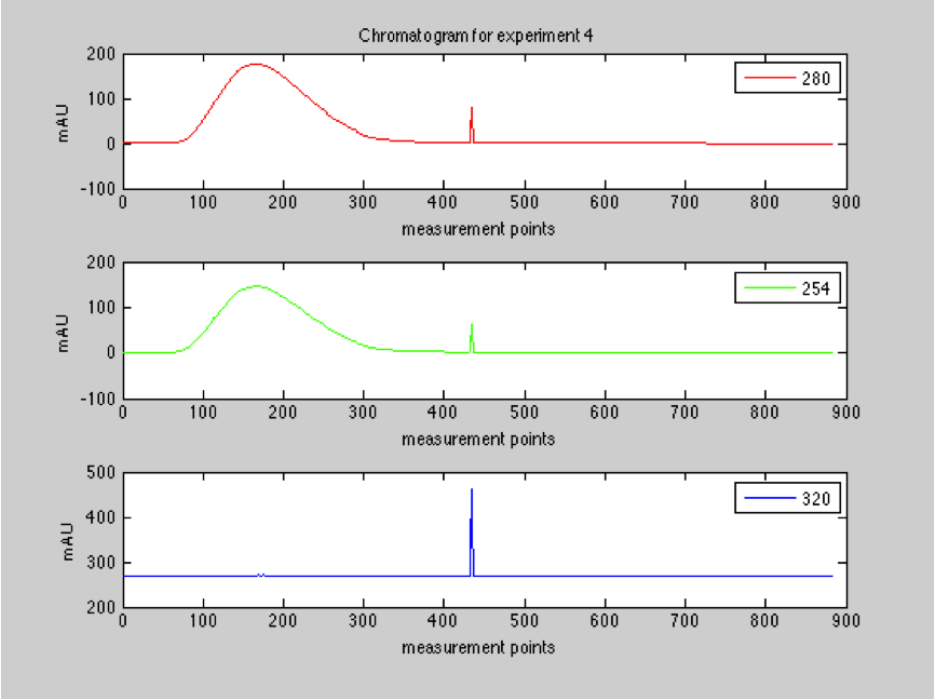

Figure 1: The chromatogram for the three different wavelengths for the first run, experiment 4 according to experimental design.

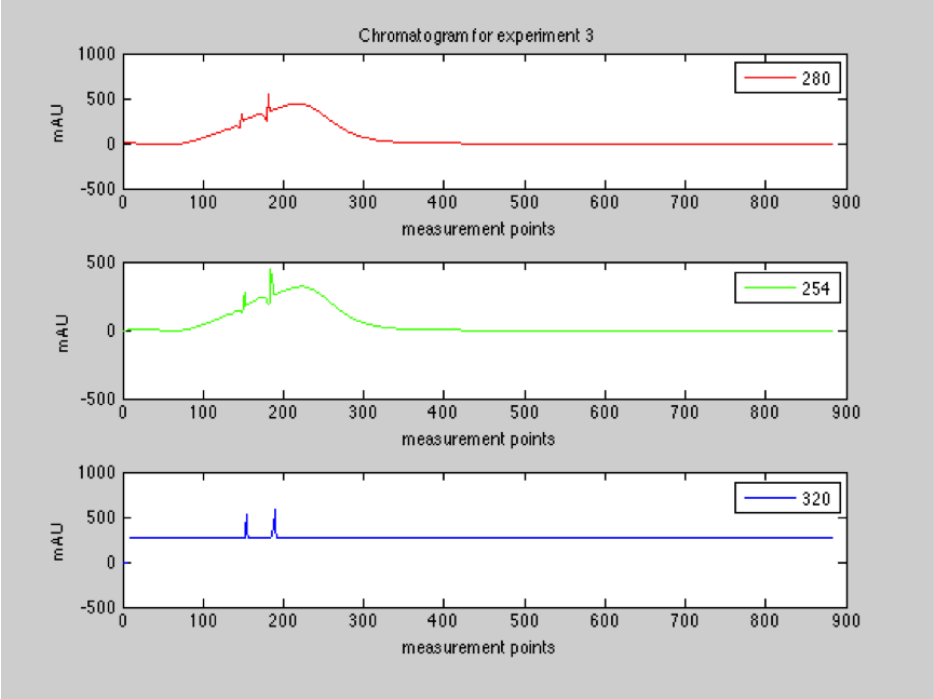

Figure 2: The chromatogram for the three different wavelengths for the last run, experiment 3 according to experimental design.

