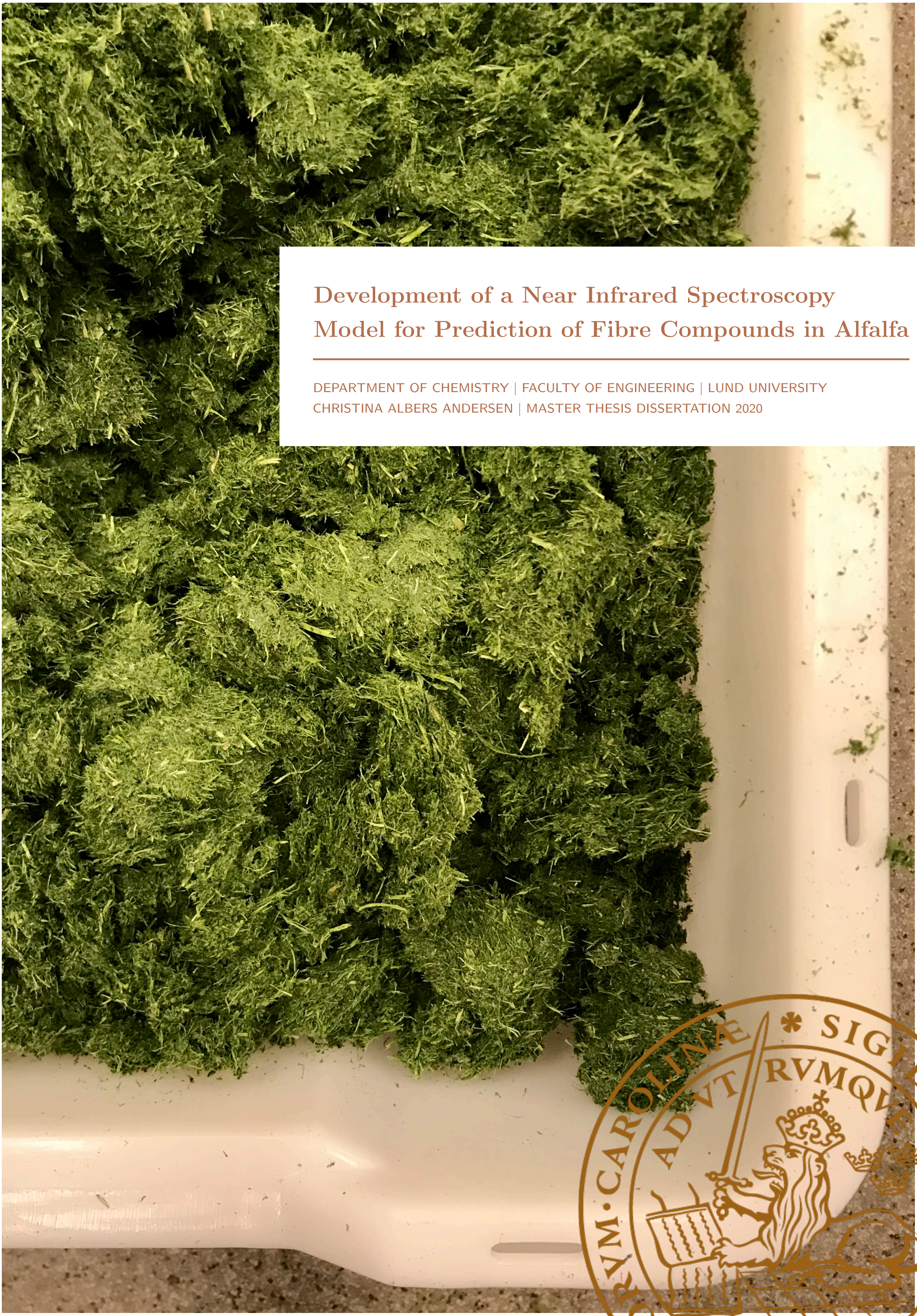


Development of a Near Infrared Spectroscopy Model for Prediction of Fibre Compounds in Alfalfa

DEPARTMENT OF CHEMISTRY | FACULTY OF ENGINEERING | LUND UNIVERSITY
CHRISTINA ALBERS ANDERSEN | MASTER THESIS DISSERTATION 2020

Development of a Near Infrared Spectroscopy Model for Prediction of Fibre Compounds in Alfalfa

M.Sc. Degree Project in Applied Microbiology
Course Code: KMBM05
30 ECTS

August 19, 2020

By
Christina A Andersen

Supervisor:	
Jenny Schelin	Associate Professor and Senior Lecturer The Faculty of Engineering, Lund University
Assistant supervisors:	
Peter Rudahl Jensen	Professor and Head of Research Group The Technical University of Denmark
Mikkel Hansen	PhD Student The Technical University of Denmark
Examinator:	
Ed van Niel	Associate Professor and Senior Lecturer The Faculty of Engineering, Lund University

Cover photo: Alfalfa pulp derived from protein powder production. Photo: Christina A Andersen, 2020
Published by: Lund University

POPULAR SCIENTIFIC ARTICLE

Investigation of the Fibre Content in the Promising New Food Ingredient, Alfalfa Protein Powder

Christina A Andersen

Sustainable food is a key topic becoming more important with time. Trends within the food sector are pointing in directions of locally grown plant based food solutions, but consumers do not want to compromise regarding nutritional qualities. In Denmark, the climate is perfectly suited for cultivation of alfalfa, a legume also known by lucerne and Medicago sativa. For centuries harvested alfalfa has been used as feed for cows being able to digest a large amount of fibres within alfalfa. Alfalfa does however also contain greater amounts of essential amino acids than for example the popular food ingredient soy, making alfalfa an interesting subject for research within the area of potential human food resources. In order to make the best use of the wanted and advantageous amino acids within the fibrous alfalfa, as much of the protein content as possible needs to be extracted from the legume. This is done by pressing harvested and wetted stems, leaves and flowers of alfalfa into a protein rich green juice, and a fibrous pulp, see Figure 1. The pH of this green juice is decreased to lower the water solubility of the wanted proteins, and thus precipitate them. The green juice is then centrifuged resulting in a pellet containing proteins amongst other alfalfa compounds. This pellet is freeze dried into a protein powder in order to concentrate the wanted proteins. This process is performed 10 times in total, the first time untreated raw wetted alfalfa is pressed into green juice as mentioned, the following nine times the fibrous pulp from the prior press is wetted and pressed into more green juice. The aim of re-pressing the fibrous pulp is to extract the highest total amount of protein from one batch of alfalfa. This protein powder production from raw untreated alfalfa to protein powder, does increase human digestibility of alfalfa by increasing the amount of protein per weight.

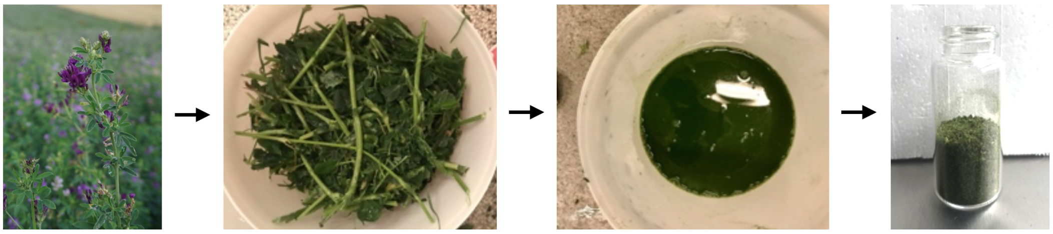


Figure 1 The protein powder production from raw untreated alfalfa to protein powder. Harvested and wetted stems, leaves and flowers of alfalfa are pressed into a protein rich green juice, which is further processed to a freeze dried protein powder. Photos: Jonas M Thomasen [1] and Christina A Andersen, 2020.

The produced protein powder might have potential to be used as a food ingredient on the market when its compound composition has been further outlined. The compound composition also has to be outlined in order to extract and use the protein from alfalfa in the most cost efficient and sustainable way. Due to the composition of raw alfalfa, dietary fibres are suspected to be found in high amounts in the protein powder. Since dietary fibre determination with traditional chemical analysing methods is very time consuming, this project investigates the fibre fractions of alfalfa protein powder, and a potential method for rapid determination of the fibre content.

In this project, the possibility to develop a model for determination of the amount and type of fibre fractions within the protein powder produced from alfalfa in a fast and cheap manner without performing traditional chemical experiments, is thus investigated. The model will be developed from near infrared (NIR) spectra of the protein powder related to enzymatically determined nutrient contents of the protein powder. NIR spectra are fingerprints of a given food sample, representing the unique physical and chemical composition of it, because the measured spectra reflect the amount of certain molecular bonds in various types of molecules, such as fibre molecules. Since NIR spectra are affected by all compounds of an analysed sample, protein, fibre, carbohydrate and ash contents are determined for each of the 10 presses leading to produced protein powder. These results are also used with the purpose of outlining the protein powder contents in general. The NIR spectra of the protein powder are preprocessed in order to produce the best correlation between the spectra and the measured fibre contents. When the best suited preprocessing methods are found, a model being able to predict the fibre content in the 10 different samples was successfully developed. The model is not validated, and therefore it is challenging to draw a conclusion regarding the model quality. For higher chances of success, and in order to produce a more robust model, big datasets, and independent validation sets are required. The results of this project do, however, encourage further investigation and optimisation of this kind of model development.

MASTER THESIS DISSERTATION

Development of a Near Infrared Spectroscopy Model for Prediction of Fibre Compounds in Alfalfa

Christina A Andersen

Abstract

Background: This project investigates if it is possible to develop a calibration model from near infrared (NIR) spectroscopic measurements, for determination of the amount and type of fibre fractions within protein powder produced from the legume alfalfa, without performing wet experiments. Alfalfa is also known as Medicago sativa and lucerne, but is in this project further referred to as alfalfa. Such a model would be applicable as a protein powder production process control, by scanning a small amount of sample during the production process, immediately resulting in a fibre content value. With this result, one will know when the process should be stopped by means of nutritional values. Except from fibres, alfalfa contains large amounts of nutrients, for example essential amino acids. The advantageous amino acids are thus extracted from the fibrous alfalfa during the protein powder production process.

The alfalfa protein powder is produced from stems, leaves and flowers of intact, freshly harvested alfalfa plants. The raw alfalfa was frozen during storage, then thawed and wetted prior to the first press, which is resulting in a protein rich green juice, and a fibrous pulp. The pH of the green juice is decreased to precipitate proteins. The green juice is then centrifuged resulting in a pellet consisting of the total water soluble solid content extracted from alfalfa. The pellet is freeze dried into a protein powder in order to concentrate the protein content. This process is performed 10 times in total, the first time untreated raw wetted alfalfa is pressed into green juice as mentioned, the following nine times the fibrous pulp from the prior press is wetted and pressed into new samples of green juice. The aim of re-pressing the fibrous pulp is to extract the highest total amount of protein from one batch of alfalfa.

This protein powder production from raw untreated alfalfa to protein powder, does increase human digestibility of alfalfa by increasing the amount of protein per weight. Protein powder derived from each of the 10 presses was collected in separate fractions to determine to which extent the fibre profile is changing using an enzymatic gravimetric method. The amounts of protein, insoluble dietary fibre (IDF), soluble dietary fibre (SDF), total dietary fibre (TDF), available carbohydrates (ACH) and ash were determined, since NIR spectra are affected by all compounds of the protein powder. NIR spectra from all 10 presses are related directly to the determined TDF contents, which are used as reference values in order to calibrate a partial least squares (PLS) model that produces predicted TDF values.

Attempts were also made to conduct NIR spectra earlier in the protein powder production process, from the green juice prior to centrifugation and from the pellet prior to freeze drying. A cellulose gluten powder dilution series comparable to the 10 presses of protein powder was prepared, to test if a calibration model could be developed from NIR spectral data of powder containing cellulose as one of the main components. The cellulose gluten spectra were also compared with protein powder spectra during spectral compound analyses.

Results: The nutrient profile determination resulted in a total decreasing amount of protein from 43.12% w/w for press 1 to 37.84% w/w for press 10. The TDF content increased from 22.80% w/w for press 1 to 47.47% w/w for press 10. ACH decreased from 5.43% w/w for press 1 to 1.10% w/w for press 10, while the amount of determined ash decreased from 8.24% w/w for press 1 to 2.70% w/w for press 10. Usable and promising NIR spectra were conducted from all measured protein and cellulose gluten powder samples. A calibration model predicting TDF contents for each of the 10 presses was developed with a wavenumber range from 6,800 cm^{-1} to 4,100 cm^{-1} and R^2 = 0.98. For all 10 presses, the mean deviation from the reference TDF contents was 0.76% w/w. NIR spectra from the green juice and pellet could not be conducted with the available NIR instrument and presetting options.

Conclusions: It is challenging to convert complex NIR spectra into usable information. Since a broad wavenumber spectrum was chosen for the model development, it was easy to fit the spectra to almost any kind of reference values, even though the spectra do not describe those reference values. It also has to be kept in mind that the model is not validated. Therefore it is hard to draw conclusions regarding the model quality. It can be concluded though, that NIR spectra obtained from the protein powder of alfalfa look promising for further investigation, since a good correlation between the TDF amounts and NIR spectra could be seen. Of future work the first priority should be to validate this produced model. If that looks promising, both a new independent validation set and a larger data set to produce a new calibration model is required to further test the model robustness.

Svensk Sammanfattning

Bakgrund: Detta projekt undersöker om det är möjligt att utveckla en kalibreringsmodell utifrån spektroskopiska mätningar i det nära infraröda (NIR) området, för bestämning av mängden och typen av fiberfraktioner i proteinpulver producerat från baljväxten alfalfa, utan att utföra våtexperiment. Alfalfa benämnas även Medicago sativa och lucerne, men kallas inom detta projekt alfalfa. En sådan typ av modell skulle kunna tillämpas som en kontroll av proteinpulvrets produktionsprocess, genom att skanna en liten mängd prov från ett steg i produktionsprocessen, vilket omedelbart resulterar i ett fiberinnehållsvärde. Med ett sådant resultat får man reda på när processen ska stoppas enligt näringsvärdena. Förutom fibrer, innehåller alfalfa en hög andel essentiella aminosyror. Aminosyrorna skall därför gärna extraheras från den fibrösa alfalfa under produktionen av proteinpulver.
Proteinpulvret produceras från stjälkar, blad och blommor av intakta, nyskördade alfalfa baljväxter. Den råa alfalfa har av lagringsskäl frysts ner. Frusen alfalfa har därför tinats, fuktats och blivit pressad med en skruvpress, vilket resulterade i en proteinrik grön juice och en fibrös massa. pH-värdet i den gröna juicen sänktes för att fälla ut proteiner. Den gröna juicen centrifugeras sedan, vilket resulterade i en pellets bestående av det totala vattenlösliga fasta innehållet extraherat från alfalfa. Pelleten frystorkades till ett proteinpulver för att koncentrera upp proteininnehållet. Denna process utfördes totalt 10 gånger, första gången med rå alfalfa som pressades till grön juice, följande nio gånger återfuktades den fibrösa massan från den tidigare pressen och pressades sedan till nya separata prover av grön juice.
Syftet med att pressa den fibrösa massan från den tidigare pressen, är att extrahera den högsta totala mängden protein från en batch av alfalfa. Detta sätt att producera proteinpulver på, ökar smältbarheten av alfalfa genom att öka mängden protein per vikt. Proteinpulver från var och en av de 10 pressarna uppsamlas i separata fraktioner för att bestämma i vilken utsträckning fiberprofilen förändrades med användning av en enzym metod. Mängderna protein, olösliga kostfibrer (IDF), lösliga kostfibrer (SDF), totala kostfibrer (TDF), tillgängliga kolhydrater (ACH) och ask bestämdes, eftersom NIR-spektra påverkas av alla föreningar i proteinpulvret. NIR-spektra från alla 10 pressar relaterades direkt till det bestämda TDF-innehållet, som används som referensvärden för att kalibrera en partial least squares (PLS) modell, som i sin tur skall producera förutsagda TDF-värden.
Försök gjordes också för att möjliggöra mätning av NIR-spektra tidigare i proteinpulvrets produktionsprocess. NIR-spektra från den gröna juicen innan centrifugering, och från pelleten före frystorkning försöktes mätas. En utspädningsserie av cellulosa-glutenpulver jämförbar med de 10 pressarna av proteinpulver framställdes för att testa om en kalibreringsmodell kunde utvecklas utifrån NIR-spektra med pulver innehållande cellulosa som en av huvudkomponenterna. Cellulosa-glutenspektra jämfördes också med proteinpulver spektra för att jämföra förekomsten av kemiska föreningar i spektrumen.

Resultat: Bestämning av näringsprofilen resulterade i en total minskande mängd protein från 43,12% w/w för press 1 till 37,84% w/w för press 10. TDF-innehållet ökade från 22,80% w/w för press 1 till 47,47% w/w för press 10. ACH minskade från 5,43% w/w för press 1 till 1,10% w/w för press 10, medan mängden bestämd ask minskade från 8,24% w/w för press 1 till 2,70% w/w för press 10. Användbara och lovande NIR-spektra togs fram för alla uppmätta prover för både proteinpulver och cellulosa-glutenpulver. En kalibreringsmodell som producera förutsagda TDF-värden för var och en av de 10 pressarna utvecklades från vågtalet 6.800 cm^{-1} till 4.100 cm^{-1} med $R^2 = 0{,}98$. För alla 10 pressar var den genomsnittliga medelavvikelsen från referens-TDF-innehållet 0,76% w/w. Uppmätta NIR-spektra med det tillgängliga NIR-instrument och dess förinställningsalternativ från grön juice och pellet kunde inte användas.

Slutsatser: Det är svårt att konvertera komplexa NIR-spektra till användbar information. Eftersom ett brett vågtalsspektrum valdes för modellutvecklingen, är det enkelt att anpassa spektra till nästan alla typer av referensvärden, även om spektrumen inte beskriver just dessa referensvärden. Modellen är inte validerad, och det är därför svårt att dra slutsatser angående modellkvaliteten. Slutsatsen att NIR-spektra erhållna från proteinpulvret i alfalfa ser lovande ut för ytterligare undersökningar, kan dock dras, eftersom en god korrelation mellan TDF-värdena och NIR-spektrumen kunde ses. För framtida arbete bör första prioritet vara att validera denna producerade modell. Om det ser lovande ut, krävs både en oberoende validering av denna modell, och sedan en större datauppsättning för att producera en ny kalibreringsmodell, för att öka modellens robusthet.

Dansk Resumé

Baggrund: Dette projekt undersøger om det er muligt at udvikle en kalibreringsmodel ud fra spektroskopiske målinger i det nær infrarøde (NIR) område, for at bestemme mængden og typen af fiberfraktioner i proteinpulver produceret fra bælgplanten alfalfa, uden at udføre klassiske kemiske laboratorieeksperimenter. Alfalfa, også kendt under navnene Medicago sativa og lucerne, vil fortsat refereres til som alfalfa. En sådan model kan anvendes som proceskontrol ved at scanne en lille mængde proteinpulverprøve under produktionsprocessen, hvilket umiddelbart vil resultere i en fiberindholdsværdi. Dette resultat vil give en indikation af, hvornår processen skal stoppes i forhold til proteinpulverets næringsværdier. Udover fibre indeholder alfalfa en høj andel essentielle aminosyrer. Disse essentielle aminosyrer ekstraheres fra den fiberholdige alfalfa under produktionsprocessen af proteinpulver.

Proteinpulveret er produceret af stængler, blade og blomster fra intakte, friskhøstede alfalfa-planter. Friskhøstet alfalfa blev nedfrosset for at muliggøre en længere opbevaringstid. Ved projektets start, blev den frosne alfalfa tøet op, fugtiggjort og presset til en proteinrig grøn juice med en skruepresse, hvilket desuden resulterede i en fiberholdig grøn pulp. pH i den grønne juice sænkes for at udfælde proteinerne. Derefter centrifugeres den grønne juice, hvilket resulterer i en pellet bestående af det samlede vandopløselige faste indhold ekstraheret fra alfalfa. Pelleten frysetørres til et proteinpulver for at opkoncentrere de ønskede proteiner. Denne proces udføres i alt 10 gange. Første gang udføres den med optøet ubehandlet alfalfa, og de følgende ni gange med den fiberholdige grønne pulp fra det forudgående pres, der fugtiggøres og presses til nye prøver af grøn juice.

Målet ved at genpresse den fiberholdige grønne pulp, er at ekstrahere den højeste samlede mængde protein fra en portion presset alfalfa. Denne slags proteinpulverproduktion øger fordøjeligheden af alfalfa ved at øge mængden af protein pr. vægt. Proteinpulver afledt fra hver af de 10 pres blev opsamlet i separate fraktioner for at bestemme i hvilket omfang fiberprofilen ændrede sig ved anvendelse af en enzymatisk gravimetrisk metode. Mængderne af protein, uopløselige kostfibre (IDF), opløselige kostfibre (SDF), totale kostfibre (TDF), tilgængelige kulhydrater (ACH) og aske blev undersøgt, da NIR-spektre påvirkes af hele proteinpulverets indhold. NIR-spektre fra alle 10 pres blev relateret direkte til det undersøgte TDF-indhold, der bruges som referenceværdier for at kalibrere en partial least squares (PLS) model, der forudsiger TDF-værdier.

Det blev også forsøgt at muliggøre måling af NIR-spektre tidligere i proteinpulverproduktionsprocessen. Der blev forsøgt at måle NIR-spektre af den grønne juice før centrifugering, og på pelleten før frysetørring. En fortyndingsserie med celluloseglutenpulver sammenlignelig med proteinpulver fra de 10 pres af alfalfa, blev fremstillet for at teste, om der kunne udvikles en kalibreringsmodel ud fra NIR-spektre målt på pulver med cellulose som en af hovedbestandsdelene. Celluloseglutenspektrene blev også sammenlignet med proteinpulverspektrene for at sammenligne tilstedeværelsen af kemiske forbindelser i spektrene.