In the PCA the scores form a vertical line and the loadings forms a clutter close to origo indicating that nothing in the material was significant. The Bi-plot with both scores and loading from PCA with the ternary protein solution is shown i Figure 3. Principal component 1 and 2 has the highest degree of explaination hence no further information can be expected in the pther principal components. The PCA for the binary protein solution give the same result as for the ternary solution.

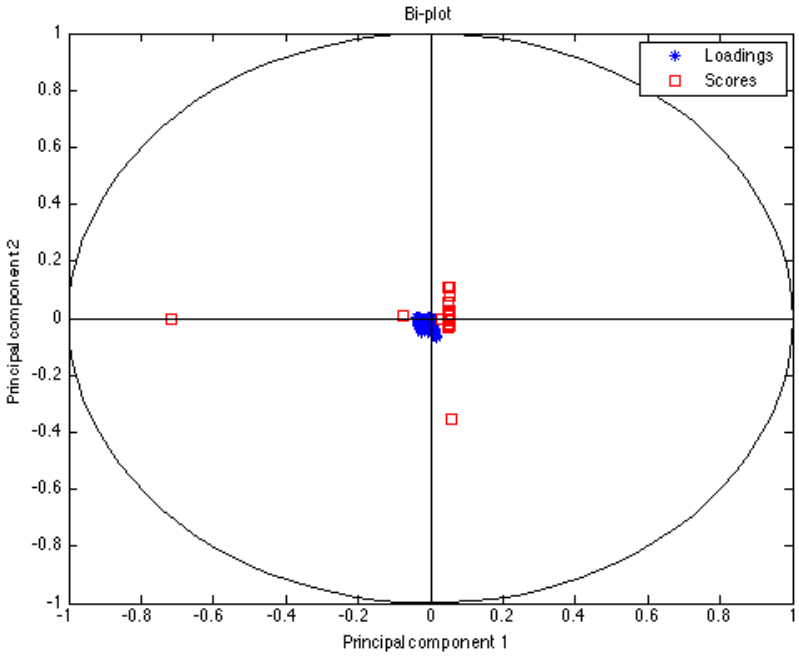

Figure 3: Bi-plot from PCA with ternary protein solution.

Using PLS to create a model between the concentrations in protein solution and absorbances are unsuccesful. The algorithm needs at least as many parameters as responses therefore is is not possibly to predict the concentrations of the three proteins with chromatograms at given wavelengths. When the data was reanalyzed using chromatograms at 280 nm and 254 nm to predict the concentrations of two proteins the results was still inconclusive.

The graphical interpretation of PLS called a W*q-plot is not possible to obtain for neither binary or ternary protein solutions since the data matrices are singular.

The dummy factor set to test for variation between groups and to help determine the magnitude of noise showed no obvious difference between groups. The magnitude of the noise was not established since the result of analysis did not bear any results.

4.2.1 Case

The written algorithm, ownpls2 A.4, was tested using a different data material that previously successfully has been described by the PLS-model. The data material consisted of an NMR-analysis of 40 wines in regards to content

of e.g. ethanol, glycerol, lactic acid, methanol and malic acid. A comparison was made using an algorithm written by Andreas Håkansson for the Chemometric course,FMS210, this algortihm is shown in Appendix A.5. The RMS-error showed no difference between the different algorithms. Figure 4 and 5 shows the RMS-error plotted against number of principal components.