Resultater: Den enzymatiske næringsprofilbestemmelse resulterede i en total faldende mængde protein fra 43,12% w/w for pres 1 til 37,84% w/w for pres 10. TDF-indholdet steg fra 22,80% w/w for pres 1 til 47,47% w/w for pres 10. ACH faldt fra 5,43% w/w for pres 1 til 1,10% w/w for pres 10, mens mængden af aske faldt fra 8,24% w/w for pres 1 til 2,70% w/w for pres 10. Brugbare og lovende NIR-spektre fra alle målte protein- og cellulosegluten-pulverprøver blev indsamlet. En kalibreringsmodel, der forudsagde TDF-indhold for hvert af de 10 pres, blev udviklet med et bølgetalsområde fra 6.800 cm^{-1} till 4.100 cm^{-1} med $R^2 = 0,98$. For alle 10 pres var den gennemsnitlige middelafvigelse fra reference-TDF-indholdet 0,76% w/w. NIR-spektre med det tilgængelige NIR-instrument og forindstillingsmulighederne fra den grønne juice og pellet var ikke brugbare.

Konklusioner: Det er svært at konvertere komplekse NIR-spektre til brugbar information. Eftersom der blev valgt et bredt bølgetalspektrum til modeludviklingen, er det potentielt set let at tilpasse spektrene til enhver form for referenceværdier, selvom spektrene ikke beskriver disse referenceværdier. Da modellen ikke er valideret er det svært at drage endelige konklusioner vedrørende modelkvaliteten. Det kan imidlertid konkluderes, at de producerede NIR-spektre fra alfalfa proteinpulveret ser lovende ud i forhold til igangsættelse af relevante yderligere undersøgelser, da en god sammenhæng mellem TDF-værdierne og NIR-spektrene kunne ses. Ved et eventuelt fremtidigt arbejde bør første prioritet være at validere denne model. Hvis det ser lovende ud, kræves både en uafhængig validering og herefter et større datasæt for at producere en ny kalibreringsmodel med en øget modelrobusthed.

Preface

This thesis has been prepared at the National Food Institute at the Technical University of Denmark, DTU, for the degree Master of Science in Engineering, M.Sc. Eng.

It is assumed that the reader has a basic knowledge in the areas of chemistry and food science.

The following software is used throughout the project: LaTeX as text editing program, MATLAB, Unscrambler and MS Excel as data handling programs, MS Word for creating figures and Mendeley Desktop for reference handling.

Further, the thesis follows the guidelines of the journal Biotechnology for Biofuels based on the LaTeX template *BioMed Central Tex Template v1.06* retrieved from *www.biotechnologyforbiofuels.biomedcentral.com*. Relevant changes are made to fit the master thesis setup according to Lund University standards.

Finally I would like to thank my supervisor at Lund University, Jenny Schelin for her help and guidance through my thesis project. I also would like to thank my assistant supervisors at the Technical University of Denmark (DTU), Peter Rudahl Jensen and Mikkel Hansen for letting me be part of their project, and for all kinds of practical help and guidance in the labs. Furthermore, I am really grateful for each of many questions Bo Munk Jørgensen, Heidi Olander Petersen, Timothy John Hobley and Ida Kallehauge Nielsen has helped me answer during my time at DTU. And last, but not least, the time spend at DTU would not have been as fun without co-student Therese Mollwitz Heltved.

Finally, I would like to thank all my friends and family for believing in and supporting me throughout my degree.

Contents

1 Introduction

Sustainable food is a key topic becoming even more important with time. InnoGrass is a highly relevant newly started project at the Technical University of Denmark, amongst other partners, with professor Peter Ruhdal Jensen as project leader, looking at a sustainable use of proteins from the green biomass Medicago sativa, a legume also known by lucerne and alfalfa. In this project, alfalfa is used for further references.

With the use of protein powder produced from raw alfalfa, InnoGrass wants to enrich plant based foods and be competitive considering comparable plant protein sources, especially regarding amino acid profiles. Alfalfa shows an amino acid profile similar to that of milk and meat, which with its low environmental impact makes it a competitive protein source on the market. Seen from a food ingredient perspective, it is also a cheap resource [2].

The protein powder, also suspected to contain high amounts of dietary fibre, is not yet approved by the European Food Safety Authority (EFSA) to be used as human food and is therefore still defined as novel food. Novel food is defined as food that had not been humanly consumed in the European Union (EU) before May 1997, where the first novel food regulation was developed. Novel food is more commonly known as newly developed innovative food, which chia seeds and UV treated milk are examples of [3].

During the production process from raw untreated alfalfa to protein powder, which is applied to increase human digestibility, as much of the high value protein as possible should be extracted. Thus a high amount of dietary fibre is seen as being negative. But dietary fibres still account for a large part of the protein powder, and are not only considered negative related to human health benefits. In order to extract and use the protein in the most cost efficient and sustainable way, the protein powder compound composition has to be outlined. Since dietary fibres are suspected to be found in high amounts due to the plant composition, and since dietary fibre determination with traditional chemical analyses is very time consuming, this project investigates the fibre fractions of alfalfa protein powder.

Protein powder derived from alfalfa is not only interesting for future human consumption, but also as a highly relevant sustainable feed source as protein supplement in Denmark eventually replacing imported soy [4].

1.1 Overall Aim, Specific Objectives and Hypothesis

The overall aim of this project is to develop a model for determination of the amount and type of fibre fractions within the protein powder produced from alfalfa in a fast and cheap manner without performing wet experiments, if that is found to be possible. The model will be developed from near infrared (NIR) spectra related to enzymatically determined nutrient contents of the protein powder.

The specific objectives are to produce protein powder from alfalfa, determine the nutrient profile of the protein powder, and relating the nutrient profile to NIR spectra from different steps of the protein powder production process, by producing a calibration model. By relating the nutrient profile to NIR spectra, a model for determination of the total dietary fibre (TDF) content could probably be made.

The protein powder production process from raw alfalfa is roughly divided into three steps, see Figure 2. After each of the first nine presses, the pulp is rewetted and re-fed into the screw press with the purpose of extracting as much protein from a batch of alfalfa as possible.

Previous determinations of TDF within food samples using preprocessed NIR spectra have shown promising results. One example by Kim et al, is the determination of TDF in homogenised meals containing similar amounts of protein as the protein powder analysed in this project [5]. The same enzymatic reference method as in this project was used. With that knowledge, the hypothesis for this project is that NIR spectra of the protein powder will be usable and could be related to the TDF content.

1.2 Applications for a Fibre Compound Prediction Model

Determination of fibre fractions in alfalfa and the extent of them is essential for getting a novel food product on the market with regards to the rate of digestion, and for labelling purposes [6, 7]. A routine method for determination of the fibre fractions, being both sustainable, time and cost efficient, would thus be ideal. Fibres of many plant tissues

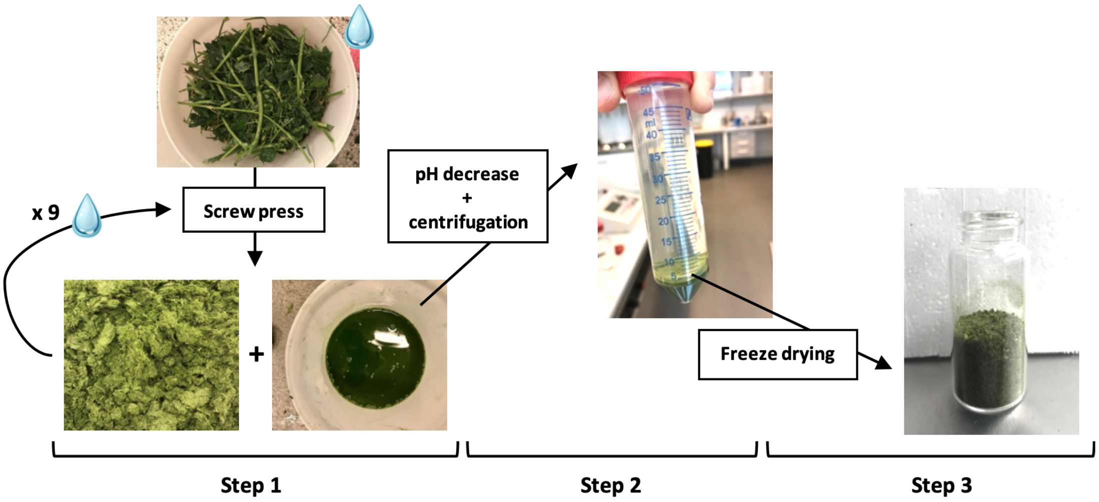


Figure 2 Flow chart of alfalfa protein powder production. Step 1) Frozen alfalfa is thawed by wetting it. Thawed alfalfa is put into a screw press, producing pulp and green juice. The pulp is re-extracted and re-wetted nine times before it is discarded, containing more fibre after each press. At some point the pulp consists of a fibre to protein ratio that is too high to be worth continuing the process of rewetting and pressing the pulp. Each of the nine presses of the pulp results in new pulp and green juice. The green juice consists of the total water soluble solid content. Step 2) pH of the green juice is decreased, and the resulting green juice is centrifuged to precipitate the wanted proteins. This step produces a pellet and a supernatant, further referred to as brown juice being discarded. Soluble compounds, such as carbohydrates are expected to be of high concentration in the brown juice, while the precipitated proteins and insoluble compounds such as fibres are expected to be of high concentration in the pellet. Step 3) The protein pellet was freeze dried into the resulting protein powder in order to concentrate the nutrient compounds. Photos: Christina A Andersen, 2020.

are nowadays determined using NIR spectroscopy. To be able to use NIR spectroscopy as routine method, a sample specific calibration model has to be built. Samples are scanned within the near infrared light region. These spectra are preprocessed and correlated to reference fibre contents obtained by a traditional chemical analysis method. This correlation shows if the spectra might contain relevant and usable information regarding the fibre contents. If that is the case, a calibration model can be developed, which should afterwards be validated. A validated model could be able to predict fibre contents from a simple NIR spectrum of new alfalfa samples, see Figure 3.

This type of model is applicable as a protein powder production process control. The model allows for determining the fibre content by scanning a small amount of sample during the production process, immediately resulting in a fibre content value. By knowing the amount of fibre at a given time in the process, one will know when the process should be stopped by means of nutritional values, since the amount of fibre is indirectly related to the protein content for each of the 10 presses. Thus a model will be beneficial for optimising the production process in order to obtain as much of the protein within alfalfa as possible while still being cost efficient.

Ideally, a model is developed for the green juice, since it is the earliest outcome of the production process, then for the pellet and lastly for the powder. A determination of the fibre content in the green juice, would allow for a rapid answer to whether one should continue pressing the pulp further to obtain a useable product or not. Finally that would result in a faster, cheaper and more sustainable production process. Compared with traditional fibre analysis of food products, for example an enzymatic analysis method taking four days and generating chemical waste, a NIR model analysis is preferred especially for big scale productions [5].

1.3 The Green Biomass Alfalfa

Alfalfa shown in Figure 4, is part of the legume family and currently one of the most important forage crops worldwide due to its high protein content [9, 10]. Historically alfalfa has been used as forage crop in Denmark since the 18th century. The Danish climate conditions are thus very favourable for cultivation of alfalfa. Alfalfa is a sustainable

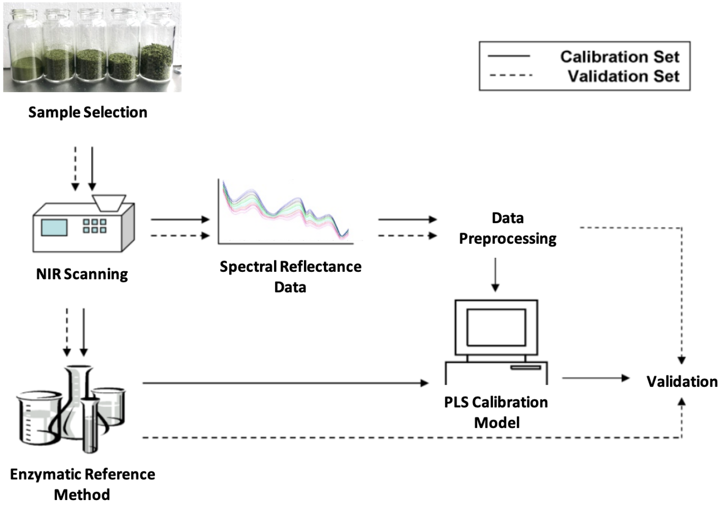


Figure 3 The basic steps of building a NIR calibration model. A selected set of calibration samples is scanned with a NIR instrument. The spectral data is preprocessed, and related to reference values obtained by, in this project, an enzymatic reference method with a PLS calibration model. When a calibration model is built, an independent validation set should be used to validate the calibration model [8]. This figure is modified and adapted from Agelet and Hurburgh [8].

protein source. The growth areas in Denmark are big, it is densely grown without the use of pesticides, and can be harvested up to four times a year.

Alfalfa is grown in large parts of the world. 70% of the worldwide production area is accounted for in USA, eastern Europe, and Argentina, while 20% is in France, Spain, Italy, Canada, China and Australia [9]. Alfalfa easily adapts to different environments and shows a high draught tolerance due to its deep root system improving the efficiency of water usage [10].

Figure 4 Flowering alfalfa grown on a field in Denmark [1].

Alfalfa has a capacity to fix nitrogen through symbiosis with the soil bacteria rhizobia, ensuring a high protein level, mainly within the leaves. Alfalfa changes quickly in quality during growth. It is found that from the second week of growth, the protein content of alfalfa starts to decrease while the fibre content increases, which decreases the total plant digestibility [10]. If alfalfa is cut early, the nutritional value is high, but the chance of regrowth and the amount of harvested biomass is low. If cut late, more biomass can be harvested, but of a lower nutritional value [10]. As a results accurate and rapid nutrient quality evaluations are crucial [11].

The sudden change of nutritional value is correlated with alfalfa flowering. Photosynthetic resources are shifted away from the leaves to production of new plant structures. With the flowering transition follows an increase in tissue lignification as well as in the ratio of stem to leaf tissue. The highest combined amount of biomass yield and nutritional quality of alfalfa is considered at a 10% blooming stage, which results in alfalfa being cut approximately four times a year [10].

1.3.1 Molecular Compounds of Alfalfa

Dry matter derived from alfalfa as a whole plant consists of 30% w/w cellulose, 20% w/w protein, 14% w/w sugars and starch, 11% w/w pectin, 10% w/w lignin, 9% w/w ash/minerals, 3% w/w hemicellulose and 2% w/w oils/lipids [12, 13, 14]. Available carbohydrates account for sugars and nonresistant starch.

Fibres within Alfalfa Dietary fibres are defined as the edible plant cell wall components being resistant to digestion and hydrolysation by endogenous enzymes during absorption in the human small intestine [15]. One way of grouping dietary fibres is by their chemical, physical, and functional properties. Thus they are divided into water soluble and water insoluble dietary fibres, which is applied in this project. Soluble dietary fibres (SDF) bypass digestion in the small intestine and are fermented in the large intestine. Within alfalfa, they consist mainly of pectins [16]. Insoluble dietary fibres (IDF) within alfalfa consist of mainly cellulose, hemicellulose and lignin [15]. Lignin is a major anti-nutritional compound of grasses and the prime factor for limiting cell wall material digestibility. The digestibility is limited by cross-linking of cellulose and hemicellulose, and in turn lignin acts as a physical barrier to microbial attack and digestibility of these polysaccharides [6]. A study conducted by Holloway et al determining the digestion of dietary fibre fractions in humans, shows that approximately 80% of cellulose and 96% of hemicellulose is digested in the human small intestine, while lignin is undigestible in both the small and large intestine [17]. Lignin thus accounts for the largest indigestible part of alfalfa.

The molecular structure of cellulose is presented in Figure 5.

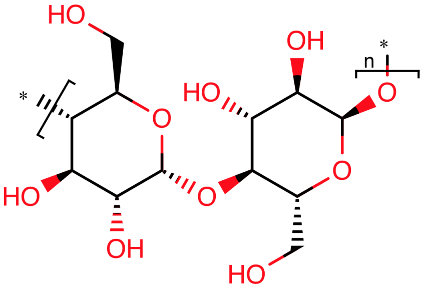


Figure 5 Molecular structure of cellulose [18]. n is a number between several hundreds and many thousand, indicating the total size of cellulose. * indicates attachment places.

When producing protein powder in this project, the aim is set for the highest protein amounts possible, and thus a high amount of dietary fibre is seen as being negative. But dietary fibres still account for a large part of the protein powder, and are not only considered negative related to human health benefits. Dietary fibres do not bind to vitamins and minerals, which leads to a higher absorption of those. Dietary fibre also accounts for slowly digestible energy [16].

Protein within Alfalfa Compared with the protein quality of isolated protein powder derived from crops like soy, whey and pea with similar amino acids profiles, see Table 1, alfalfa protein powder stands out due to its wide range of high content essential amino acids [14, 19, 20]. The amino acid requirements from the World Health Organisation (WHO) are all met only by the alfalfa protein powder presented in this table.

Table 1 Contents of essential amino acids of various protein sources. Amino acid requirements for adults per day by the World Health Organisation (WHO) are stated as comparable values. All numbers are presented as (g/100 g protein).

Essential amino acid	Alfalfa [1]	Soy [2]	Whey [2]	Pea [2]	WHO
Threonine	5.6	3.03	6.92	3.17	2.3
Methionine	2.1	0.39	2.31	0.38	1.6
Phenylalanine	5.6	4.21	3.21	4.68	3.8
Histidine	2.4	1.97	1.80	2.03	1.5
Lysine	6.2	4.47	9.10	5.95	4.5
Valine	6.6	2.89	4.49	3.42	3.9
Isoleucine	5.2	2.50	4.87	2.91	3.0
Leucine	9.0	6.58	11.03	7.22	5.9
Cysteine	1.3	0.26	1.03	0.25	0.6

[1] Values are derived from alfalfa leaf protein powder. [2] Calculated from the mean of a presented protein content range in protein powder by Gorissen et al, 2018. The used mean values were 76%, 78% and 79% for soy, whey and pea respectively [19].

1.4 Analyses Applied in this Project

The used set of protein powder samples is analysed by both a traditional chemical enzyme analysis method being the reference method, and by the NIR instrument producing spectral data. When selecting the set of samples, a sufficient number of samples must be chosen to cover all types of variations within the sample. The used samples must represent the amount of total sample [21].

Three different enzymatic methods are commonly used for determining the amount of TDF [7]:

1. The Prosky/Lee method (AOAC 985.29/991.43), which is used in this project and further described below (AOAC 991.43). Shortly, this method was introduced in 1985. Bacterial α-amylase combined with harsh conditions forms the enzymatic incubation step. This method underestimates the amount of resistant starches, when determining fibre contents.
2. The McCleary method (AOAC 2009.01/2011.25) was introduced in 2009. For the enzymatic incubation step pancreatic α-amylase is used combined with conditions being close to physiological (pH 6, 37°C). All compounds of dietary fibre are measured.
3. The rapid integrated total dietary fiber method (AOAC 2017.16) was introduced in 2015, and only differs from the McCleary method by a slightly more precise determination of dietary fibre compounds.

1.4.1 Enzymatic Analysis used as Reference Method

The enzymatic gravimetric method, Available Carbohydrates/Dietary Fiber Assay Kit provided by Megazyme is used for chemically determining the nutritional contents SDF, IDF, TDF, protein, ACH and ash, see Figure 6 [22]. In an enzymatic gravimetric method, the amount of respective analyte is determined through the measurement of mass, which is based on the Prosky/Lee method.

Available and unavailable carbohydrates are being analysed, the available ones accounting for carbohydrates digested and absorbed by the human small intestine, and the unavailable ones generally referred to as dietary fibre [22]. Legumes such as alfalfa also contain physically inaccessible starch being resistant to hydrolysis by the enzymes of the small intestine, also known by the term resistant starch (RS).

Figure 6 contains of three steps, step 1 accounting for the enzymatic incubation, step 2 describes the determination of dietary fibres, while step 3 determines available carbohydrates.

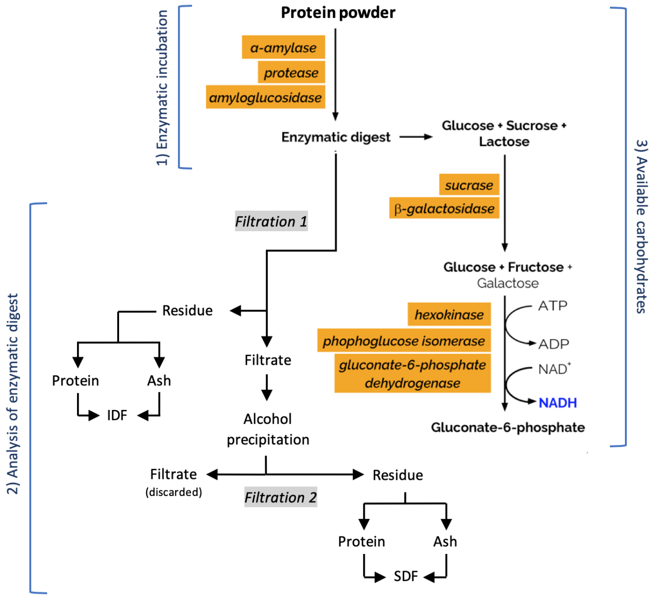


Figure 6 Enzyme assay procedure for determination of the protein power composition. In step 1, the protein powder is exposed to enzymatic incubation with use of the orange marked enzymes. In step 2, the enzymatic digest is analysed, leading to the final determined IDF (Insoluble Dietary Fibre) and SDF (Soluble Dietary Fibre) amounts. In step 3, available carbohydrates (ACH) are determined from the enzymatic digest with the use of solutions containing the active substances marked with orange. This figure is modified and adapted from Megazyme [23].