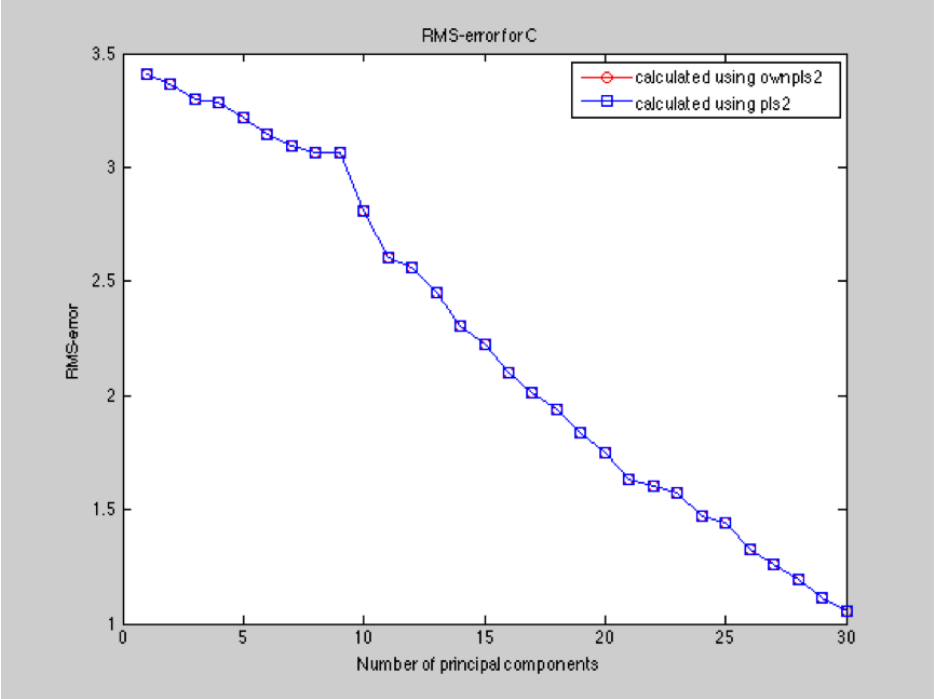

Figure 4: Root Mean Square error calculated for the C-matrix containing 17 chemical properties for the different wines. The RMS-error is calculated using both ownpls2 and pls2.

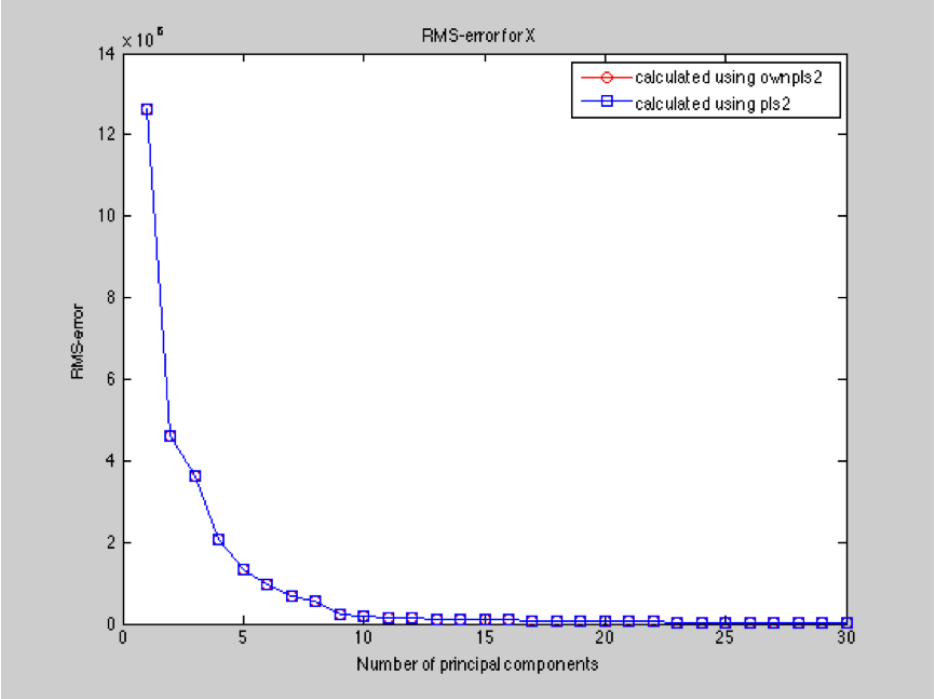

Figure 5: Root Mean Square error calculated for the X-matrix containing the NMR-spectra of 40 different wines. The RMS-error is calculated using both ownpls2 and pls2.

Even though the old and the new algorithm presents the same result there is a big difference in how fast the algorithms are. The difference in speed might not be a problem in calculations for the case study but with bigger matrices the speed of the function matters. The C-matrices in the case study and the protein quantification data are 40×17 respectivly 18×3, and the X-matrix in case study is 40×273 while it for the protein quantification data is 18×2649.

When using pls2, A.5, for the protein quantification data the speed has a great impact because of the larger matrices. The calculations running time makes the old algorithm impractial to use and the need for a faster algorithm is of necessity. Hence a new algorithm had to be written in order to handle the chromatographic data set.Figure 6 shows the calculation time in regards to the number of columns in the X-matrix. It it obvious that there is a big difference in calculation time between ownpls2 ans pls2. When the algorithms was used on the wine data calculations using pls2 took 25.1864 seconds while it only took 0.3505 seconds with ownpls2.

between functions.png

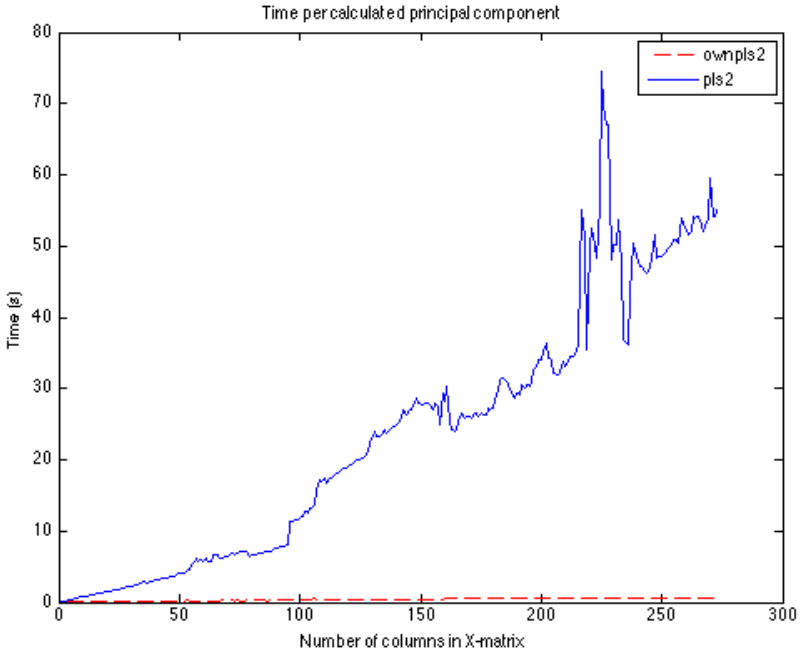

Figure 6: Comparisson between the calculations time for ownpls2 and pls2 in regards to number of columns in X-matrix.

The new algorithm ownpls2, shown in Appendix A.4, tests for convergence as if the component is satisfying i.e. the total variation is under 0.1 % the component is saved and no further iterations are necessary for that particular component and the algorithm starts over and calculated the next component. pls2, shown in Appendix A.5, on the other hand uses 1000 iterations for each calculation of a component which will take longer time.

ownpls2 follows the algorithm presented background while pls2 differs a little bit. The calculated loading matrices P and Q from pls2 are the transponates of the loadings matrices according to the principle of PLS.

A loophole with ownpls2 on the other hand is if the convergence is unsatisfying for more than 1000 iterations ownpls2 will break the loop and simply move on to calculate the next component without saving the unsatisfying one. This might lead to a matrix with to few dimensions. When tested for number of iterations per component the maximum amount using this data set was 165 and the maximum is set to 1000 in the algorithm. This leaves quite a small risk of this problem actually occur however it can be fixed by adding an if-clause saving the vector of the final iteration even though it does

not meet expectations. However this might not be the optimal solution.

The test for convergence was set to be under 0.1% in this case though it does not really appear to affect the result of the calculated scores and loadings in regard to RMS-error. A value even up to 106 is possible before any noticeable chance is observed in the RMS-errors. When the estimated values of responses with the two different algorithms are compared the results show that the predictability of models are equal for both algorithms. Figure 7 shows the predicted values plotted against calculated values for responses using both pls2, A.5, and ownpls2, A.4.