In step 1, dublicates of a food sample, in this case the protein powder, are treated with enzymes in enzymatic incubation processes mimicking the digestion process in the human small intestine. Enzymes used are α-amylase, protease and amyloglucosidase. Thermostable α-amylase depolymerises nonresistant starch and facilitates their hydrolysis into dextrins. Protease hydrolyses proteins into peptides. Amyloglucosidase facilitates starch dextrin hydrolysis into simple sugars. Resistant starch is hydrolysed during the α-amylase incubation at 95°C, which leads to an underestimation of resistant starch [7].

From the resulting enzymatic digest, a small sample is removed for further analysis of available carbohydrates, accounting for step 3 described below [22]. In step 2, the remaining part of the enzymatic digest is then washed. The resulting residue is dried, whereafter one of the duplicate residues is analysed for protein using the DUMAS method, whereas the other is incubated for at least five hours at 525°C to determine the ash content. The determined amounts of protein and ash are subtracted from the total residue weight, resulting in the weight of IDF [22]. The filtrate is treated with ethanol in a filtering and washing process to precipitate soluble fibre and remove depolymerised protein and D-glucose. The resulting filtrate is discarded, and the final residue is dried. The residue containing soluble dietary fibres as well as protein and inorganic material being ash/minerals. Protein and ash is determined for one of the duplicate each as for the IDF determination. The determined amounts of protein and ash are again subtracted from the total residue weight, resulting in the weight of SDF [7, 22].

In step 3, the sample removed for available carbohydrate analysis, which might contain glucose and sucrose, is diluted with a sodium maleate buffer, and filtered to avoid turbidity. The sample is then incubated with sucrase and β-galactosidase, with the aim to hydrolyse sucrose into D-fructose. This resulting mixture is analysed for D-glucose and D-fructose using hexokinase, phosphoglucose isomerase and glucose-6-phosphate dehydrogenase combined with added water, ATP and NADP$^+$ to facilitate the reactions. Absorbances are measured following the reaction stages, using a spectrophotometer set at 340 nm. Absorbance differences result in calculated concentration values for D-glucose and D-fructose. The amount of ACH is the sum of D-glucose and D-fructose [22].

1.4.2 Near Infrared Spectroscopy

NIR spectroscopy is a fast, accurate and robust analysis method for determining forage quality amongst many other applications that requires no or minimum pretreatment of the analysed sample. It has also shown to previously be a suitable method for determining the TDF content in dried food samples [5].

A NIR spectrum is a fingerprint of a given food sample. It represents the unique physical and chemical composition of it, because the measured spectrum reflects the amount of certain molecular bonds in various types of molecules such as cellulose [24]. A sample is exposed to light at different wavelengths, which leads to a combination of absorption, reflection and transmission of light energy, the proportions depending on the light wavenumber and sample properties such as chemical bonds, composition and thickness [24]. The light path expressed by either one of absorption, reflection or transmission as a function of the wavelength is then collected as a spectrum. NIR is part of the infrared region located in the middle of the electromagnetic spectrum, see Figure 7. The NIR wavelength region from 800-2500 nm, corresponding to a wavenumber range of 12,500-4000 cm^{-1}, and has shown to be very useful for food research [8, 25]. Throughout this project, wavenumber is further used as unit for representing the wavelength. Wavenumber is defined as the number of wavelengths per distance, most often represented by the unit cm^{-1}, and is thus a synonymous for wavelength, but presented in another unit.

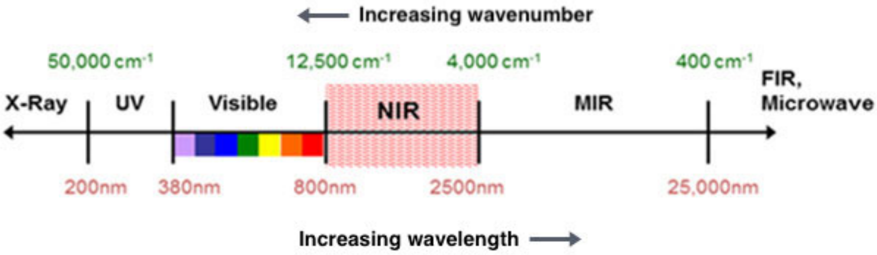


Figure 7 NIR region in the electromagnetic spectrum. The NIR region is marked with red. In addition the X-ray, UV (ultra violet), visible, MIR (mid infrared) and FIR (far infrared) and microwave regions are shown. This figure is modified and adapted from FOSS [26].

NIR Instruments Different NIR instruments exist on the market today. A short overview of four of the most used NIR instruments, mainly differing in how they generate spectra, is presented [27]:

1 Fourier Transform NIR (FT-NIR, also known by FT-IR) instruments is the kind being used in this project, and is described in more detail below. In short, FT-NIR instruments provide a higher resolution, a better wavenumber accuracy and a higher signal energy than many comparable NIR instruments, for example the dispersive NIR instruments [28]. In FT-NIR a light beam consisting of all wavenumbers of the NIR region approaches the sample at once [8].

2 Dispersive NIR instruments have been used for a longer time compared to FT-NIR. Each wavenumber is measured one at a time, directed individually to the detector, resulting in a constructed spectrum by a computer being used as signal processor.

3 Diode array spectrophotometers illuminate a sample with only white light. The reflected part of the light is separated by wavenumber and converted into a spectrum. Each wavenumber is measured by a separate diode detector, making it possible to measure all wavenumbers simultaneously.

4 MEMS (micro electro mechanical systems) introduced portable handheld NIR instruments onto the market, making it possible and easy to obtain sample spectra at many points within a process. They collect spectra fast, but with a smaller wavenumber coverage and lower resolution compared to lab-based NIR instruments.

Due to the properties of the measured protein powder, the FT-NIR instrument in this project is used in reflectance mode, see Figure 8. Basically the FT-NIR instrument consists of six parts [8]:

1 A sample compartment.
2 A lamp as light source.
3 A light wave selection system, known as the interferometer that is able to record signal values at specific wavenumbers. In this system the initial light beam is split in two, one beam is reflected in a fixed mirror while the other is reflected in a moving mirror. The beams are again recombined as an interference pattern in the beam splitter before approaching the sample. This kind of interference pattern is called an interferogram. The moving mirror produces different light frequencies between the two reflected beams, leading to a resulting light beam consisting of all wavenumbers of the NIR region approaching the sample at once.
4 A detector, which transforms the collected light energy to an electric analog signal that is further transformed to a digital signal. Detectors are the most common source of non-systematic instrument noise. This random noise can be manually reduced by taking averages of several spectra from the same sample, thereby improving the signal to noise ratio.
5 The detected digital signal being in a time domain, is turned into an actual spectrum in a frequency domain due to Fourier transform processing. The result is a spectra with a high accuracy [8].
6 A computer being used as signal processor.

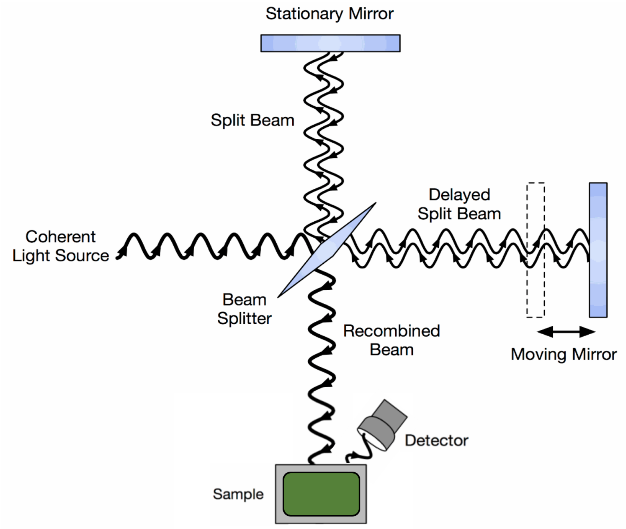


Figure 8 The principles of FT-NIR reflectance analysis. NIR light is originating from the light source, split in two, one beam being reflected in a fixed mirror while the other is reflected in a moving mirror. The beams are again recombined as an interference pattern in the beam splitter before approaching the sample, and lastly being reflected into the detector. This figure is modified and adapted from Harris et al [25].

Before conducting NIR spectra, a relevant background measurement should be obtained, and subtracted from each sample spectra. The purpose of subtracting the same background measurement from all measures samples is to correct for detector sensitivity, and the light source intensity not being equal at all wavenumbers. Without a background correction, the measured samples would be a combination of detector sensitivity, varying light source intensities and sample absorbance, which would not be usable. A background measurement should consist of a uniform sample that is not optically active, for example the powder teflon (PTFE) preferred for NIR measurements in reflectance mode and used for this project, since it effectively reflects light. The background measurement will thus subtract any unwanted residual peaks from the sample spectra.

Reflection Measurements NIR has been widely used as reflectance spectroscopy of powders, which requires an understanding of the two possible ways of reflection, diffuse and specular reflection. Specular reflection is a mirror-like reflection with no energy loss, and the incoming light beam angle corresponding to the outgoing angle of reflection. Diffuse reflection can be considered as a beam of photons where every single one of the photons will have a different fate, see Figure 9. When striking the first particle layer, a photon is either reflected at the surface or passing into the particle. The photon is then travelling through the particle and might be absorbed by a molecule that contains an appropriate bond with the appropriate energy state to accept the energy from the photon. If the photon is absorbed, it does not exist anymore. If it is not absorbed, it will continue passing through the particle until it reaches another boundary, where it will either be reflected or transmitted. If it is reflected, it continues to move until it is transmitted, absorbed or escaping the sample surface opposite to the original direction at any angle from 0°to 90°. Diffuse reflection therefore is defined as reflection of part of the incident energy at a range of angles in all 0°to 90°directions from the incident beam [21]. This kind of reflection leads to information about the given sample surface, since samples are only measured until a certain depth. By collecting the amount of light that is diffusely reflected from solid samples compound concentrations can be predicted.

When performing reflection spectroscopy, a given amount of light energy is directed onto the sample, whereafter the amount of reflected energy is measured. The result can consist of energy that is reflected, absorbed or absorbed and then reemitted or transmitted, which makes it complex and hard to quantify [24, 21].

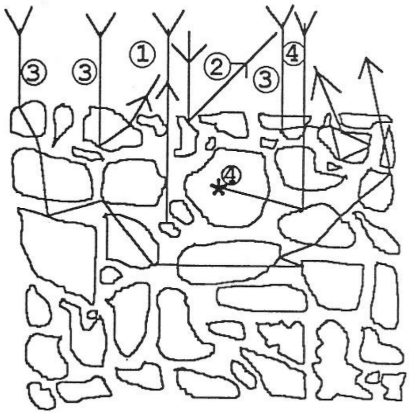


Figure 9 Pathways of diffuse reflection within particles in a sample. Each line indicates the path of a single photon. 1) Specular reflection, not detected. 2) Specular reflection, detected. 3) Diffuse reflections. 4) Absorption. Specular reflection will always be present in measurements of diffuse reflection [21].

Interpretation of NIR Spectra Within the NIR spectra in this project, the wavenumbers measure the energy of the radiation, while the reflected light measures the relative amount of energy absorbed by the sample [21]. NIR spectra are formed from molecular bond vibrations. Each individual bond within a molecule can be seen as a weak spring that naturally vibrates in a given way. Detection of intermolecular vibrations makes it possible to look directly at a sample, since absorption of the light energy is caused by bond vibration. Vibrations will arise in several directions, requiring different energy amounts. Only molecules with an electric dipole moment, and bonds vibrations that do not cancel out each other, will absorb infrared light and vibrate, see Figure 10. [21]. Such bonds are further referred to as R-H bonds.

Furthermore, molecules follow the rules of quantum mechanics, which means energy changes occur between discrete energy states. This will in turn lead to absorptions at different wavenumbers, which can be related to the type of molecular bond and in turn the type of molecule [21].

Vibrations of molecular bonds leading to light energy absorption can be explained by the harmonic and anharmonic model for potential energy, see Figure 11. Molecules do not behave exactly according to the spring model as harmonic

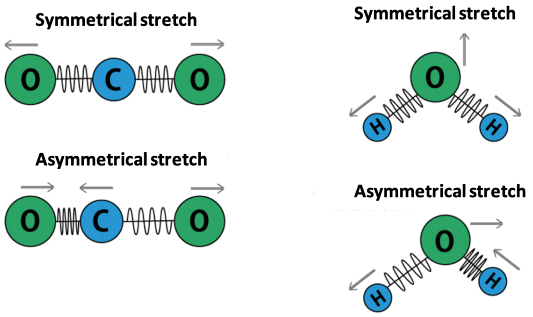


Figure 10 Typical atom vibration of a triatomic molecule. Only molecules with an electric dipole moment, and bonds vibrations that do not cancel out each other, will absorb infrared light and vibrate. These requirements are met by H_2O, but not by CO_2 [21]. This figure is modified and adapted from Jasco [29].

oscillators, but the harmonic model can be used to explain the anharmonic model [21]. Chemical bonds between atoms hold a potential energy depending on the bond length. Diatomic molecules vibrate with a given frequency more or less following the harmonic model. When the frequency of light matches the frequency of a given bond, the bond will absorb energy and move into a higher vibrational energy level (v). Harmonic transitions are defined by vibrational energy level changes of $\pm$ 1. For a given wavenumber range, some light frequencies are absorbed, some are partly absorbed and some are not absorbed, which are the ones not matching any of the energy differences for a given sample of molecules. This intensity of absorption versus wavenumber is the foundation for the NIR spectrum of a sample [30].

The model of the harmonic and anharmonic oscillator describes this transition, with the potential energy (U) as a function of the interatomic distance from equilibrium, which is the minimum energy position of a molecular bond. A diatomic molecule absorbing light energy is able to transition into energy levels that are not contiguous. Therefore, the molecular vibrations are described with the more realistically anharmonic potential instead of the harmonic potential [30].

The dissociation energy accounts for the required energy to break a given diatomic chemical bond. R-H bonds contain a high dipole moment and a large mass difference between the atoms, which causes little energy needed for the bond to break and show anharmonic tendencies. This property causes R-H bonds to be highly anharmonic and dominate NIR light induced transitions. Especially H_2O stands out, which makes it preferable to analyse completely dry products when looking at other molecules than water. Thus NIR is well suited for measuring substances containing R-H bonds in food in relatively high amounts like fibres [24].

Basically molecular vibrations arise due to two mechanisms known by overtones and combination bands, which require anharmonic behaviour [21]. A NIR spectrum consists of these overtones and combination bands of R-H bonds, see Figure 12.

Overtones are by definition electron excitations to higher energy levels, a transition from the ground state (v=0) to the second energy level (v=2) or a higher level, see Figure 11. Up to four overtones can occur, the fourth overtone being very weak and ignored. Every NIR overtone is repeating information, but with decreasing absorption bands when the overtone level increases, which produces less intense peaks in a NIR spectrum the higher the wavenumber gets. In other words, the chemical information in a NIR spectrum is repeated and overlapped through all of its wavenumber range [8].

Combination bands occur when one molecule shares the energy of a photon between two or more absorptions simultaneously, in other words if two overtones are excited at the same time. One peak in a spectrum thus represents two molecular bonds instead of one [8].

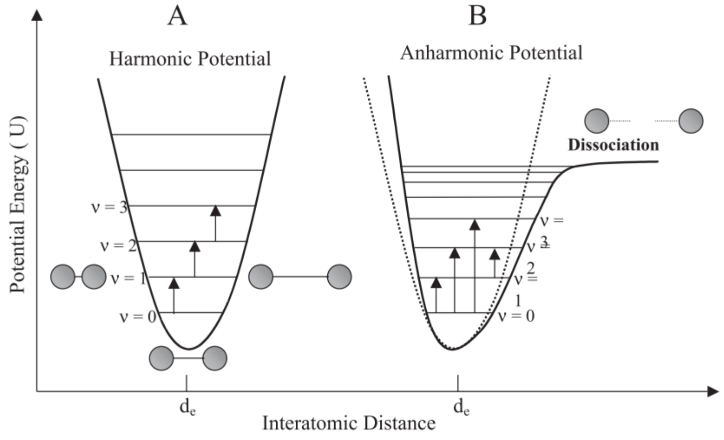


Figure 11 The harmonic and anharmonic model for potential energy of a diatomic molecule. The potential energy (U) is shown as a function of the interatomic distance from equilibrium (d_e), which is the minimum energy position of a molecular bond. When the frequency of light matches the frequency of a given bond, the bond will absorb energy and move into a higher vibrational energy level (v). The dissociation energy accounts for the required energy to break a given diatomic chemical bond. [30].

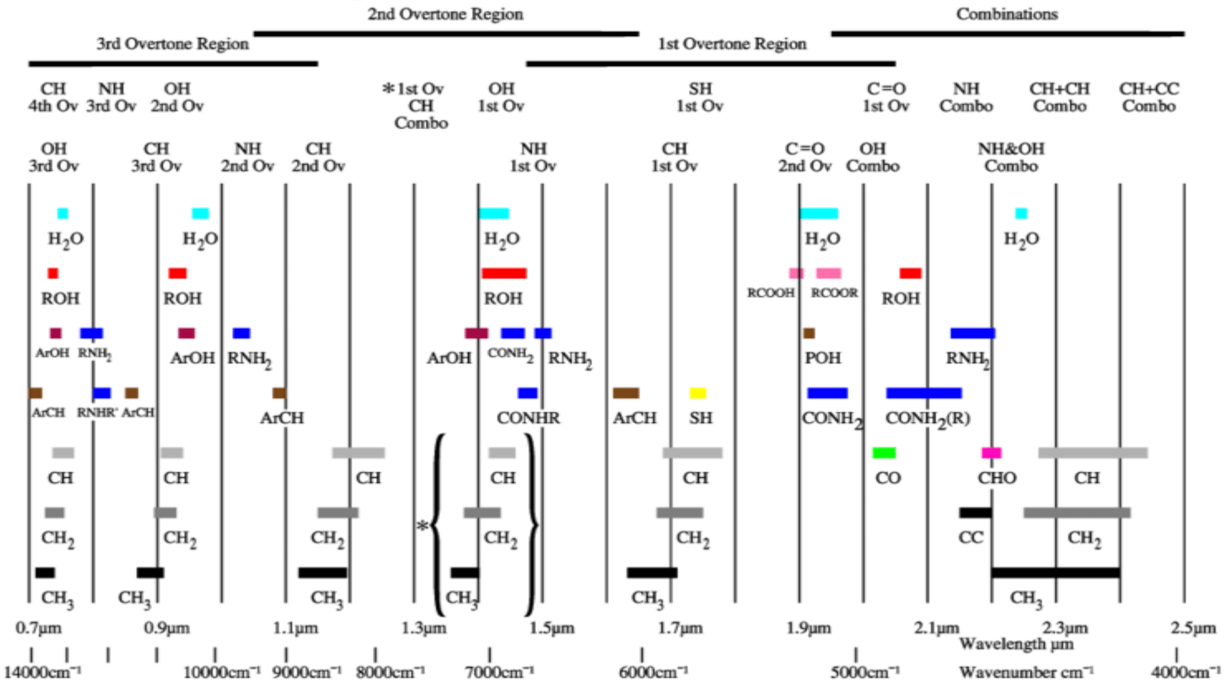


Figure 12 Absorption bands in the NIR area, presenting overtone regions, the combination region and possible detected compounds [25].

A complex molecular structure of an analysed sample will lead to complex broad and highly overlapping peaks and valleys within the NIR spectrum [31]. This makes it crucial to look only at the most relevant peaks within a wavenumber range for the given analysed sample.

NIR samples are known for no need of sample preparation. Within this project the analysed protein powder is prepared from raw alfalfa, but that is accounting for the final product from alfalfa and should have been prepared independently of NIR spectral measurements. Thus, often there is no need for additional sample preparation prior to the NIR measurements. One exception though, is that particle size differences could lead to differences in spectral data, which potentially makes sample preparation crucial for NIR measurements as well. If this is an issue, grinding and sieving of powders is widely used as sample preparations [21].

1.4.3 Multivariable Data Analysis

A NIR spectra consists of many wavenumbers resulting in hundreds of variables for each analysed sample. Therefore, multivariable data analysis is well suited for this kind of data, since it is made for handling big data sets. The goal for use of multivariable data analysis in this project is to build a calibration model from a spectrum with a given wavenumber range x and chemically determined fibre contents y, based on the simple relationship in Equation 1 [24]:

$$y = f(x) \tag{1}$$

In order to build a model, it has to be examined how the difference in spectral x-values affect the difference in the y-values. A large part of the variation in the spectral x-values might not be related to the variation in the y-values. Therefore raw spectra most often have to be preprocessed in relevant ways. A transformation of the x-values then results in a new set of x-values, whose variation might correlate better to the variation in y [24]. Which preprocessing methods are the correct ones to apply, depends on the spectral data and the y-values. Therefore different methods of preprocessing have to be tested by trial and error, with the aim to be left with preprocessed spectra, which are able to produce a perfectly fitted model with the determined fibre reference values, y [21, 24]. Relevant preprocessing methods are described in the following, where the spectral data is presented by matrix X, each row corresponding to an individual measured spectrum.

The spectral data is mean centered as part of the preprocessing methods. Mean centering accounts for a calculated mean spectrum of all analysed spectra being subtracted from each spectrum. Since the interest lies in the difference between the samples, and not in the actual values, mean centering is used. A normalisation with for example standard deviation of the spectral data is not required, since the spectral data is not compared with datasets of different origin.

Initial Removal of Spectral Noise Raw NIR data contains a large amount of noise, overlapping of data and interferences. To remove irrelevant interferences, the initial preprocessing consists of subjectively discarding wavenumber ranges clearly containing noise. In some measured wavenumber ranges, no transmitted light is detected resulting in very fluctuating spectra with no information in the given area and thus no useful result. Only noise from the instrument is detected. In other noisy wavenumber ranges all light is transmitted, which also leads to no useful information. Both parts of the spectra are discarded [24].

Principal Component Analysis Principal component analysis (PCA) reduces the original big spectral variables set, matrix X, to fewer variables by constructing new variables grouped into principal components (PCs). These variables within the PCs still explain the entire original data variation. The maximum number of computed PCs is determined by the number of samples-1 or spectral variables-1, which ever is the lowest. PC1 explains most of the data variation in the observed matrix X while the last PC explains the least [32].

Each PC consists of independent loadings and scores matrixes, which are two matrices the original matrix X has been decomposed into. The correlation is shown in Equation 2, where matrix P contains loadings and matrix T contains scores:

$$X \times P \approx T \tag{2}$$

The row vectors of T correspond to the row vectors of X, which are the wavenumber values, but are projected down onto a space defined by the column vectors of P. Each vector in matrix T corresponds to one analysed sample. The more equal the T vectors are, the more identical are the analysed samples. A difference between the samples can thus be determined by comparing the T vectors graphically.

Likewise, a projection of the column vectors in X onto a space defined by the column vectors of T, will result in the loadings matrix P, again narrowing down the number of data points. The more equal the P vectors, the more identical are the measurements for the given dataset. The loading matrix describes in that sense the relation between the wavenumbers for the same sample.

Thus one way of using a PCA plot is by determining how well the score values of PC1 correlate with the given reference values. Such a PCA plot is able to visualise the correlation among all analysed samples, and thus can be used to show if the spectra will allow for the correct triplicate samples to cluster in groups due to their theoretically similar score values. A given number of PCs showing are showing a close triplicate clustering. The remaining PCs are most likely describing some kind of spectral noise. A more quantitative way of determining whether the data seems to fit a one component model or is described by additional components, is by looking at the percentage of total variance explained by each principal component [21].

A PCA plot will always consist of the same amount of data points as analysed samples no matter how many initial wavenumber variables were chosen, and can be used after each kind of spectral preprocessing to visualise if the preprocessing method indicates a structured separation of the samples [21].

Standard Normal Variate Standard normal variate (SNV) transformation can be used when one parameter being for example a given nutrient concentration, shall be predicted from more than one similar sample measurement with known identical compound concentrations, for example if triplicate measurements are used. If the light path length through the similar samples differs, but should be identical, which is indicated by linear offset identical looking raw spectra, SNV transformation can be applied, see Equation 3. Both additive and multiplicative effects in the spectra can be removed. Differences in light path lengths could arise from differences in particle size and particle distribution, and SNV transformation thus also effectively corrects for scatter corrections [33, 24].

SNV corrects the given spectrum with only itself, see Equation 4, which is then being scaled by the corresponding standard deviation, see Equation 5:

$$g(x) = kf(x) + b,\ k > 0 \tag{3}$$

$$g(x) - < g > = a(f(x) - < f >) \tag{4}$$

$$(g(x) - < g >)/s(g) = (f(x) - < f >)/s(f) \tag{5}$$

where x is the wavenumber, and g and f are vectors of spectral data originating from the same sample, k, a and b are constants. Assuming that the light path length is not identical, k is representing this constant. $<>$ is the average over the spectral values, and $s()$ is the standard deviation. SNV transformation of f and g thus will results in the same value and a "perfect" correction is made.

Multiple Scatter Correction Multiple scatter correction (MSC) is similar to SNV transformation, but instead of correcting a spectrum with only itself, MSC produces one reference spectrum, which is the mean spectrum of all measured spectra, that is used to correct every spectrum. When using the average of all measured spectra, it is assumed that this spectrum has the most representative general shape of the type of samples in question, which might not be the case, if for example a dilution series is measured in triplicates, as for this project [24].

First and Second Derivatives Derivatives, also known as Savitzky–Golay filters are commonly used to minimise interference when not only a constant, but also a linear baseline separates the spectra from each other, see Equation 6. Derivation with respect to the wavenumber x increases random noise, and the spectrum should thus be smoothed before calculating derivatives. The first derivative, also referred to as Der1, measures the rate of change of the signal and can be used if only a change of the offset point on the y-axis is of interest, Equation 7. This is seldom the case though, and the second derivative is therefore more used [21, 24].

The second derivative, also referred to as Der2, measures the rate of change of the first derivative and can be used to eliminate a systematical increase of a linear baseline in the data, since a two fold derivation of a straight line becomes zero, see Equation 8. Afterwards, SNV transformation could be applied if the difference in the constant k should be eliminated [21, 24].

$$g(x) = kf(x) + ax + b \tag{6}$$
$$g'(x) = kf'(x) + a \tag{7}$$
$$g''(x) = kf''(x) \tag{8}$$

where x is the wavenumber, and g and f are vectors of spectral data, k, a and b are constants.

When applying smoothing followed by either Der1 or Der2 preprocessing, it is required to choose the degree of fitted polynomium and a window size, the latter being the number of wavenumber data points used for estimating the smoothed data, see Figure 13.

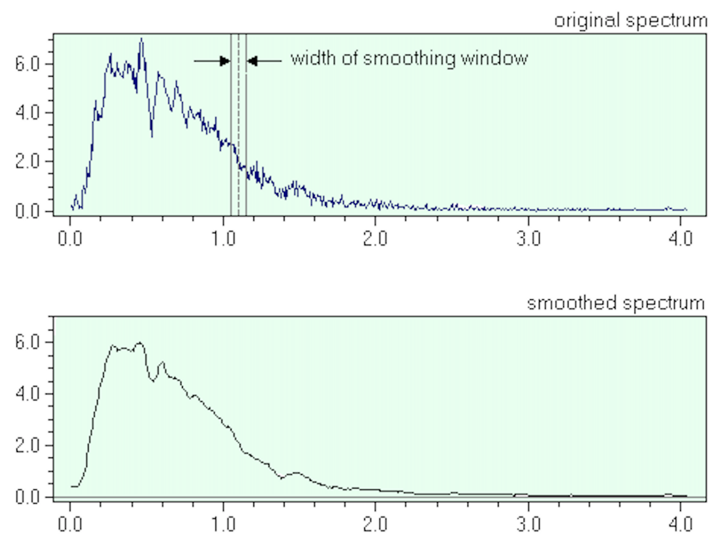


Figure 13 Choice of window size affecting spectral smoothing. The width of the smoothing window corresponds to the number of wavenumber data points used for estimating the smoothed data [34].

Development of a Calibration Model When the spectral data is sufficiently preprocessed, and an acceptable PCA plot correlation between the spectra and measured reference values is seen, a calibration model is built. A calibration model uses the relation between spectra and reference values to predict new reference values, which theoretically should be identical to the measured reference values. A well suited method for development of a calibration model with NIR data is the linear regression method partial least squares (PLS), which combines the techniques of PCA and multiple regression [5]. PLS produces a linear regression model with the least error in the sum of squares between a predicted regression line and observed reference variables projected into a new space, see Figure 14 [35].

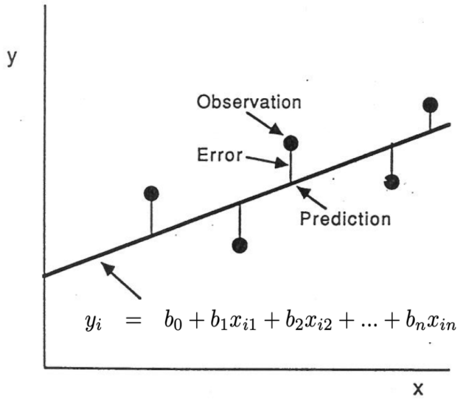


Figure 14 Basics behind PLS regression. PLS produces a linear regression model with the least error in the sum of squares between a predicted regression line and observed reference variables projected into a new space. This figure is modified and adapted from Davies [21].

From the spectral wavenumber values x put into a PLS model, new predicted variables y_{i} can be calculated according to Equation 9 [24]:

$$y_i \quad = \quad b_0 + b_1x_{i1} + b_2x_{i2} + ... + b_nx_{in} \tag{9}$$

where $x_{i1}, x_{i2}, ...x_{in}$ is the vector of spectral values for sample i, the index number corresponding to each of the measured wavenumbers. $b_0, b_1, ...b_n$ is the vector of regression coefficients common to all samples determined from the PLS regression and the observed reference values, and also calculated for each wavenumber. b_0 should be close to zero, in order to get the best possible regression line. In this way a new predicted value for each sample is calculated. The predicted variables will be presented in a vector containing as many predicted values as initial measured reference values. The initial measured reference values are put into the model as observed variables [35].

A plot of b-coefficients against the wavenumbers will indicate which wavenumbers affect the PLS model most, that is the b-coefficients being farthest away from zero, according to Equation 9.

From the PLS model, the correlation coefficient, R^2 is used to indicate how good the model correlates to the straight line y = x, since for a perfect fit within this project, that should be the regression line [21].

Using the Calibration Model for Prediction When a linear PLS model is built from known reference values, the proposed model should be validated and tested with a complete new and independent set of samples according to Figure 3 [21]. Any new measured spectra analysed with a PLS model, is preprocessed the same way as the original calibration spectra, by subtracting the original preprocessed mean spectra from the new spectra inserted into the model. In this way new spectra should be able to predicting new usable results with the PLS model.

1.5 Statistics

The standard error and precision of the reference method should be taken into consideration, since results predicted with a NIR model never will be better than the reference method. A common way to reduce the standard error, is to analyse more samples [21].

Outliers within the chemically determined contents are detected by the standard deviation method. Outliers are identified if deviating more than two standard deviations from the mean.

Outliers within the model data are detected by the median absolute deviations method (MAD). If a value is more than three scaled median absolute deviations (MAD) away from the median of the data, an outlier is detected [36].

Theoretically, no outliers should exist, since the triplicate spectra are all measured on the same kind of sample, but due to human errors amongst others they could exist. To increase the robustness of a model, big outliers are discarded, but smaller differences between samples will results in a more robust model. A possible source of error is that an outlier is detected, which actually corresponds to a true variation in data [24].

2 Materials and Methods

All chemical analyses were performed in laboratories of The National Food Institute at the Technical University of Denmark in Copenhagen.

2.1 Protein Powder Production

According to the supplier of alfalfa for the InnoGrass project, Pauli Kiel from Biotest ApS in Middelfart, Denmark, the used alfalfa was manually cultivated with a scythe by farmer Arne Hviid at his farm on Å Strandvej 33, 5631 Ebberup in Denmark. The used batch of alfalfa was harvested the 20th of October 2019 at a length of 25-30 cm, being one of four yearly harvests. Immediately the harvested alfalfa plants were placed in sealed plastic bags, transported to DTU and directly put into storage in a -22°C freezer still in sealed plastic bags. Thus the alfalfa plants were put into storage in the freezer approximately two hours after harvest. The whole plant, that is stems, leaves and flowers are used for the protein powder production.

Prior to the protein powder production, see Figure 2, frozen alfalfa is collected from storage in the -22°C freezer and submerged in room tempered water in an open plastic container to defrost it. The initial amount of frozen alfalfa and all steps of the production process are documented through weighing, see Table 4 in Results. For weighing in larger scales, that is the raw alfalfa and the pulp in the first protein powder production process, the scale Signum 1 from Sartorius Mechatronics, Germany with a 0.0001 kg resolution, measuring up to 35 kg was used. For weighing in smaller scales, that is the green juice, the brown juice and the protein pellet, the scale LE6202S from Sartorius Mechatronics, Germany with a 0.01 g resolution, measuring up to 6,200 g was used.

Defrosted alfalfa is then pressed 10 times in total at a frequency of 40 Hz into a green juice consisting mainly of protein and fibre amongst other nutrients, using the screw press CP-4 from Vincent Corporation, Florida, U.S.. This accounts for step 1 in Figure 2, presented in the Introduction, and describing the steps of the protein powder production process.

For the first press, thawed and wetted alfalfa is fed into the screw press. For the following presses, the pulp from the prior press is fed into the screw press. Each press results in a fibrous pulp that is being rewetted with room tempered water of double its weight to hydrolyse it and easier obtain the proteins. The rewetted pulp is then fed into the screw press as the "new alfalfa". The last and 10th portion of pulp is not looked at further in this project, since the fibre to protein ratio is too high to be worth continuing the process of rewetting and pressing the pulp. The high amounts of raw alfalfa being pressed in relation to the size if the screw press, did not allow for performing all 10 presses in one single day, thus two production days were needed. During the night in between, the pulp was stored in a -22°C freezer, which potentially could lead to a higher protein content in the final powder, since the plant cells could be damaged due to the cold environment, thus leading to easier obtainable protein.

In step 2, the pH of the green juice from step 1 is decreased to the isoelectric point pH = 3.5 of the proteins within the green juice mainly consisting of the protein RuBisCO, to precipitate the proteins in order to get the highest possible amount of protein in the produced powder [37]. For the pH decrease, 1M HCl from the producer Sigma-Aldrich, Germany is used. With each press, less HCl is required to reach pH 3.5, since the amount of protein with a high buffer capacity decreases. For pH measurements, the used pH meter is a CHECKER® - pH Tester 0-14 pH from Hanna Instruments, Italy with a 0.1 pH resolution and an accuracy of ± 0.2 pH. The green juice is stored in a +1°C refrigerator for maximum 24 hours before being centrifuged.

A centrifugation at 4200 × g and 4°C for 10 minutes results in a protein rich pellet and a supernatant, being a brown juice. The Heraeus Multifuge X3R centrifuge from Thermo Fisher Scientific, Massachusetts, U.S., with a capacity of 4 × 1000 mL is used. The brown juice will not be looked at further and is discarded. The pellet is collected and frozen at -40°C prior to freeze drying it into the final protein powder, step 3 in Figure 2. To achieve sublimation during freeze drying, the freeze drying process is proceeded at a pressure below 0.612 kPa, the triple point of water [38]. A pressure of 0.08 kPa at a temperature from -40°C to 20°C within 72 hours within a Heto PowerDry DW8 Freeze Dryer by Thermo Fisher Scientific, Massachusetts, U.S., was used. See Table 4 in Results for specific amounts of evaporated moisture during freeze drying.

The protein powder is homogenised, by milling it down to a particle size of 1 mm prior to the following analyses. For this process, a DLFU universal laboratory disk mill from Bühler, Switzerland is used. The protein powder milled down to 1 mm is further referred to as the original protein powder. It is stored in a -18°C freezer in sealed plastic bags. When taking out samples for analysis, the amount of needed protein powder is transferred in 50 ml tubes from Sarstedt, Germany.

For the NIR spectra measurements, the original protein powder was examined, but also further divided into particle sizes by sieves of the sizes 1 mm, 0.5 mm, 0.25 mm, 0.125 mm and $<$ 0.125 mm using a MLUA universal laboratory sieve from Bühler, Switzerland. Further throughout the project, these protein powder fractions are referred to by the size of the sieve instead of a particle size interval. Stacked sieves were set to a vibrational speed of 300 rpm for 5 minutes, following procedures according to ISO 8130-1:2019 [39]. The protein powder was thus divided into different particle sizes, since the NIR spectra potentially could be particle size sensitive. Table 9 in Appendix A1 presents the distribution of particle fractions.

Since model building depends on the location of clear spectral peaks, similar peak locations of these measured spectra might result in similar derived models. Whether clear peak locations in a mean spectrum from each different particle size significantly differs from each other is being determined with a one way ANOVA, since the data set is found to be normally distributed. A significance level of $\alpha = 0.05$ was used, with a null hypothesis H_0 stating that the differences between the mean spectral peaks are not statistically significant.

During the protein powder production, all green juice was centrifuged, and the entire amount of pellet was freeze dried, to produce as much protein powder as possible. Later during the project process, it was decided to look into NIR spectra of the green juice and pellet. Therefore, new batches of alfalfa were pressed 10 times, but in smaller amounts. All steps of the process are documented through weighing, see Table 10 in Appendix A1. For weighing all amounts of this smaller production, the scale LE6202S from Sartorius Mechatronics, Germany, with a 0.01 g resolution, measuring up to 6200 g was used.

The same procedures and amounts of added water as for the first batch were followed, but there was no need of freeze drying the pellet, and a smaller screw press, an electric juice extractor of the model Angel Juicer Angelia 8500S was used, which operates with only an on/off and reverse button, if material is stuck. Both green juice and pellet was stored in a -18°C freezer until used for NIR measurements one week later.

A list of all used alfalfa samples is presented in Table 2:

Table 2 Overview of analysed alfalfa samples. The type of sample and method of analysis is presented. Original particle size refers to freeze dried protein powder milled down to 1 mm. The presented particle size of the remaining protein powder samples refer to the size of the sieve used for particle division, instead of presenting the particle sizes as intervals.

Type	Method of analysis
Powder, original	NIR and enzymatic analysis
Powder, 1 mm	NIR
Powder, 0.5 mm	NIR
Powder, 0.25 mm	NIR
Powder, 0.125 mm	NIR
Powder, $<$ 0.125 mm	NIR
Liquid, green juice	NIR
Semi solid, pellet	NIR

2.2 Determination of Protein Powder Contents

The enzymatic gravimetric method, Available Carbohydrates/Dietary Fiber Assay Kit provided by Megazyme, Ireland is used for chemically determination of the nutritional contents, SDF, IDF, protein, ACH and ash [22]. Procedures of this method are strictly followed, as well as materials used are strictly identical to what is stated in the assay, except from deviations stated in this section.

Moisture The moisture content of the protein powder was determined separate from the assay, by determining the dry weight of duplicate samples of 1.0 g. Each sample was exposed to 140°C for 10 minutes using the infrared moisture determination balance AD-4714A general-purpose moisture determination balance by A&D Weighing, Tokyo, Japan.

Lipids The content of lipids in the protein powder was determined prior to this project, ensuring that less than 10% w/w of the protein powder consists of lipids. Furthermore, the greatest part of the lipid content in raw alfalfa is soluble and washed out with the brown juice during production of the protein powder. The powder is thus suitable for use in the assay.

General Deviations and Comments The general deviations and comments are stated following the order of occurrence in the assay. Throughout the assay, the scale AG204 DeltaRange from Mettler Toledo, Ohio, U.S., with a 0.1 mg resolution, measuring up to 81 g was used for weighing in small scales, while the scale LE6202S from Sartorius Mechatronics, Germany with a 0.01 g resolution, measuring up to 6200 g was used for weighing in larger scales. Furthermore, the Finnpipette™ F2 GLP kit from Thermo Fisher Scientific, Massachusetts, U.S., was used for all pipetting. For all magnetic stirring, the magnetic stirrer MIXdrive 15, 40015 with 15 stirring positions was used together with the MIXcontrol 20, 90200, both from 2mag magnetic ^{e}motion, Germany. The filter paper used throughout the enzyme assay are Fisherbrand™ microglass fiber filter discs, Ø 47 mm from Thermo Fisher Scientific, Massachusetts, U.S.. Chemicals from Sigma-Aldrich, Germany are used as reagents not provided as part of the enzyme assay.

Sodium azide as a preservative was not used for stabilising the content of the provided bottle 1 or the sodium maleate buffer, since an extended shelf life for more than three years and more than one year respectively was not required.

Throughout the assay, filter paper is used for filtration steps instead of proposed fritted crucibles, due to availability reasons. This deviation is expected to not affect the final result. For the ash determination though, regular crucibles are used prepared as stated in part A.a.2.a., A.a.2.e., A.a.2.f. and A.a.2.g. in the enzyme assay. The proposed micro cleaning solution in part A.a.2.c. is not used for additional cleaning of crucibles, since it is expected that they are delivered sufficiently clean into the laboratory.

In part A.b.2.b. the solutions were stirred for 10 minutes at 350 rpm. In part A.b.3.a. the solutions were stirred for 2 minutes at 150 rpm.

Prior to performing this enzyme assay, different available water bath opportunities were tested, in order to find the best suited, since no available water bath fitted the exact description of the assay. A mashing bath with 8 beakers of the model LB8 from Lochner, Labor + Technik, Germany was chosen, see Figure 26 in Appendix A1 [40]. The assay states that the bottles in the water bath should be completely sealed, which the LB8 mashing bath does not allow due to the placement of stirrers, see Figure 15. Thus the bottles are weighed before and after exposed to the water bath, at part A.b.3.b. and A.b.8.b. respectively, in order to determine the amount of evaporated water, which is then added in part A.b.8.b. to be able to still use dilution factors and calculations as they are stated in the assay. An average of 3.40 g demineralised water was added to each bottle. Furthermore, shaking could not be applied during the water bath, but continuous agitation was applied at 100 rpm. The temperatures and times were controlled by a program following the enzyme assay and specifically made for these analyses.

In part A.b.6.b. the solutions were stirred at 300 rpm. In part A.b.8.e. it was chosen to store the solutions below -10°C before determination of ACH. For heating added water and ethanol to 60°C, in between part A.b.8. and A.b.9., see page 11 in the enzyme assay procedure, and in part A.b.9. respectively, a water bath was heated by the DT Hetotherm Heating circulator, type 21 DT-2 from Heto Holten Lab Eqiupment.