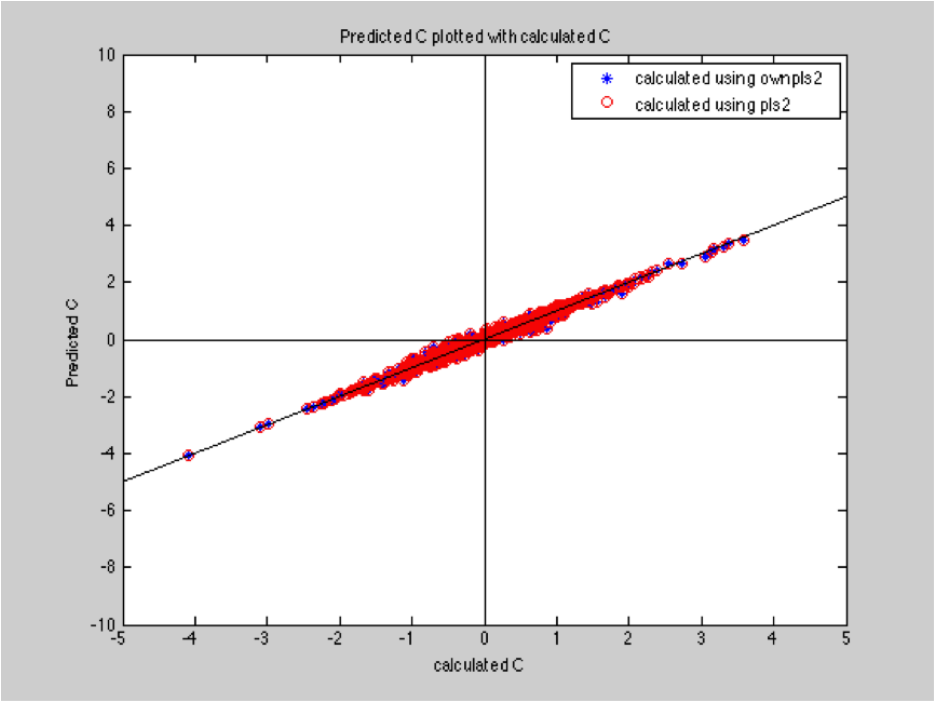

Figure 7: Predicted values of C plotted against the calculated values for C.

4.3 Future Work

Since PLS has been useful in previous studies, (Brestrich et. al, 2014) and (Fonville et. al, 2010), it is an interresting field to continue exploring. The main thing to improve from this thesis is the experimental plan.With an experimental plan using proteins that co-elute in a neutral pH i.e. lysozyme, cythochrome C and Ribonuclease A and measurements at wavelengths that gives a reading a data set could be obtained that is valid to work as a basis for a PLS-model. Using a DAD can assure that the the data set is big enough and

contains enough information to describe a ternary protein solution. Using a different data set to validate the model increases the significance of the model.

5 Conclusion

The experimental work did not give the desired results and therefore protein quantification was not possible. The PLS2-algorithm written in Matlab was found to be functional and fast. PLS has been found useful before in protein quantification previously and using an improved experimental plan to find proper data to use as model basis is promising.

6 Bibliography

Richard G. Brereton, 2003, *"Chemometrics: data analysis for the laboratory and chemical plant"* "upplaga (om ej 1:a uppl.)", West Sussex England, John Wiley Sons Ltd

Nina Brestich, Till Briskot, Anna Osberghaus, Jürgen Hubbuch, 2014, *"A Tool for Selective Inline Quantification of Co-Eluting Proteins in Chromatography Using Spectral Analysis and Partial Least Squares"* "Biotechnology and Bioengineering Vol. 9999, No. xxx", förlagsort, förlag

Richard D. Cramer III, 1993, *"Partial Least Squares (PLS): its strengths and limitations"* "Pespectives in drug discovery and design, ", förlagsort, förlag

Judith M. Fonville, Selena E. Richards, Richard H. Barton, Claire L. Boulange, Timothy M. D. Ebbels, Jeremy K. Nicholson, Elaine Holmes, Marc-Emmanuel Dumas, 2010, *"The evolution of Partial Least Squares models and related chemometric approaches in metabonomics and metabolic phenotyping"* "Journal of Chemometrics Vol. 9999, No. xxx", förlagsort, förlag

Sigrid K. Hansen, Erik Skibsted, Arne Staby, Jürgen Hubbuch, 2011, *"A Label-Free Methodology for Selective Protein Quantification by Means of Absorption Measurements"* "Biotechnology and Bioengineering Vol. 108, No. 11", förlagsort, förlag