In part A.b.9. the solutions were stirred for 5 minutes at 400 rpm. In part A.b.10.a. filter paper was weighed instead of crucibles, and part A.b.10.b. and A.b.10.c. were thus ignored. The vacuum pump, Diaphragm vacuum pump, VP

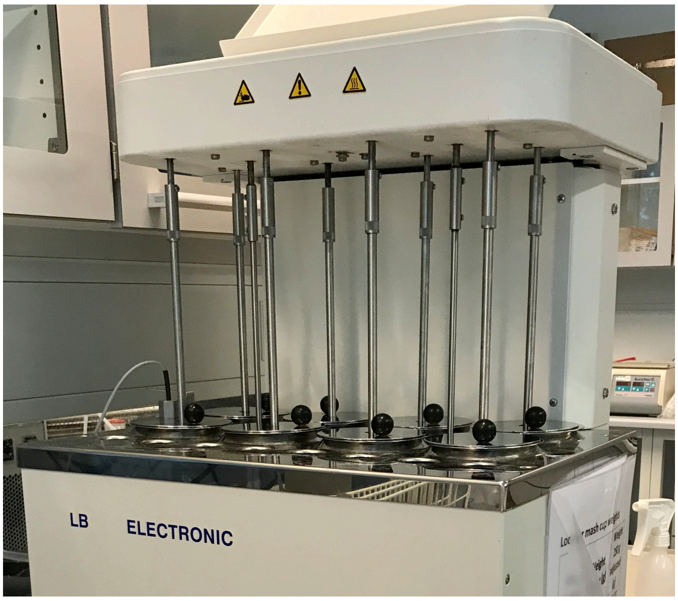


Figure 15 Water bath with 8 beakers of the model LB8 from Lochner, Labor + Technik, Germany. The cups are not completely sealed due to the placement of stirrers. Photo: Christina A Andersen, 2020.

820 from VWR™, U.S., was used for filtering purposes in part A.b.11. and A.b.12.. In part A.b.13. a 105°C air oven with reference number S.680315 from Elektrohelios, Sweden was used to dry the resulting filter papers placed on weighed tin foil. Directly after part A.b.14. the weighed dried filter paper including tin foil was stored in a -18°C freezer until used for further analyses.

Protein The DUMAS method, as proposed by ISO 16634-2:2009, is used for determining the total nitrogen content instead of the Kjeldahl method as stated in the enzyme assay. Compared with the Kjeldahl method, DUMAS is less time consuming, which is preferred when analysing the amounts of samples looked into in this project. The DUMAS method was the easiest protein determination method available. The rapid MAX N exceed from elementar, Germany is used as N/protein analyser according to the DUMAS method. The conversion factor 6.25 is used for protein content calculations [41]. The presented total protein contents are determined separate from the enzyme assay with less risk of human errors.

Since filter paper is used for filtrations in the assay, the DUMAS measurements within the assay will include one filter paper each. Thus a separate blank measurement containing only the filter paper is conducted, in order to determine how the filter paper affects the determined nitrogen content. It was seen that the blank filter paper would not affect the sample measurements, since the collected nitrogen area was below the detectable area. Unfortunately, the DUMAS results of the protein powder samples within the assay divided into soluble and insoluble samples, could not be used, since the samples representing insoluble material showed too high nitrogen area values, while the samples representing soluble material showed too low nitrogen area values, in order to be measurable with this DUMAS method.

Since the protein representing the insoluble and soluble material should be used in order to calculate the amounts of SDF and IDF respectively, it was decided to use the initial separately determined protein contents, and conclude by subjectively looking at the amount of residue on the filter paper containing the soluble material, that the soluble material might not have contained any significant amounts of protein. This decision was made after performing the enzyme assay, and having seen that the soluble material in average weighed 0.02 g, while the insoluble material weighed 0.46 g. Figure 16 presents the ethanol precipitated soluble material before the filtering step, indicating that the amount of soluble material is limited.

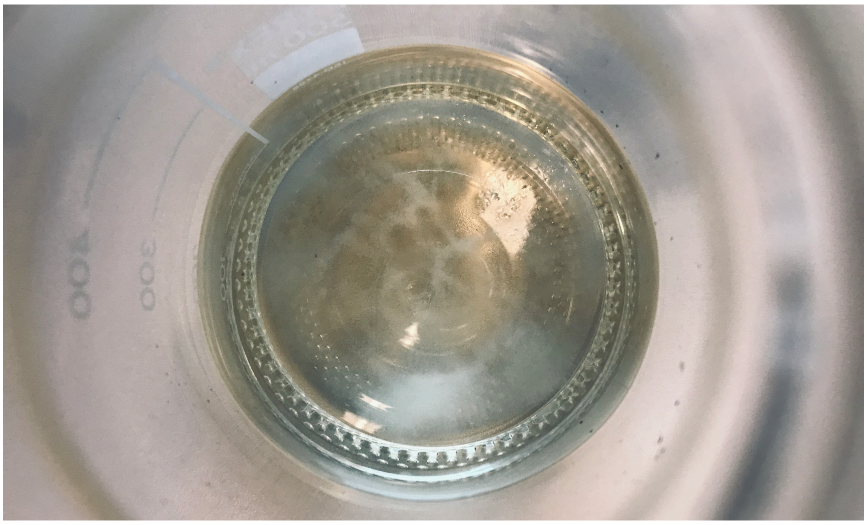


Figure 16 Ethanol precipitated soluble material before filtration within the enzyme assay. Photo: Christina A Andersen, 2020.

Ash Ash is determined in the same way as proposed by the assay, but since filter paper is used for filtrations, the filter paper is dried together with the residues. For ash determination, crucible, kieselguhr (SiO_2) and filter paper weight is subtracted. Kieselguhr (SiO_2) is referred to as celite in the assay and for further references in this project. A separate blank measurement containing only the filter paper is conducted in order to accurately subtract the weight of the filter paper.

Separate from the assay, additional triplicate ash determinations were performed. The moisture content of the protein powder was determined to 0.00%, which allowed for further use of regular crucibles and no need for fritted crucibles. The same procedures as for the ash determination in the assay were used, with the same deviations as mentioned here, except that for these measurements, the protein powder was directly placed on celite since filter paper could be left out.

For all ash determinations, the ash oven D6450, type M110 from Heraeus, Germany was used. Presented ash contents within the protein powder are mean values of detected ash in the assay and separate from the assay.

Available Carbohydrates Available carbohydrates (ACH) are determined according to the procedures in the assay. 50 ml tubes and 15 ml tubes from Sarstedt, Germany are used for the liquid transfer and handling of the samples. For all ACH determinations, the UV-VIS spectrophotometer Genesys 10S from Thermo Fisher Scientific, Massachusetts, U.S., was used.

Dietary Fibre Dietary fibre divided into SDF and IDF are determined according to the procedures in the assay, but affected by the mentioned deviations.

2.3 Cellulose Gluten Samples

As part of the preparations for NIR measurements and data handling, NIR spectra were measured from samples with a 10 fold dilution series of gluten powder derived from wheat diluted with cellulose powder to produce powder samples comparable to the protein powder. Both cellulose and gluten powder are from Sigma-Aldrich, Germany. The dilutions contain 50% cellulose and 50% gluten, 60% cellulose and 40% gluten, 70% cellulose and 30% gluten, 80% cellulose and 20% gluten and 90% cellulose and 10% gluten respectively. A spectra of 100% cellulose was measured as well. See Table 11 in Appendix A1 for exact weighed amounts.

The cellulose gluten samples were prepared in 50 ml tubes from Sarstedt, Germany using the scale AG204 DeltaRange from Mettler Toledo, Ohio, U.S., with a 0.1 mg resolution, measuring up to 81 g. They were well mixed by manually shaking for 1 minute, ensuring that the samples at hand are representative of the batch, before they were transferred to 20 ml scintillation glass vials provided by Q-Interline, Denmark, further referred to as glass vials, used for NIR measurements. The measured NIR spectra were preprocessed in MATLAB while the needed programs for multivariable data analysis were developed, making them comparable with preprocessed spectra from the protein powder.

2.4 NIR Measurements

For each sample measured by the NIR instrument, triplicate NIR spectra have been conducted if possible, to be able to look at intra-sample variation. The largest, 1 mm (for press 1 and 3) and smallest, < 0.125 mm (for press 1, 2, 4, 6, 7, 8, 9, and 10) fractions of the protein powder particle sizes did not contain sufficient amounts of powder in order to measure NIR spectra in triplicates.

The used NIR instrument is a FT-IR model FTLA2000-154 analyser with serial number 1331416-001 from ABB, Switzerland provided by Q-Interline, Denmark. At least four hours prior to the measurements, the NIR instrument was turned on to allow the cooling detector to reach its set point and the system to be stable. Prior to all measurements, it was manually checked that the instrument had a clean window. When placing the background glass vial on the NIR instrument, it was also ensured that a peak% between 20% and 80% was detected. The instrument automatically subtracts the measured background spectrum from all analysed spectra. A glass vial containing PTFE (polytetrafluoroethylene), also known as teflon, from ABB, Switzerland provided by Q-Interline, Denmark, was used as background.

NIR measurements require correct presettings to obtain usable spectral quality explained and presented in the following and summarised in Table 3.

Sample Vial Solid powder samples were placed in a 20 ml scintillation glass vial provided by Q-Interline, Denmark with an internal diameter of 27.4 mm. Glass vials are used for all measurements in this project due to their light properties. Glass has a refractive index of 1.5, leading to reflection losses at the surface, only resulting in an approximately 4% decrease in light intensity [42].

Sample Placement A sample placed in the NIR instrument used in this project when in reflectance mode, can be either still standing or rotating. For this project, rotating samples were analysed. Glass vials were filled up at least to the required minimum sample level to cover the entire light beam, see Figure 17. The vials were rotated at an angle of 45°using a spinner mounted on the NIR instrument. This analysing method allows for the sample to tumble inside the vial, and thus measures a bigger area of the sample than a stationary method, leading to a more representative spectrum when looking at heterogenous samples.

Resolution The resolution corresponds to the number of scans, which shall be identical for the background measurement and all samples that shall later be compared. A lower resolution leads to more scans. The resolution is chosen separately for each different kind of measured sample, see Table 3.

Scans When rotating a sample, a number of scans corresponding to one or more complete rotations of the glass vial were chosen, for the spectrum to cover the entire circumference of the sample. The number of scans corresponding to the time one rotation of the glass vial took, was read from the computer software program FTIR Control Panel, and used as presetting.

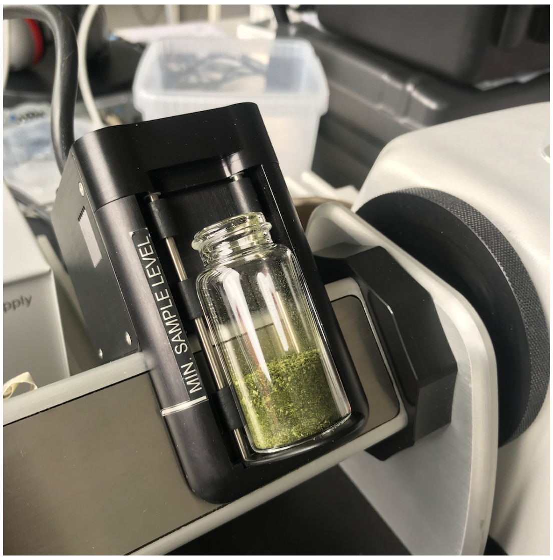


Figure 17 Rotating glass vial on FT-NIR. The glass vial is filled up at least to the minimum required sample level to cover the entire light beam. Photo: Christina A Andersen, 2020

Gain Gain that is also known as the signal amplification, is adjusted according to the peak% signal of the background sample, and according to the expected peak% variation during all measured samples. Identical gain settings have to be used for all samples that shall be compared. Accordingly, the peak% was tested for the two most extreme samples in each of the two powder experiments. The peak% was read from live spectra with the software program FTIR Control Panel, and should be between 20% and 80% for ensuring optimum performance. A peak% higher than 80% is not preferred, since the detected light signal could easily reach the 100% limit and thus the measured spectrum is not usable. A too low peak% leads to a too low detected light signal is, and thus a low signal to noise ratio. The highest peak% is seen for the sample reflecting most of the light.

Data Type The data output type is decided by how the instrument is set up, that is if the NIR instrument is assembled for use in the reflectance or transmission mode. For this project, the reflectance mode is chosen. It was possible to choose between absorbance or transmission in the computer settings. When reflectance should be measured in this project, the chosen computer setting was transmission (%trans). Choosing %trans as computer setting implied that the measured reflectance spectra were constructed according to the instrument set up, which could thus either have been a reflectance setup or a transmission setup. %trans was chosen due to its higher, but not too high peak% values leading to a lower signal to noise ratio, compared to the absorbance setting.

Wavenumber Range The standard wavenumber range setting from 12,000 - 2,000 cm^{-1} was initially chosen, since the same setting showed to cover the entire informative spectral range during a protein powder test run. Although this setting was chosen, the actual total wavenumber range covers 15,792 - 15 cm^{-1}, which was found by opening the raw spectra spc-files in Unscrambler, see Figure 27 in Appendix A1.

For NIR measurements of the green juice and pellet the same presettings as for the protein powder measurements were tested when measuring the reflectance, as well as a setup measuring transmission using a smaller 1 ml glass vail with the dimension 40x8 mm, provided by Q-Interline, Denmark was tested. In addition, the green juice was shaken manually directly before the measurement to minimise sedimentation effects. Samples from the green juice and the pellet were tested at both extremes, the first and tenth press, in order to determine the peak% and thus choose the correct gain presetting.

Table 3 Summary of used NIR presettings. The presented presettings are used for all samples of cellulose gluten powder and protein powder.

	Cellulose gluten powder	Protein powder
Sample vial	20 ml scintillation glass vial, Ø 27.4 mm	20 ml scintillation glass vial, Ø 27.4 mm
Sample placement	Still standing	Rotating
Background sample	PTFE	PTFE
Spectral resolution	8 cm^{-1}	16 cm^{-1} [1]
Number of scans	16 [2]	36
Gain	High E	High E
Data output type	Reflectance (%trans)	Reflectance (%trans)
Wavenumber range	15,792 - 15 cm^{-1}	15,792 - 15 cm^{-1}

All samples are measured at room temperature. [1] Most powders will be sufficiently sampled at 16 cm^{-1} or 32 cm^{-1}, which was not known for the cellulose gluten sample measurements. [2] The standard setting of the used FT-NIR.

Finally, all spectra were collected by a provided computer connected to the NIR instrument using the software program Grams AI. The spectral values were exported in txt format, aligned in MS Excel with its corresponding wavenumbers detected with Unscrambler, and imported to MATLAB where they were further analysed.

2.5 Multivariable Data Analysis

The conducted raw NIR data was analysed by initially creating an MS Excel sheet containing all spectral data as rows and the corresponding wavenumbers as columns. This sheet was uploaded to MATLAB, where it was further analysed by multivariable data analysis roughly following these steps:

1 An initial zoomed wavenumber range is chosen as the first preprocessing. Areas clearly containing too much noise and no information are discarded. For spectra that shall be initially compared, identical wavenumber ranges are chosen.

2 It is determined whether a linear correlation between the areas under the zoomed NIR spectra and the reference values exists, in which case there would be no need for further preprocessing. For the cellulose gluten samples, a clear pattern is detected, but in order to get comfortable with the multivariable data analysis, further preprocessing was still performed. For the protein powder samples, no clear pattern was seen, and further preprocessing has to be applied.

3 With an ANOVA test, it was determined that the particle sizes did not lead to any significant difference in spectral data. Therefore, MSC does not seem to be a suitable preprocessing and is not used. A linear offset in the theoretically identical triplicate sample spectra of the original protein powder was on the other hand seen, which is why SNV is used, and expected to be the most promising preprocessing. Smoothing combined with both Der1 and Der2 were tested as well, with the presettings; window size = 11, degree of fitted polynomium = 2. This leaves three different preprocessing opportunities tested one at a time, which are compared with PCA plots and resulting PLS models.

4 The preprocessing method with the combined best PCA clustering of triplicates and PLS model according to the R^2 value, is chosen to look further into.

5 Having found the best suited preprocessing method, an initial PLS plot covering a large part of the wavenumber range, is made to examine calculated b-coefficients plotted against the wavenumber, in order to see which wavenumber ranges explain the PLS model. Additionally, the best suited cellulose gluten spectral peaks are compared with similar peaks of the protein powder, to determine which wavenumbers seem to explain relevant compounds. From this data analysis, the wavenumber used further is narrowed down.

6 Having narrowed down the wavenumber spectrum sufficiently, a final PLS plot is made, and predicted new mean TDF values are calculated and presented.

7 An outlier detection of the predicted values is made, that is whether there are outliers in between the triplicate predictions. If outliers are detected, they are not used in calculation of the mean predicted reference value.

MATLAB code developed for data handling of the cellulose gluten samples is presented as text in Appendix A3. MATLAB code developed for data handling of the protein powder samples is presented as text in Appendix A4. Copyright used MATLAB code is presented as text in Appendix A5.

2.6 Feasibility Studies

Prior to the actual measurements, feasibility studies were performed to see if the protein powder would result in usable NIR spectra using the available NIR instrument. Furthermore, the cellulose gluten samples were used to prepare the model development, and all parts of the reference enzymatic method were tested several times to minimise errors when conducting the actual reference value results.

3 Results

All collected results in this project are presented in this section of the report. The results are divided into representable subsections reflecting the research question and objectives of the study.

3.1 Protein Powder Contents

Raw alfalfa was subjected to repeated presses in order to extract the protein content. The presses were performed on freshly harvested plants and the 10 presses were performed immediately after each other. From each press, the obtained green juice was collected, pH was decreased to precipitate proteins, and the juice was centrifuged, whereafter the resulting pellet was stored at -18°C until freeze drying it into the resulting protein powder, which lastly was analysed for different compounds. The pulp was repeatably pressed for 10 times before being discarded. The weighed amounts of pulp, green juice, brown juice, protein pellet, protein powder and the amount of evaporated moisture during freeze drying is presented in Table 4.

Table 4 Protein powder production using the screw press CP-4 from Vincent Corporation. For each press from 1-10, weighed amounts of pulp (g), green juice (g), brown juice (g), protein pellet (g), protein powder (g) and the amount of evaporated moisture during freeze drying (% w/w) is presented. ND for pulp of press 1 indicates that press 1 started with the use of raw alfalfa. Initially the weight of frozen alfalfa was 50,823 g, and the weight of alfalfa including water for thawing was 81,498 g. The leftover pulp after press 10 was 5,827 g.

Press	1	2	3	4	5
Pulp	ND	16,221	12,449	9,549	8,769
Green juice	53,131	33,703	26,566	34,220	17,578
Brown juice	47,938	42,101	25,020	32,297	16,814
Protein pellet	4,834	3,959	969	1,810	693
Protein powder	818.4	511.0	136.9	268.5	99.70
Evaporated moisture	83.07	87.09	85.87	85.16	85.61

Press	6	7	8	9	10
Pulp	7,494	7,108	6,587	6,183	5,849
Green juice	14,514	15,244	13,175	12,749	15,900
Brown juice	13,572	13,696	12,091	11,789	14,653
Protein pellet	874	1,003	1,027	974	1,113
Protein powder	126.8	134.8	133.6	120.3	120.8
Evaporated moisture	85.49	86.56	86.99	87.65	89.15

The results of the laboratory work leads to uncertainties regarding exact amounts of conducted raw material when transferring the green juice from buckets to weighing glass. Also, a source of error is that the last part of each pulp fastened inside the screw press and was used for the following press.

One set of 1-10 presses with one batch of alfalfa plants (n=1) was performed for producing the protein powder, which was analysed in duplicates, that is two samples from each of the 10 presses. In order til calculate IDF and SDF amounts, one of each duplicate determined the ash content, and the other determined the protein content (the protein determinations within the enzyme assay could unfortunately not be used, see Section 2.2). Thus duplicate measurements resulted in one IDF and SDF value respectively. Independent of the enzyme assay, ash and protein contents were determined in triplicates, and within the assay D-glucose, D-fructose and thus ACH was determined in duplicates. Standard deviations (SD) are included if more than one measurement was performed. They do not exist for the determined fibre contents, since IDF and SDF was only determined once. The moisture content was experimentally found to be 0.00% w/w by drying the protein powder. The determined protein powder contents of each of press 1-10 are summarised in Table 5. IDF, SDF and the sum of IDF and SDF, presented as TDF is included in the table, as well as D-glucose, D-fructose and the sum of D-glucose and D-fructose, presented as ACH. Thus, the sum of all presented components is higher than 100% w/w.

Table 5 Enzymatically determined protein powder contents (% w/w) ± SD. The determined protein powder contents of press 1-10 are presented as protein (n=3), IDF (insoluble dietary fibre) (n=1), SDF (soluble dietary fibre) (n=1), TDF (total dietary fibre) (n=1), D-glucose (n=2), D-fructose (n=2), ACH (available carbohydrates) (n=2) and ash (n=4). Standard deviations are included if n > 1. IDF, SDF and the sum of IDF and SDF, presented as TDF is included in the table, as well as D-glucose, D-fructose and the sum of D-glucose and D-fructose, presented as ACH. Thus, the sum of all presented components is higher than 100% w/w.