A Appendix

A.1 Experimentalplan

The non-randomized experimental plan. The proteins are described as ratios

Experiment	BSA	Lys	IgG
1	0	1	0
2	0,25	0,5	0,25
3	0,33	0,33	0,33
4	0,4	0,2	0,4
5	0,45	0,1	0,45
6	0,5	0	0,5
7	1	0	0
8	0,5	0,25	0,25
9	0,3 3	0,33	0,33
10	0,2	0,4	0,4
11	0,1	0,45	0,45
12	0	0,5	0,5
13	0	0	1
14	0,25	0,25	0,5
15	0,33	0,33	0,33
16	0,4	0,4	0,2
17	0,45	0,45	0,1
18	0,5	0,5	0

A.2 ownprincomp

```
%OWNPRINCOMP uses singular value factorization to calculate P, the loading
%matrix, and uses..... to calculate T, the score matrix. Singular value
%factorization calulates Xh(X*), the hermitian conjugate, multiplied with
%X, the data matrix, to create a qudratic matrix. The eigenvalues of Xh*X
%are calculated and determines the order of the principal components. The
%eigen vectors of the Xh*X-matrix are the principal components and the
%first component is the one with the largest eigen value.

% The hermitian conjugate are the transponate of the matrix and then the
% signs of the complex parts of the matrix are changed. In reality the
% hermitian conjugate will be the same as the transponate since the
% X-matrix will not contain any complex numbers.
[m,n]=size(X);
A=zeros(size(X));
%center X with a for-loop using the size of the matrix X.
for k=1:n
    A(:,k)=X(:,k)-mean(X(:,k));
end
%A is the new centered version of the data-matrix
Aherm=A'; %the hermitian conjugate
AhermA=Aherm*A;
[eigenvectors,eigenvalues]=eig(AhermA);

[sortedeigenvectors, sortedeigenvalues] = sortem(eigenvectors, eigenvalues);

%the eigenvalues are made into a vector
% and sorted in descending order
%latent=sort(diag(eigenvalues),'descend');
latent = diag(sortedeigenvalues);
%sort the eigen vectors so that the eigen values decreases for each column
%sortedeigenvectors= eigenvectors(end:-1:1);
P=sortedeigenvectors;
%the score-matrix is calculated by multiplying the data-matrix with the
%loading-matrix
T=A*P;
%Hotellings Tsquare
%sort the eigenvalues-matrix in descending order
%sortedeigenvalues= eigenvalues(end:-1:1);
Tsquare=zeros(length(X(:,1)),1);
for k=1:m
   Tsquare(k,:)=T(k,:)*inv(sortedeigenvalues)*transp(T(k,:));
end
end
```

A.3 ownpcacov

```
%OWNPCACOV uses the covariance matrix to calculate the principal components
% Returns loadings (P), latent, and explained
% input X matrix
%Sofia Henryson
[m,n]=size(X);
A=zeros(size(X));
%center X with a for-loop using the size of the matrix X.
for k=1:n
    A(:,k)=X(:,k)-mean(X(:,k));
end

%A is the new centered version of the data-matrix
Aherm=A'; %the hermitian conjugate
%The covariance matrix,C, is calculated using the centered data and the
%hemritian conjugate.
% A is an m*n-matrix.

C=(1/(m-1))*(Aherm*A);

%calculate the eigenvectors and eigenvalues of the covaraince-matrix
[eigenvectors,eigenvalues]=eig(C);
%sort the eigen vectors so that the eigen values decreases for each column

%sort the eigenvalues-matrix in descending order
% sort the eigenvectors-matrix in descending order in regard to eigenvalues
[sortedeigenvectors, sortedeigenvalues] = sortem(eigenvectors, eigenvalues);

%The principal components are the eigenvectors of the covariance matrix

P=sortedeigenvectors;
%latent=sort(diag(eigenvalues),'descend'); % output, the sorted eigenvalues
latent = diag(sortedeigenvalues);

% calculate the explanation of each PC
%add all the eigenvalues together to get the total variance
latenttot=sum(abs(latent));

ex=zeros(length(X),1);

for k=1:n
    ex(k,1)=abs(latent(k))/latenttot;
end

% expalained gives the degree of explanation of each principal component
explained=ex;
end
```