Press	1	2	3	4	5
Protein	43.12 ± 0.26	44.08 ± 0.83	47.40 ± 0.93	40.07 ± 0.68	42.14 ± 1.06
IDF	21.13	24.83	29.34	35.03	34.62
SDF	1.67	1.21	2.17	2.22	2.41
TDF (IDF + SDF)	22.80	26.04	31.52	37.25	37.03
D-glucose	3.15 ± 0.63	2.27 ± 0.48	1.79 ± 0.76	1.20 ± 0.65	2.14 ± 0.00
D-fructose	2.29 ± 0.47	0.30 ± 0.30	0.04 ± 0.06	0.38 ± 0.53	0.00 ± 0.00
ACH (D-glucose + D-fructose)	5.43 ± 1.10	2.57 ± 0.18	1.83 ± 0.82	1.58 ± 1.18	2.14 ± 0.00
Ash	8.24 ± 0.28	3.15 ± 0.55	3.07 ± 0.37	2.72 ± 0.28	2.40 ± 0.18

Press	6	7	8	9	10
Protein	41.29 ± 0.77	39.08 ± 0.13	38.69 ± 0.59	38.24 ± 0.82	37.84 ± 0.80
IDF	39.36	41.56	42.16	44.94	45.61
SDF	1.51	1.71	1.82	1.77	1.86
TDF (IDF + SDF)	40.87	43.27	43.98	46.70	47.47
D-glucose	1.61 ± 0.12	1.00 ± 0.12	1.06 ± 0.07	0.77 ± 0.12	0.55 ± 0.18
D-fructose	0.34 ± 0.36	0.38 ± 0.30	0.04 ± 0.06	0.17 ± 0.00	0.55 ± 0.78
ACH (D-glucose + D-fructose)	1.95 ± 0.24	1.38 ± 0.42	1.10 ± 0.13	0.94 ± 0.12	1.10 ± 0.61
Ash	3.16 ± 0.61	2.95 ± 0.50	2.95 ± 0.57	2.52 ± 0.25	2.70 ± 0.29

Generally the enzymatically determined protein powder contents are as expected. The samples being rewetted and pressed most times contain less protein and more fibre. The produced pellet from each press, being freeze dried into the protein powder, consists of the total water soluble solid content extracted from each press of alfalfa. The first press is derived from raw alfalfa, while the remaining presses are derived from the pulp of the prior press, the pulp containing more fibre after each press. At some point the pulp consists of a fibre to protein ratio that is too high to be worth continuing the process of rewetting and pressing the pulp. The proteins of alfalfa are more water soluble than the fibres, thus existing in the largest extent in the first presses.

Looking closer at the determined protein contents, the protein powder of press 2 and 3 contains the most protein, which deviates from the general picture. This could be explained by a mechanical destruction of the product by the screw press that had only happened to a minimal extent in press 1. By mechanically destructing the plant cells of alfalfa, the proteins are easier obtainable. A double screw press would have been more efficient and might have resulted in a high amount of proteins already after the first press.

IDF which according to theory contains mostly of water insoluble cellulose, hemicellulose and lignin is according to theory accounting for the largest parts of TDF in alfalfa. Even though the protein powder is derived from the water soluble solid content extracted from each press of alfalfa, the TDF content accounts for large parts of the powder, according to findings in this project. This can be explained by the mechanical destruction of the product due to the screw press, and thus a destruction of the fibre molecules. Within the screw press the biggest particle size of the produced green juice is affected by a filter. If a filter with a lower particle size allowance had been used, the amounts of fibre would be expected to be lower, but so would the desired amount of protein.

Throughout the absorbance measurements used for ACH determination, very low changes in the differences of the measured absorbance values due to facilitated reactions were detected for each sample. The differences in absorbance values were too low, to determine sufficiently accurate results according to the enzyme assay. Especially the determined amounts of D-fructose were affected hereby, leading to barely any detection of D-fructose compared to D-glucose.

The high ash content in press 1 compared to the rest of the presses might be due to minerals and other compounds found on the surface of the unwashed raw alfalfa prior to the protein powder production.

All determined contents sum up to between 75% w/w and 90% w/w, the lowest determined amount for the first presses and the highest determined amount for the last press. In these calculations, lipids are not included, and as the calculated standard deviations indicate, method uncertainties also have to be taken into account.

3.2 NIR Measurements

Throughout this subsection, all collected NIR spectra and relevant spectral preprocessing is presented.

3.2.1 Cellulose Gluten Spectra

As part of the preparations for NIR measurements and data handling, NIR spectra were measured from samples with a 10 fold dilution series of gluten powder diluted with cellulose powder to produce samples comparable to the protein powder. Figure 23B in Section 3.3.2, presenting clear spectral peaks for raw zoomed cellulose gluten spectra, is used as comparable spectral data, when analysing the protein powder spectra.

It was determined that a wavenumber range of 5,290 - 3,960 cm^{-1} combined with SNV preprocessing resulted in the highest degree of PCA clustering and best PLS model regarding linearity, Figure 18. Since the cellulose gluten powder mainly consists of cellulose, it is expected, that the clear peaks of the spectra within the chosen wavenumber range mainly reflect the cellulose contents, which might also be confirmed by the close to perfect PLS model of these spectra, see Figure 18B. The corresponding PCA plot is presented in Figure 18A. Mean spectra from SNV preprocessed triplicate data are calculated, and with a PLS model related to the reference cellulose content, see Figure 18B.

Appendix A2 presents raw NIR spectra obtained from the cellulose gluten powders in Figure 28 and 29. The SNV preprocessed spectra are presented in Figure 30 in Appendix A2. Table 12 in Appendix A2 presents the predicted cellulose contents from the PLS model related to the reference cellulose contents.

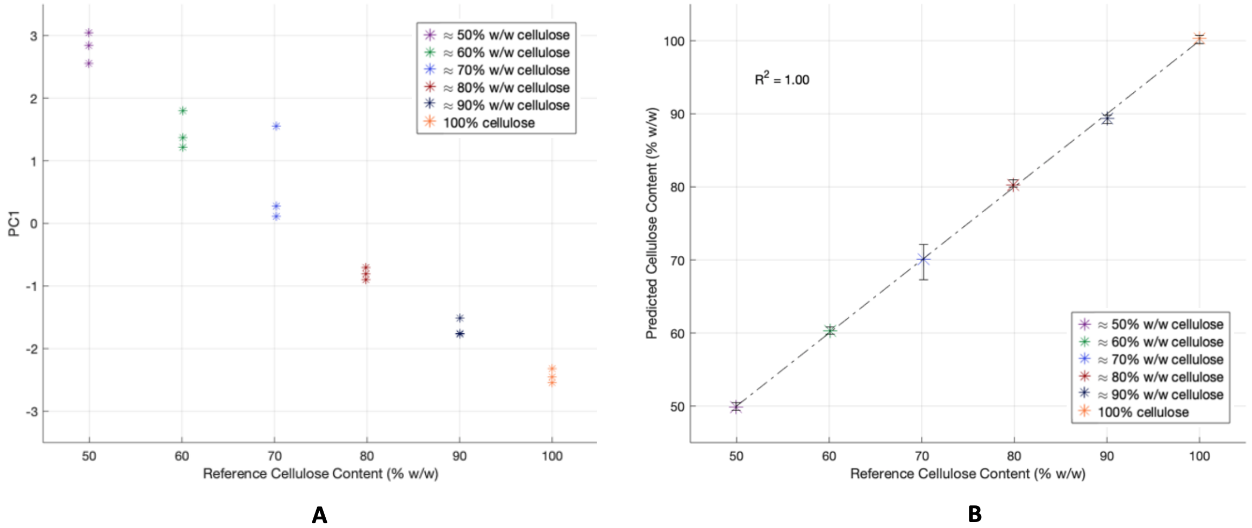


Figure 18 The triplicate sample clustering presented by a PCA (principal component analysis), and a PLS (partial least squares) plot with cellulose as reference contents and SNV preprocessed spectra. Both plots are based on the zoomed NIR spectra (5,290 - 3,960 cm^{-1}) of cellulose gluten powder for all 6 cellulose gluten powder triplicates. (A) The PCA plot with PC1 (Principal Component 1) plotted against the reference content cellulose. PC1 is PCA constructed variables that explains most of the data variation in the observed wavenumbers. (B) PLS model of cellulose gluten powder. Determined R^2 of the PLS model was 1.00, calculated with respect to a x=y line. Error bars are showing the variation in predicted cellulose content from the PLS model, while the coloured dots represent mean predicted values.

3.2.2 Freeze Dried Protein Powder of Different Particle Sizes

As particle size is known to have an impact on the outcome of measured NIR spectra, the original protein powder was sieved into fractions of different particle sizes, which were analysed and compared. The freeze dried protein powder milled down to 1 mm is further referred to as the original protein powder. Figure 19 presents a visual overview of the different sieve sizes used for dividing the particle fractions, including the original particle size. Raw spectra containing protein powder of these different particle sizes are presented in Figure 31, 32, 33, 34 and 35 in Appendix A2.

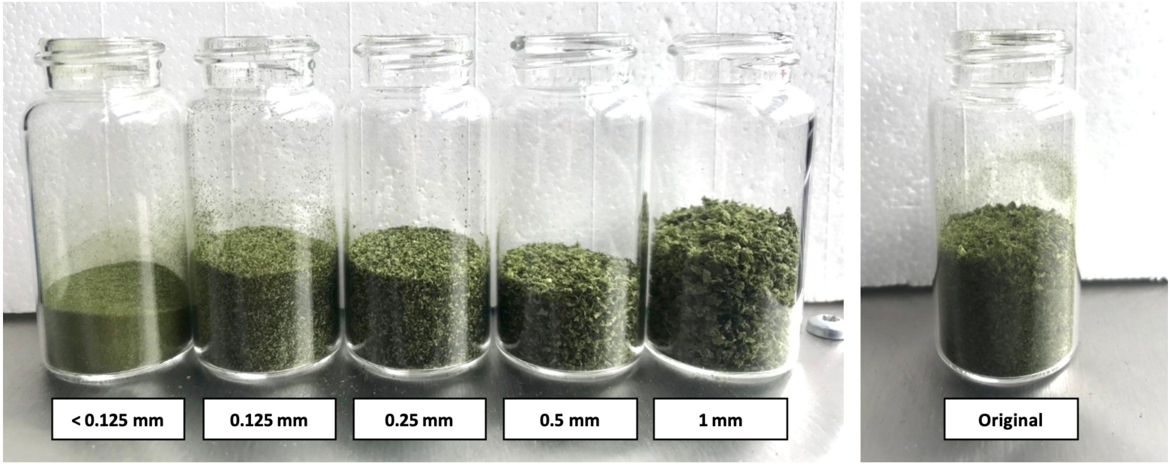


Figure 19 Visual overview of the different protein powder fractions obtained by sieving with sieve sizes presented in the figure. Original particle size refers to freeze dried protein powder milled down to 1 mm. Photos: Christina A Andersen, 2020.

Since the kind of model building performed in this project depends on the location of clear spectral peaks, similar peak locations of these different spectra, might result in similar derived models. Whether seven clear peak locations in a mean spectrum from each different particle size significantly differ from each other, is determined with a one way ANOVA test, since the data set is found to be normally distributed. Table 13 in Appendix A2 presents these spectral peak locations. An ANOVA test with a significance level of $\alpha = 0.05$ was used. A P-value of 1.00, being larger than α means that the null hypothesis H_0 is accepted, and that the differences between the mean spectral peaks of the different particle sizes are not statistically significant. Since particle size differences do not significantly affect spectral peak locations, it is decided to develop the model of this project from the original particle size, since that is the easiest obtainable size. The MSC preprocessing method is not looked further into, since it is found that particle size does not affect the interesting information.

Protein Powder of Original Particle Size Raw spectra containing protein powder of the original particle size (freeze dried protein powder milled down to 1 mm) are presented in Figure 20. The received peak% signal when obtaining these spectra was between 30% and 40%, which meets the requirements for for optimum performance, since it is inbetween 20% and 80%.

At both ends of the spectral data, > 8,000 cm^{-1} and < 3,500 cm^{-1}, the raw NIR data contains a large amount of noise. Either no light or all light is detected resulting in fluctuating spectra with no information in the given area, and thus no useful result. Only noise from the instrument is detected. Such parts of the spectra are discarded, as well as the wavenumber range 4,000 - 3,500 cm^{-1}, since NIR spectroscopy does not cover this area, and thus no potential compounds are found here. Also, the wavenumber range 8,000 - 7,300 cm^{-1} is discarded, since this area does not represent any compounds of special interest, and at the same time contains some extent of noise.

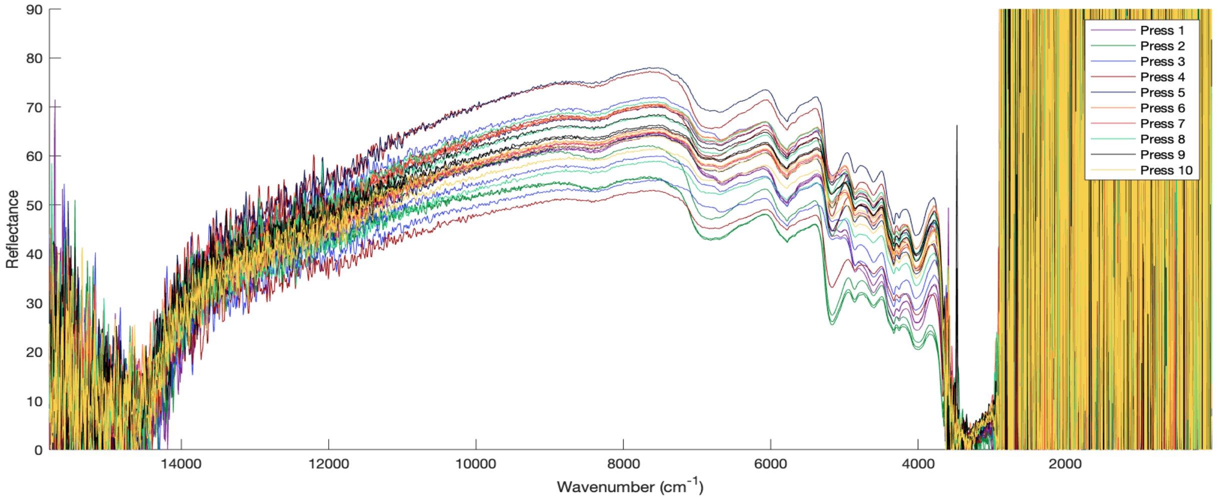


Figure 20 Raw NIR spectra of original particle size protein powder. Reflectance as a function of wavenumber is presented for all 10 presses of rewetted alfalfa pulp in triplicates. Original particle size refers to freeze dried protein powder milled down to 1 mm.

An initially zoomed part of the raw spectra is presented in Figure 21. Peaks of interest and spectral differences within measured triplicate samples, and between each of the 10 presses do now appear clearer. The triplicate samples, presented by identical colours in Figure 21 are similar in curve appearance, with the biggest difference being the light path length difference. No triplicate sample clearly stands out in another way. The different presses deviate mostly between 5,600 - 4,600 cm^{-1}.

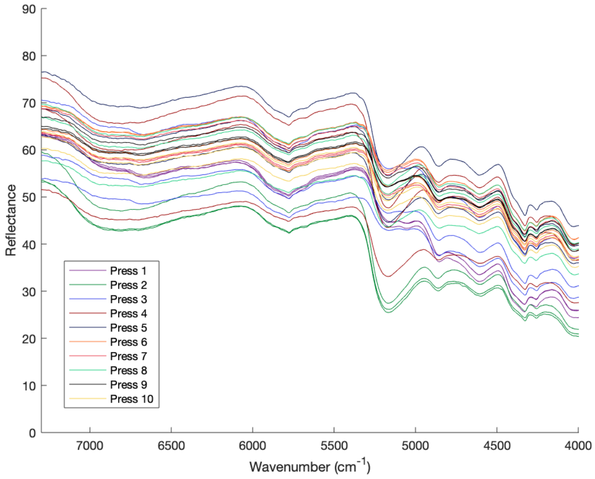


Figure 21 Zoomed NIR spectra (7,300 - 4,000 cm^{-1}) of original particle size protein powder. Reflectance as a function of wavenumber is presented for all 10 presses of rewetted alfalfa pulp in triplicates. Original particle size refers to freeze dried protein powder milled down to 1 mm.

3.2.3 Green Juice and Pellet

In addition to the freeze dried protein powder, samples of green juice and pellet were analysed with NIR spectroscopy. Using the presettings obtained during measurement preparations, it was however determined that spectral data from the green juice and pellet were not usable. For the green juice, the received peak% signal was 1-2%. For the pellet, the received peak% signal was 5-8%.

3.3 Model Development

Since no triplicate sample stands out, and the biggest difference between the triplicate samples lies in the light path length difference, it was decided to continue data preprocessing with all triplicate samples as presented in Figure 21.

3.3.1 Comparing Preprocessing Methods

From the initial zoomed spectral data, it was compared how well no preprocessing, SNV preprocessing, Der1 preprocessing and Der2 preprocessing respectively would fit to a PLS model, with reference values being the determined TDF. How these preprocessing methods affect the spectral data is shown in Figure 36 in Appendix A2. It was seen that the initial clear peaks represented in the zoomed spectra in Figure 21 are all represented in all three preprocessed spectra. From these preprocessed spectra, it can also be seen that SNV preprocessing separates the triplicate samples to the highest extent, also when taking the spectra with no preprocessing into account.

The highest extent of triplicate separation seen for SNV preprocessing, is thus confirmed by Figure 22, showing the triplicate sample clustering in PCA plots of respective compared spectral data. PC1 explains most of the data variation in the observed spectral matrix X of all four compared spectral data, the percentage being shown in each respective figure text. All score values of PC1 separate the triplicate samples, but not in a structured way relating to the TDF reference values. Score values differing most from 0 represent the NIR spectra of samples standing out to the highest extend from the rest. The score values of PC2 were looked at as well (data not shown), but no usable separation of triplicates was seen. These unstructured results might indicate that these NIR spectra do not contain the relevant information in order to fit a model to the TDF contents. The same tendencies were found for the rest of the measured nutrient contents used as reference values (data not shown).

A PLS model is developed for each of the compared spectral data. Table 6 explains the accuracy of each of these PLS models. A good PLS model is represented by a low number of PLS components combined with a high percent

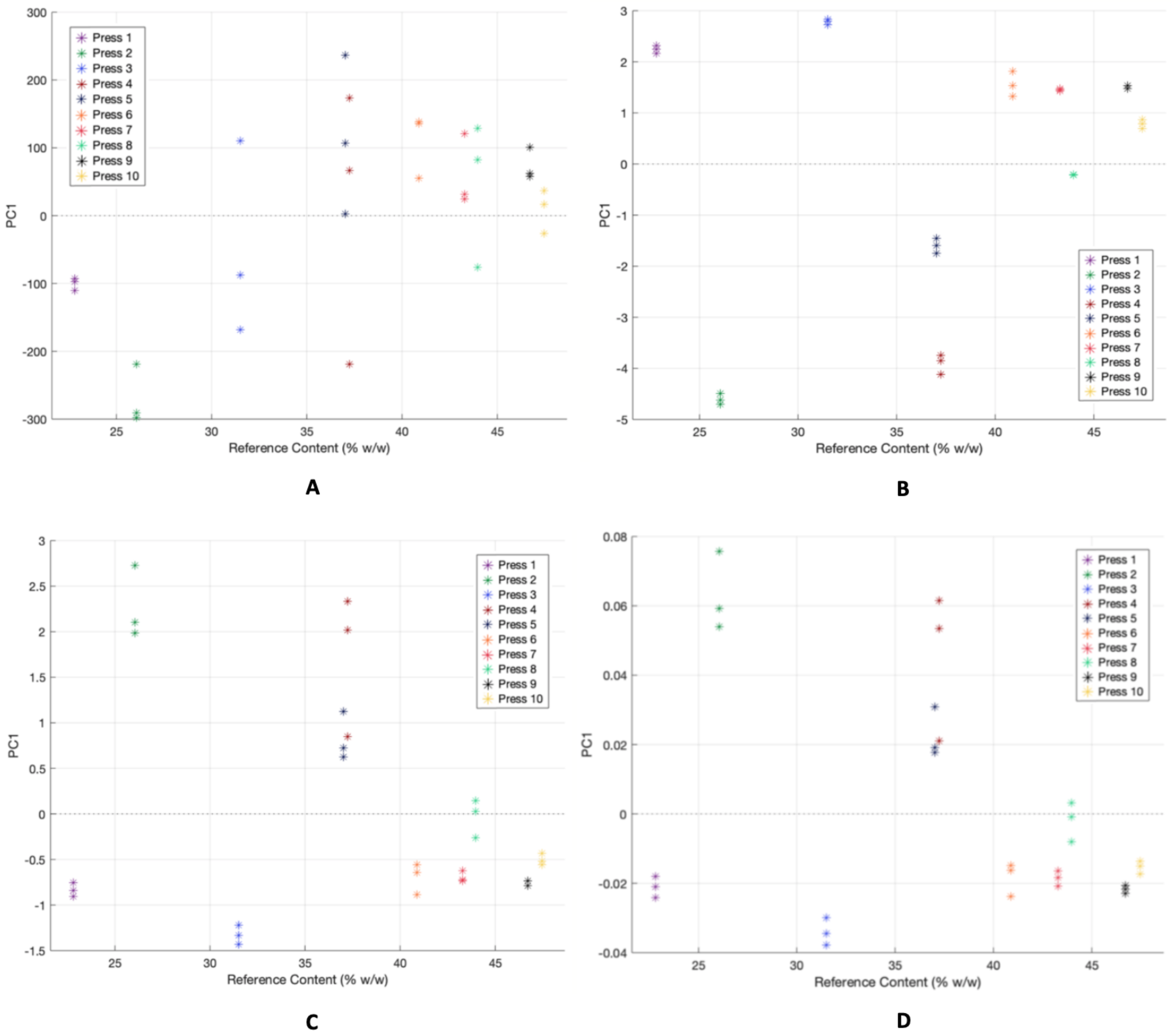


Figure 22 The triplicate sample clustering presented by PCA (principal component analysis) plots of PC1 (principal component 1) from spectra with different preprocessing methods, plotted against TDF (total dietary fibre) as reference contents. PC1 is PCA constructed variables that explains most of the data variation in the observed wavenumbers. Each PCA plot is based on the zoomed NIR spectra (7,300 - 4,000 cm^{-1}) of original particle size protein powder, for all 10 presses of rewetted alfalfa pulp. Original particle size refers to freeze dried protein powder milled down to 1 mm. (A) Represents spectra with no preprocessing. PC1 explains 93.9% of the observed spectral matrix X. (B) Represents SNV (standard normal variate) preprocessed spectra. PC1 explains 85.0% of the observed spectral matrix X. (C) Represents 1st derivative preprocessed spectra. PC1 explains 81.6% of the observed spectral matrix X. (D) Represents 2nd derivative preprocessed spectra. PC1 explains 53.0% of the observed spectral matrix X.

variance explained in reference matrix Y and a high R^2 value that is related to the straight line x = y. The results in Table 6 indicate that the SNV preprocessed spectra produce a PLS model with the highest accuracy for all three parameters.

From here, it was decided to further continue with SNV preprocessed spectra, since that shows the best PCA clustering, and with the least PLS components produces the best PLS model.

Table 6 Accuracy of tested PLS models with different preprocessing methods. Zoomed NIR spectra of original particle size protein powder from 7,300 - 4,000 cm^{-1} that are not preprocessed are compared to the preprocessing methods SNV (standard normal variate), 1st derivative and 2nd derivative. The number of used PLS (partial least squares) components are presented together with the percentage of variance explained in reference matrix Y, and R^2 values for each PLS model.

Method	Number of PLS components [1]	R^2
No preprocessing	5 (97.5%) [2]	0.9744
SNV	3 (98.2%) [2]	0.9812
1st derivative	4 (97.7%) [2]	0.9763
2nd derivative	4 (97.9%) [2]	0.9784

[1] Should be as low as possible to not risk overfitting. [2] Percent variance explained in reference matrix Y. As more components are added to the model, the model will do an apparently better job fitting the original data Y, simply because at some point most of the important predictive information in Y will be present in the components.

3.3.2 Choosing Wavenumber Range

To be able to decide which wavenumber range is best suited for the reference TDF values, it is determined where compounds of interest might show spectral peaks, combined with determining which wavenumbers PLS model determined b-coefficients affect the model.

Spectral Compound Determination Since the protein powder is a complex sample containing many different compounds, the peaks in the protein powder spectra are compared with the peaks in the less complex cellulose gluten spectra, see Figure 23. Exact peak locations are determined from a mean spectrum of all spectra in respective figure. Table 7 presents the potential compounds related to the peak numbers.

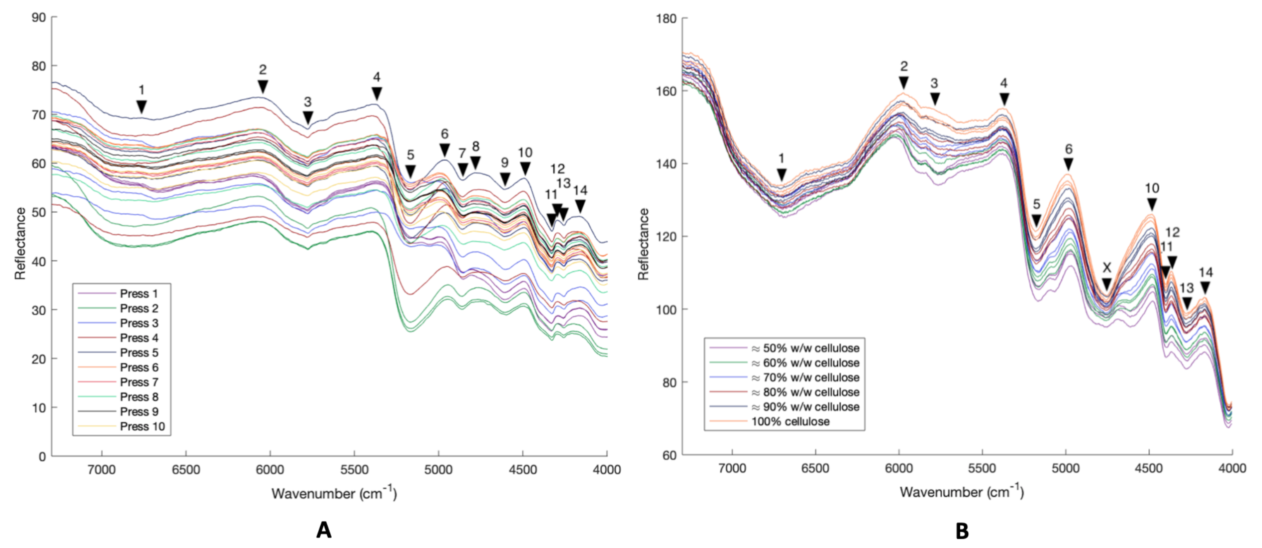


Figure 23 NIR spectra (7,300 - 4,000 cm^{-1}) of (A) original particle size protein powder and (B) cellulose gluten powder respectively including peaks. Reflectance as a function of wavenumber is presented, with peaks marked with numbers corresponding to their occurrence. (A) All 10 presses of rewetted alfalfa pulp in triplicates are presented. Original particle size refers to freeze dried protein powder milled down to 1 mm. (B) All 6 cellulose gluten powder triplicates are presented.

From the cellulose gluten powder spectra in Figure 23B, it is observed that the reflectance measurements at peak 5, 6, 10, 11 and 12 strictly follow the sample composition, whereas the remaining peaks do not correlate as good to the sample composition. Peak 5 and 10 could be potential water peaks, but since such a good sample composition correlation is seen, it might indicate, that water does not dominate the spectra. Since the raw cellulose gluten powder spectra explain the sample composition to this great extend, they are directly used regarding compound explanation.

Table 7 Spectral peak similarities of protein powder and cellulose gluten powder samples. Peak numbers are correlated to at which wavenumber they are detected, the corresponding overtones and combination bands, and to possible detected molecular bonds.

Peak	Wavenumber (cm^{-1})	Overtones and combinations	Possible detected bonds
1	6,700	NH	R-NH_2
2	6,000	CH	CH_3, CH_2
3	5,800	CH, SH	CH_3, CH_2, CH, SH
4	5,400	C=O	R-COOH
5	5,200	C=O, OH	H_2O, R-COO-R, POH, $CONH_2$
6	4,950	C=O, OH	CO
10	4,500	NH+OH, CH+CH	H_2O, CH_3, CH_2
11, 12, 13, 14	4,400, 4,300, 4,250, 4,150	CH+CH, CH+CC	CH_3, CH_2, CH

The protein powder though, does not show similarly clear raw spectra, thus when having determined the clear peak locations from Figure 23A, Figure 25A is looked at further, regarding compound explanation at these relevant peaks.

The TDF (and IDF) amounts containing cellulose, in the protein powder do not strictly follow the number of presses. The TDF (and IDF) amounts increase in the order; press 1, 2, 3, 5, 4, 6, 7, 8, 9 and 10. If this order is detected at peaks expressing compounds relatable to cellulose, they would account for the most suited peaks for at model predicting the TDF content. Peak 5, 6, 10, 11 and 12 strictly follow the sample composition of the cellulose gluten powder are therefore further looked at for the protein powder as well.

Peak 5 and 6 are detected as wavenumbers assigned to the combination band of OH bonds. This place in the spectra should thus detect if samples contain differences in compounds containing a large extend of OH bonds, for example cellulose. Both peak 5 and 6 of the protein powder though, do not show the same clear tendencies of ordering the samples according to their cellulose content as does the cellulose gluten powder.

Peak 10, 11 and 12 are detected as wavenumbers assigned to the combination band of CH bonds amongst others. This place in the spectra should thus detect if samples contain differences in compounds containing a large extend of CH bonds. Cellulose was therefore expected to be more expressed at peak 5 and 6, but could also be expressed at peak 10, 11 and 12. Peak 10, 11 and 12 in Figure 25A are again not ideally sorted according to the cellulose contents.

By looking subjectively at the discarded brown juice of the protein powder production process, it was seen that its colour changed with the number of presses from light brown, to light green, to being clear. This might indicate that a changing parameter for press 1-10 and for the model development, other than the measured nutrient contents, could be colour compounds like chlorophyl.

b-Coefficient Determination A large wavenumber range from 10,000 - 4,000 cm^{-1} is chosen and SNV preprocessed, see Figure 24A. To see at which wavenumbers the PLS b-coefficients affect the PLS model, a corresponding plot showing the b-coefficients of each wavenumber is shown in Figure 24B. Figure 24B confirms that the wavenumber range used to initially compare preprocessing methods from 7,300 - 4,000 cm^{-1} was suitable, since wavenumbers from 10,000 - 7,300 cm^{-1} show a b-coefficient value close to 0. The PLS model is mostly affected by high b-coefficients, the relevant clear ones seen at 7,000, 6,000, 5,000, 4,500 and 4,000 cm^{-1}. At 7,000 cm^{-1} no clear known peak is seen according to Figure 23A, this wavenumber is therefore not included in the chosen range, which is started at 6,800 cm^{-1} to include peak 1, and stopped at 4,100 cm^{-1}, since also no known peak is seen after peak 14 at 4,150 cm^{-1} in Figure 23A.

Had more time been available, it would have been possible to develop software for cutting out middle parts of the spectra. In this project it is thus only possible to look at a complete wavenumber range. The SNV preprocessed spectra, b-coefficients and PLS plot for this chosen final wavenumber range is shown in Figure 25. From Figure 25B it is seen that close to the entire chosen wavelength range explains the produced PLS model, since almost all

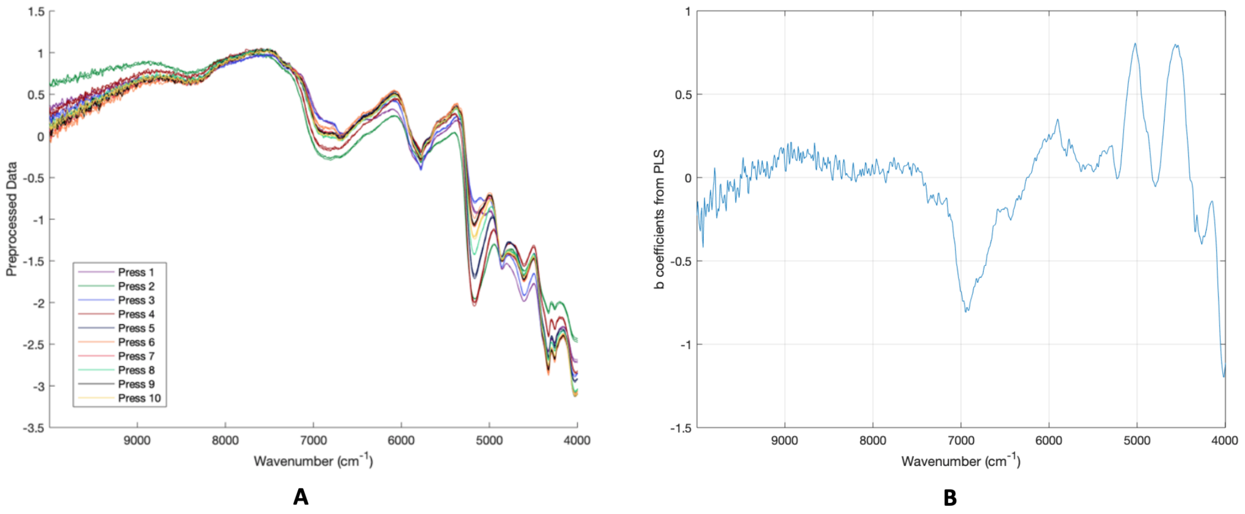


Figure 24 (A) SNV preprocessed NIR spectra of original particle size protein powder, and (B) the corresponding b-coefficients plot derived from a PLS model (10,000 - 4,000 cm^{-1}). (A) Reflectance as a function of wavenumber is presented for all 10 presses of rewetted alfalfa pulp in triplicates. Original particle size refers to freeze dried protein powder milled down to 1 mm. (B) Number of used PLS components = 3, with 98.2% variance explained in Y. Determined R^2 of the PLS model was 0.9812.

b-coefficients deviate from 0. Clear peaks in the spectra can be seen where known water bands often occur. This might indicate that the samples have not been completely dry when measured. Although they were determined to be completely dry before the NIR measurements, they could have absorbed water from air humidity during storage and transfer into the NIR glass vials. Also, some parts from some presses of the freeze dried protein powder showed resistance when being milled down into the original protein powder of 1 mm. If water within samples were an issue, it could have been checked by measuring the moisture content directly after the NIR measurements. Potential clear water peaks in the spectra are seen at peak 5 and 10 at 5,200 and 4,500 cm^{-1} respectively. The peak at 5,200 cm^{-1} does not seem to affect the PLS model much, since the b-coefficients around that wavenumber are close to zero. At 4,500 cm^{-1} the model is affected more. High b-coefficients at known water band wavenumbers could result in a poor model quality. Figure 25C shows the produced PLS model, with triplicate samples occurring close to each other, which is preferred. Press 5 and 9 show the highest degree of triplicate sample gathering.

From the produced PLS model in Figure 25C, predicted mean TDF values are calculated. The predicted values from 22.84 % w/w TDF to 45.75 % w/w TDF, as well as the deviation from the actual reference values are presented in Table 8. In Table 8 two outliers are detected with the used method, the median absolute deviations method (MAD), one from press 5 and 9 respectively. If a value is more than three scaled median absolute deviations (MAD) away from the median of the data, an outlier is detected [36]. As mentioned, press 5 and 9 show the highest degree of triplicate sample gathering in the PLS plot in Figure 25C. Thus a potential outlier is detected even though the actual deviation is smaller than for triplicate samples in the remaining presses, and this outlier detection does not highly affect the outcome of the comparison between predicted mean values and reference values. It could even be, that keeping these theoretical outliers would result in a more robust model, since a possible source of error is that an outlier is detected, which actually corresponds to a true variation in data [24].

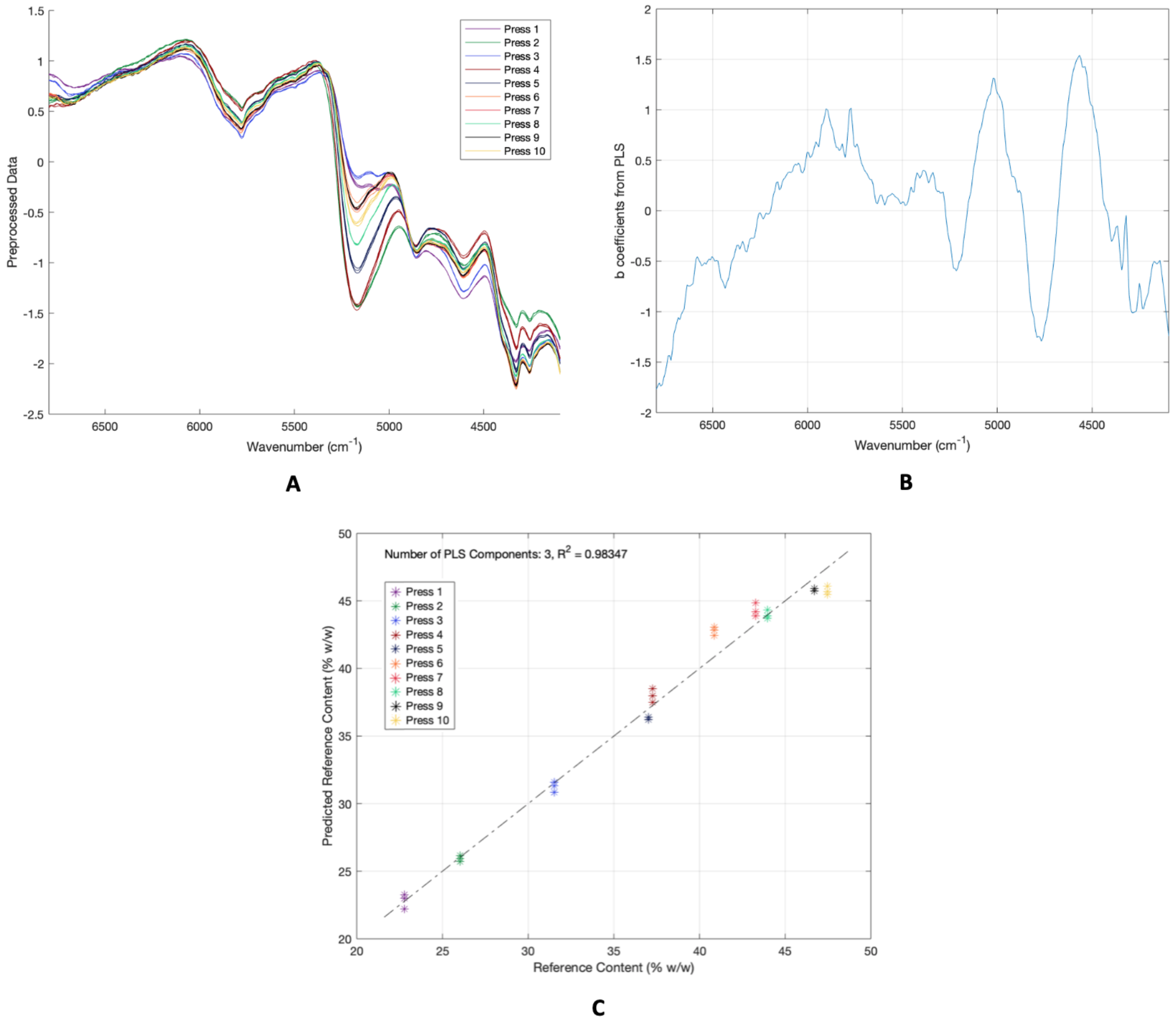


Figure 25 (A) SNV preprocessed NIR spectra of original particle size protein powder, (B) the corresponding b-coefficients plot (6,800 - 4,100 cm^{-1}) derived from (C) the PLS model. (A) Reflectance as a function of wavenumber is presented for all 10 presses of rewetted alfalfa pulp in triplicates. Original particle size refers to freeze dried protein powder milled down to 1 mm. (B) b-coefficients plot with peaks affecting the PLS model. (C) PLS model with number of used PLS components = 3, with 98.4% variance explained in Y. Determined R^2 of the PLS model was 0.9835.

Table 8 Predicted TDF (total dietary fibre) contents (% w/w) from the PLS model including standard deviations. For each of the 10 presses the PLS model has predicted a mean value of the TDF content within the analysed protein powder, as well as standard deviations (SD), the deviation from the actual measured reference values and lastly, the actual measured reference values are presented.

Sample	Predicted mean values ± SD	Deviation from reference	Reference
Press 1	22.84 ± 0.54	0,04	22.79
Press 2	25.93 ± 0.17	0.11	26.04
Press 3	31.26 ± 0.38	0.26	31.52
Press 4	38.00 ± 0.50	0.75	37.25
Press 5	36.22 ± 0.02 [1]	0.81	37.03
Press 6	42.79 ± 0.31	1.92	40.87
Press 7	44.32 ± 0.50	1.05	43.27
Press 8	43.98 ± 0.31	0.00	43.98
Press 9	45.73 ± 0.01 [1]	0.97	46.70
Press 10	45.75 ± 0.28	1.72	47.47

[1] One detected outlier for this press is not considered in the mean calculations.

4 Discussion

Throughout this section, the presented results are discussed and evaluated according to theory, and against other findings and future work within this area of research.

Determination of TDF by traditional methods such as the enzymatic determination used as reference method in this project, is very time consuming, taking several days to complete. Thus a rapid method for determination of TDF would be preferable. The result of this project was a model, derived from NIR measurements and reference enzymatically obtained TDF values, which was able to predict TDF values of alfalfa protein powder. A lack of available model validation software and time to produce independent validation sets, made it not possible to validate the model.

NIR model evaluation of TDF in protein powder derived from re-presses of the legume alfalfa has not yet been published. The obtained results thus contribute to existing NIR studies on alfalfa by increasing the knowledge within the field of NIR modelling possibilities.

An interesting result when visually comparing the spectra of different protein powder particle sizes, is that a decrease in particle size, shows a higher reflectance signal, and thus a smaller absorbance, which is a known phenomenon in NIR measurements. This observation could be explained by the increased particle density due to the smaller particle size [21]. In a NIR study by Ramalho et al, specifically evaluating particle size influence on NIR reflectance spectra, a higher reflectance signal due to a decrease in particle size were confirmed. Also, the claim that spectral peaks were not influenced by particle size were confirmed, which was claimed in the Results part and leading to the decision to continue data handling with only the original particle size of the protein powder [43].

Ramalho et al obtained different particle sizes by using sieves distinguishing between the particle sizes 0.42 mm, 0.25 mm, 0.15 mm and $<$ 0.15 mm. A slightly better NIR model accuracy was shown for the $<$ 0.15 mm particle size, which might indicate that these particles are more evenly distributed and better homogenised [43]. Looking at the spectra in this project in Figure 31, 32, 33, 34 and 35 in Appendix A2 with this knowledge, the lowest particle sizes do tend to show more evenly distributed spectra over the total spectral range of reflection. Especially this trend is seen in Figure 34. These findings might indicate that the produced model in this project could be more accurate, if derived from samples of a lower particle size. That would require a larger extent of sample preparation, which might not compensate for the extent of increased accuracy.