A.4 ownpls2

```
%Uses Partial Least Square regression to calculate score and loading
%matrices. Uses X and C to return the scores- (T) loadings- (P) (Q) and
    weight matrix- (W) in the model:
%X=T*P'+E
%C=T*Q'+F
% Sofia Henryson 2014
    disp('Start: ownpls2...')
    X=Xorg;
    C=Corg;
    %Initiates the matrices
    T = [];
    P = [];
    Q = [];
    W = [];

    maxIter = 1000; %maximum number of iterations
    a=0.001; %limit for acceptable convergance
    Chat = zeros(size(Corg)); %prelocate Chat
    for i=1:length(X)
        u=C(:,1); %Step 3
        tinit=zeros(size(Xorg(:,1)));
         %algoritm follows Breretons A.2.3.
        for j=1:maxIter
            h=X'*u;    % Step 4
            that=X*h/(sum(h.^2).^.5); %step 5
            phat=that'*X/sum(that.^2); % step 6
            qhat=C'*that/sum(that.^2); %step 7
            w=h/(sum(h.^2).^.5);

            %calculate a new u if i=1 or if the residual quadratic sum is
                too big
             if (j==1) || (norm(tinit-that)>a) %check for convergence
                 u=C*qhat/sum(qhat.^2);
                 tinit = that;
             else %save the values in matrices
                T(:,i)=that;
                P(:,i)=phat';
                Q(i,:)=qhat;
                W(i,:)=w;
                 break;
             end
        end

        Xresid=X-that*phat; % removes the effect of the new pls-component
        Chat=Chat+that*qhat'; %determines a new concentration estimate
        Cresid=Corg-Chat; %removes the effect of the new pls-component
        X=Xresid; %replace X with Xresid
        C=Cresid; %replace C with Cresid
    end
    disp('Finished: ownpls2');
end
```

A.5 pls2

```
function [T,P,Q,W] = pls2(X,C)
% Beräknar matriserna T, P och Q så att
%   X = T*P' + E och
%   C = T*Q' + F
% (E och F små)
% X har storlek (I*J) och C (I*N).
% W är viktsmatrisen.
% Centrering och standardisering
% av X och C måste ske före denna funktion.
% (Stegen i kommentarerna hänvisar till
% algoritmen i Brereton A.2.3)
% Andreas Hellqvist, 2011
T=[]; P=[]; Q=[]; W=[]; %Initierar T, P och Q
NItter = 1000;    %Maximala antalet itterationer per komponent
if size(XOrg,1) ~= size(COrg,1)
    error('Fel dimensioner på X och C, se help pls2 för instruktioner')

end
X=XOrg; C=COrg;
CHatt=zeros(size(C)); %Gissar CHatt som bara nollor
for j=1:size(X,2) %Beräknar komponenterna en efter en
    u=C(:,1);     %Gissar u som C                    %Steg 3
    for i=1:NItter
        h=X'*u;                                     %Steg 4
        w = h/(h'*h)^0.5;
        %tHatt=X*h/(h'*h)^0.5;                       %Steg 5
        tHatt = X*w;
        pHatt=tHatt'*X/(tHatt'*tHatt);              %Steg 6
        qHatt=C'*tHatt/(tHatt'*tHatt);              %Steg 7
        u=C*qHatt/(qHatt'*qHatt);                   %Steg 8
        tHattGammal=0;
        %Kontrollerar för kovergens                  Steg 9
        % (här används 0.5% av den totala variationen som gräns för att
            bryta)
        if (tHatt-tHattGammal)'*(tHatt-tHattGammal)/(tHatt'*tHatt)<0.005 &&
            i>1
            T=[T,tHatt];
            P=[P,pHatt'];
            Q=[Q,qHatt];
            W=[W,w];
            break;
        end
        tHattGammal=tHatt; %tHatt som den beräknades i förra ittereringen
        if i == NItter %Om ittereringen inte konvergerat tillräckligt
            T=[T,tHatt];
            P=[P,pHatt'];
            Q=[Q,qHatt];
            W=[W,w];
        end
    end
    X     = X-tHatt*pHatt;                          %Steg 10
    CHatt= CHatt+tHatt*qHatt';                      %Steg 11
    C     = COrg-CHatt;                             %Steg 11
    %Residualkvadratsumma
end
```