Since a broad wavenumber range is used for the prediction of TDF in this project, a risk exists that it is easy to fit the spectra to almost any kind of reference values, even though the spectra might not actually describe those reference values. Had a smaller part of the spectrum been chosen instead, a risk of excluding important information would have existed. In a study by Kim et al using NIR measurements to look into the TDF contents in complex homogenised, dried and defatted meals, a wavenumber range from 9,090 - 4,000 cm^{-1} is analysed [5]. This study includes a validation, and states that the developed model could be further used for screening TDF within the examined homogenised meals, which might indicate that the used wavelength range for this project is not too broad.

NIR spectra from complex samples such as this protein powder, are often hard to interpret compared to spectra only containing one or few compounds. The resulting model prediction might thus be less precise, since the spectra are results of many compounds existing in the sample, which is also found to be an issue in a study by Kim et al looking into the fibre amounts of homogenised meals with NIR [5].

Within this project, just one batch of alfalfa was analysed. The produced model is developed to fit a powder derived from a legume containing a complex biological system that changes both due to geographical and seasonal changes. Thus it has to kept in mind that therefore, amongst other mentioned factors, the robustness of the model might be limited.

4.1 Sources of Error

The collected NIR spectra and the determined TDF values do not relate perfectly. Sources of error to take into account for future similar studies are stated in this part.

The used glass vials for the NIR measurements might not have been completely clean, although when manually checked they seemed clean. Traces of washing liquid not removed by the lab dish washer could be a source of error [21]. By wiping of every used glass vial with ethanol, this source of error could be minimised.

The glass vials should ideally be filled by spooning samples into them, since pouring could lead to particle size separations and orientation of non spherical particles [21]. This was not taken into consideration at time of sample measurement, and could thus be a source of error.

When using a reference method together with NIR modelling, it is indirectly assumed that the reference method is free from errors. The enzyme assay reference method used in this project was performed using a limited amount of replicate measurements. The protein powder NIR spectra are presented in triplicates, an additional source of error might be varying compound concentrations within the triplicates, leading to non-systematical triplicate spectra. Additional replicate measurements would have minimised result deviations in general.

5 Conclusions

From theoretical obtained knowledge prior to the start of this project, within the research area of relating NIR spectra to sample compounds similar to this alfalfa protein powder, it was decided to determine if NIR spectra of the actual protein powder would be usable and could be related to the TDF content.

Based on the results obtained in this project, it was possible to develop a calibration model for determination of TDF contents of alfalfa. The NIR spectra derived from the protein powder of alfalfa were successfully related to the enzymatically obtained TDF reference values. The final PLS model of this project shows a good correlation of reference TDF values and predicted TDF values. It has to be kept in mind though, that the model is not validated, and therefore it is hard to draw a conclusion regarding the model quality. For higher chances of success, and in order to produce a more robust model, big datasets, and independent validation sets are required.

The results of this project supports the use of NIR equipment to determine the composition and quality of alfalfa in a nondestructive way. The results also encourage further investigation and optimisation of this kind of model development. Suggested future work with the data obtained in this project and in this area of research are thus outlined below.

5.1 Future Work

Of future work the first priority should be to validate this produced model, ideally first by cross validating it, by taking out a triplicate sample measurement, developing the model without it and determine to which extend this triplicate sample fits the model. If that looks promising, both a new independent validation set and a larger data set to produce a new calibration model is required to further test the model robustness.

The second priority should be to test additional NIR instrument presettings, different wavenumber ranges, additional data preprocessing methods and combinations of these held up against each other, in order to see which combinations would match the TDF content in the best possible way.

Additional suggested future work, in order to better determine which kind of molecular compounds are being looked at in the protein powder NIR spectra, is to relate the NIR spectra to the rest of the determined protein powder contents, and for each different content fit a separate model. Also, it could be tested to save the brown juice from each press, dry it and analyse it with NIR as well. Theoretically that would result in spectra with inverse peaks compared to the protein powder spectra, since soluble compounds of high concentration in the brown juice, would be of corresponding low concentration in the protein powder derived from the pellet. Carbohydrates would for example be represented in a larger degree in the brown juice, while dietary fibres would show opposite results.

Lastly, when the exact extend of molecular compounds of the protein powder are determined, NIR spectra could be obtained from pure powder representing each compound separately, to see which spectral peaks in the total protein powder NIR spectra should be looked at for each respective compound.

References

1. Thomasen, J.M.: Observation af Foder-Lucerne (2019). https://www.naturbasen.dk/observation/3179194/foder-lucerne Accessed 2020-07-25
2. Stender, D., Staermose, D., Heiner, C., Ruhdal, P.: Protein from Green Biomass as a Food Resource (2017). www.sustain.dtu.dk
3. EFSA: Novel Food (2020). https://ec.europa.eu/food/safety/novel{_}food{_}en Accessed 2020-07-25
4. Andersen, J.: Grøn Revolution: Dansk Græs skal Erstatte Sydamerikansk Soya (2020). https://www.dr.dk/nyheder/indland/groen-revolution-dansk-graes-skal-erstatte-sydamerikansk-soya?fbclid=IwAR0yTePGEE{_}Ow2juVBcOUbRk4-8OmDDX2IIyoFkdfboMFn-Ds8P9GbWsHSY{#}!/ Accessed 2020-08-03
5. Kim, Y., Singh, M., Kays, S.E.: Near-Infrared Spectroscopy for Measurement of Total Dietary Fiber in Homogenized Meals. Journal of Agricultural and Food Chemistry **54**(2), 292–298 (2006). doi:10.1021/jf051975b
6. Chaves, A.V., Waghorn, G.C., Tavendale, M.H.: A Simplified Method for Lignin Measurement in a Range of Forage Species. Proceedings of the New Zealand Grassland Association **64**, 129–133 (2002)
7. Megazyme: Measurement of Dietary Fiber: Current Methodology (2020). https://www.megazyme.com/focus-areas/dietary-fiber-portal/measurement-of-dietary-fiber Accessed 2020-01-24
8. Agelet, L.E., Hurburgh, C.R.: A Tutorial on Near Infrared Spectroscopy and Its Calibration. Critical Reviews in Analytical Chemistry **40**(4), 246–260 (2010). doi:10.1080/10408347.2010.515468
9. Veronesi, F., Brummer, E.C., Huyghe, C.: Alfalfa. In: Handbook of Plant Breeding 5, pp. 395–437. Springer, (2010)
10. Lorenzo, C.D., García-Gagliardi, P., Antonietti, M.S., Sánchez-Lamas, M., Mancini, E., Dezar, C.A., Vazquez, M., Watson, G., Yanovsky, M.J., Cerdán, P.D.: Improvement of Alfalfa Forage Quality and Management through the Down-Regulation of MsFTa1. Plant Biotechnology Journal **18**(4), 944–954 (2020). doi:10.1111/pbi.13258
11. Kratchunov, I., Naydenov, T.: Estimation of Lucerne Forage Quality by means of Morphological and Meteorological Data. European Journal of Agronomy **4**(2), 263–267 (1995). doi:10.1016/S1161-0301(14)80053-1
12. Milic, D., Karagic, D., Vasiljevic, S., Mikic, A., Mijic, B., Katic, S.: Leaf and Stem Chemical Composition of Divergent Alfalfa Cultivars. Biotechnology in Animal Husbandry **27**(4), 1505–1511 (2011)
13. et al Putnam, D.: Agronomic Practices and Forage Quality. Proceedings National Alfalfa Symposium (2000)
14. Sheen, S.J.: Comparison of Chemical and Functional Properties of Soluble Leaf Proteins from Four Plant Species. J. Agric. Food Chem **39**, 681–685 (1991)
15. FOSS Analytics: Fibre Analysis of Animal Feed vol. April, (2018)
16. Lattimer, J.M., Haub, M.D.: Effects of Dietary Fiber and its Components on Metabolic Health. Nutrients **2**(12), 1266–1289 (2010). doi:10.3390/nu2121266
17. W D Holloway, C Tasman-Jones, S.P.L.: Digestion of Certain Fractions of Dietary Fiber in Humans. Am J Clin Nutr. **6**, 927–930 (1978)
18. ChemAxon: Software Solutions and Services for Chemistry & Biology (2020). https://chemaxon.com/ Accessed 2020-07-29
19. Gorissen, S.H.M., Crombag, J.J.R., Senden, J.M.G., Waterval, W.A.H., Bierau, J., Verdijk, L.B., van Loon, L.J.C.: Protein Content and Amino Acid Composition of Commercially Available Plant-Based Protein Isolates. Amino Acids **50**(12), 1685–1695 (2018). doi:10.1007/s00726-018-2640-5
20. WHO/FAO/UNU Expert Consultation: Protein and Amino Acid Requirements in Human Nutrition. Technical report (2002). www.who.int/bookorders
21. Davies, A.M.C.: Introduction to NIR Spectroscopy. The Second European Symposium on Near Infrared (NIR) Spectroscopy (1993)
22. Megazyme: Available Carbohydrates and Dietary Fiber Assay Procedure. Technical report (2018). www.megazyme.com
23. Megazyme: Available Carbohydrates Dietary Fiber Assay Kit (2020). https://www.megazyme.com/available-carbohydrates-dietary-fiber-assay-kit Accessed 2020-01-28
24. Jørgensen, B.M.: Multivariate Spectrometric Methods for Determining Quality Attributes. In: Bremmer, H.A. (ed.) Safety and Quality Issues in Fish Processing, pp. 475–494. Woodhead Publishing Limited, (2002). Chap. 24
25. Harris, P.J., Altaner, C.M.: Workshop on Commercial Application of IR Spectroscopies to Solid Wood, p. 64. Wood Technology Research Centre, (2013)
26. FOSS Analytics: NIR Technology for Routine Analysis of Food and Agricultural Products (2017). https://www.fossanalytics.com/en/news-articles/technologies/nir-technology Accessed 2020-07-26
27. AB Vista: Understanding the Differences between NIR Machines (2018). https://www.abvista.com/news/June-2018/Differences-between-NIR-machines.aspx Accessed 2020-07-27
28. Thermo Fisher Scientific: FT-NIR Frequently Asked Questions (2016). https://www.thermofisher.com/blog/materials/ft-nir-frequently-asked-questions/ Accessed 2020-07-26
29. Jasco: Principles of Infrared Spectroscopy (1) Molecular Vibrations and Infrared Absorption (2020). https://www.jasco-global.com/principle/principles-of-infrared-spectroscopy-1-molecular-vibrations-and-infrared-absorption/ Accessed 2020-07-30
30. Pasquini, C.: Near Infrared Spectroscopy: Fundamentals, Practical Aspects and Analytical Applications. Journal of the Brazilian Chemical Society **14**(2), 198–219 (2003). doi:10.1590/S0103-50532003000200006
31. Chelladurai, V., Jayas, D.S.: Near-infrared Imaging and Spectroscopy. In: Imaging with Electromagnetic Spectrum: Applications in Food and Agriculture vol. 9783642548, pp. 87–127. Springer, (2014). Chap. 6
32. MathWorks: Principal Component Analysis of Raw Data - MATLAB pca (2020). https://se.mathworks.com/help/stats/pca.html Accessed 2020-07-28
33. Bi, Y., Yuan, K., Xiao, W., Wu, J., Shi, C., Xia, J., Chu, G., Zhang, G., Zhou, G.: A Local Pre-Processing Method for Near-Infrared Spectra, Combined with Spectral Segmentation and Standard Normal Variate Transformation. Analytica Chimica Acta **909**, 30–40 (2016). doi:10.1016/j.aca.2016.01.010
34. Epina ImageLab: Filtering of Spectra (2020). http://www.imagelab.at/help/smoothing.htm Accessed 2020-07-28
35. MathWorks: Partial Least Squares Regression - MATLAB plsregress (2020). https://se.mathworks.com/help/stats/plsregress.html Accessed 2020-07-28
36. MathWorks: Find Outliers in Data - MATLAB isoutlier (2020). https://se.mathworks.com/help/matlab/ref/isoutlier.html{#}bvolfgk Accessed 2020-07-27
37. Kobbi, S., Bougatef, A., Le flem, G., Balti, R., Mickael, C., Fertin, B., Chaabouni, S., Dhulster, P., Nedjar, N.: Purification and Recovery of RuBisCO Protein from Alfalfa Green Juice: Antioxidative Properties of Generated Protein Hydrolysate. Waste and Biomass Valorization **8**(2), 493–504 (2017). doi:10.1007/s12649-016-9589-y
38. Bhatta, S., Janezic, T.S., Ratti, C.: Freeze-Drying of Plant-Based Foods. Foods **9**(1) (2020). doi:10.3390/foods9010087
39. ISO: ISO 8130-1:2019 - Coating powders — Part 1: Determination of Particle Size Distribution by Sieving (2019). https://www.iso.org/standard/68393.html
40. Lochner Labor + Technik GmbH: Maischapparat LP Electronic (2020). http://lochner-europe.de/leistungen/ Accessed 2020-07-27

41. ISO: ISO/TS 16634-2:2009 - Food Products - Determination of the Total Nitrogen Content by Combustion according to the Dumas Principle and Calculation of the Crude Protein Content — Part 2: Cereals, Pulses and Milled Cereal Products (2020). https://www.iso.org/standard/46280.html Accessed 2020-07-27
42. Ulizio, M.: Optical Properties of Glass: How Light and Glass Interact (2015). https://www.koppglass.com/blog/optical-properties-glass-how-light-and-glass-interact Accessed 2020-08-05
43. Ramalho, F.M.G., Simetti, R., Arriel, T.G., Loureiro, B.A., Hein, P.R.G.: Influence of Particles Size on NIR Spectroscopic Estimations of Charcoal Properties. Floresta e Ambiente **26** (2019). doi:10.1590/2179-8087.039718
44. Zarrebini, A., Ghadiri, M., Dyson, M., Kippax, P., McNeil-Watson, F.: Tribo-Electrification of Powders due to Dispersion. Powder Technology **250**, 75–83 (2013). doi:10.1016/j.powtec.2013.10.006

Appendices

A1 - Materials and Methods

Table 9 Particle size distribution as a result of sieving the protein powder. For each press from 1-10, the initially weighed original protein powder, the amounts of respective sieved particle size, the final protein powder as a sum of all determined particle size amounts, and the loss during sieving is presented.

Press		1	2	3	4	5	6	7	8	9	10
Initial protein powder	(g)	40.23	30.01	24.41	17.02	18.35	16.31	15.61	16.17	14.12	12.30
1 mm	(g)	0.95	2.22	1.69	3.23	2.76	3.67	5.33	3.59	2.31	2.30
1 mm	(% w/w)	2.57	7.82	9.49	20.57	16.14	25.88	38.18	25.46	20.25	21.40
0.5 mm	(g)	3.64	3.97	3.28	3.40	3.48	3.36	2.97	3.51	2.70	2.52
0.5 mm	(% w/w)	9.86	13.99	18.43	21.66	20.35	23.70	21.28	24.89	23.66	23.44
0.25 mm	(g)	9.74	9.84	4.37	4.06	4.47	3.21	2.89	3.37	3.15	2.96
0.25 mm	(% w/w)	26.40	34.67	24.55	25.86	26.14	22.64	20.70	23.90	27.61	27.53
0.125 mm	(g)	15.10	8.39	4.26	2.89	3.45	2.11	1.70	2.16	1.93	1.85
0.125 mm	(% w/w)	40.92	29.56	23.93	18.41	20.18	14.88	12.18	15.32	16.91	17.21
$<$ 0.125 mm	(g)	7.47	3.96	4.20	2.12	2.94	1.83	1.07	1.47	1.32	1.12
$<$ 0.125 mm	(% w/w)	20.24	13.95	23.60	13.50	17.19	12.91	7.66	10.43	11.57	10.42
Final protein powder	(g)	36.90	28.38	17.80	15.70	17.10	14.18	13.96	14.10	11.41	10.75
Loss [1]	(g)	3.33	1.63	6.61	1.32	1.25	2.13	1.65	2.07	2.71	1.55
Loss [1]	(% w/w)	8.28	5.43	27.08	7.76	6.81	13.06	10.57	12.80	19.19	12.60

[1] During grinding and sieving of fine particles with a high total surface area leading to a high charge-to-mass ratio, electrostatic charges were observed, which caused adhesion to the walls of the equipment, leading to the relatively high loss of protein powder [44].

Table 10 Green juice and pellet production using the angel juicer. All amounts are presented in (g). The production was made in two batches at two different days. For batch 1 the initial frozen sample weight was 1,198.1 g and the sample weight when thawed was 1,940 g. The leftover pulp after press 10 was 69 g. For batch 2 the initial frozen sample weight was 1,060 g and the sample weight when thawed was 1,749 g. The leftover pulp after press 10 was 51 g. Green juice was conducted in 50 ml samples from batch 2, while protein pellet from pressing 1, 2 and 3 was conducted from batch 1 and the rest of protein pellets were conducted from both batches.

Press	1 [1]	2 [1]	3	4	5	6	7	8	9	10
Pulp	ND	398	136	108	97	81	83	78	71	68
Green juice	1,457	962	274	361	183	160	162	152	135	179
Brown juice	451	512	247	325	164	141	157	137	123	164
Protein pellet	71	31	27	36	19	14	15	13	12	16

Press	1	2	3	4 [1]	5	6	7	8	9	10
Pulp	ND	400	114	82	71	65	58	55	47	46
Green juice	1,158	995	240	294	138	126	122	109	93	130
Brown juice	ND	ND	ND	141	79	71	67	56	42	73
Protein pellet	ND	ND	ND	13.4	8.6	6.7	6.5	5.9	3.9	7.5

[1] The weighed amount of brown juice and protein pellet does not correspond to the amount of green juice, since not all green juice was centrifuged.

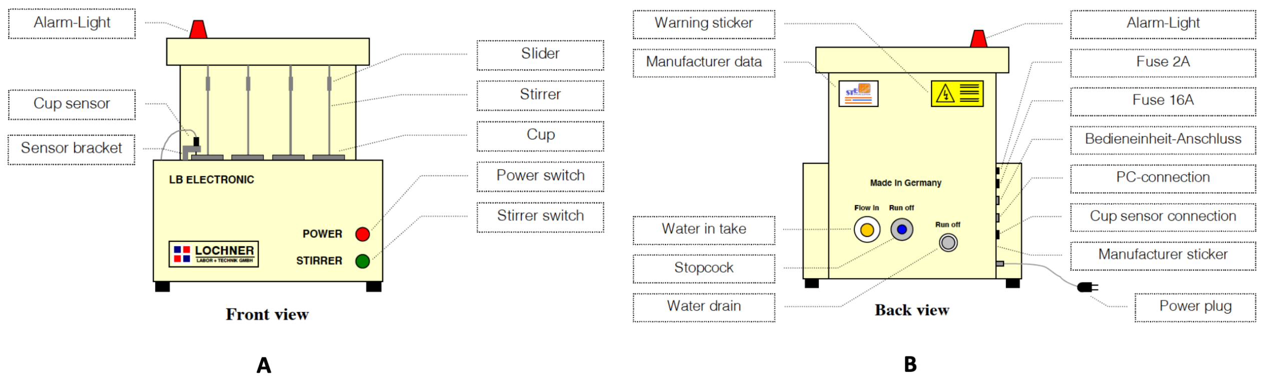
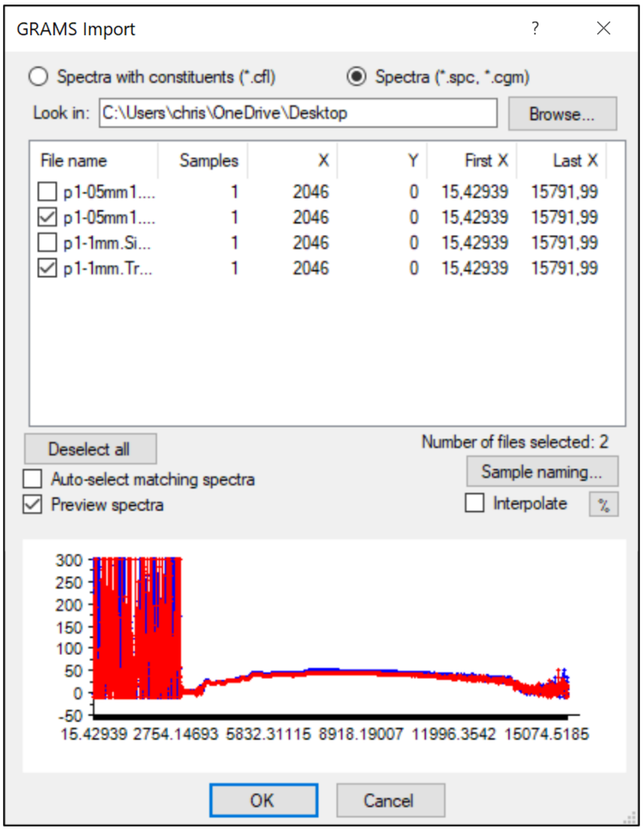


Figure 26 (A) Front and (B) back view of the LB8 mashing bath used as water bath [40].

Table 11 Production of cellulose gluten samples. For each sample mentioned in this table, triplicates has been prepared and analysed.

Dilutions		50_50_C_G	60_40_C_G	70_30_C_G	80_20_C_G	90_10_C_G	100_0_C_G
Cellulose	(g)	5.0241	6.0195	7.0601	8.0443	8.976	10.00
Gluten	(g)	5.0326	3.9978	2.9967	2.0278	0.992	0.000
Cellulose	(% w/w)	49.96	60.09	70.20	79.87	90.05	100.0



GRAMS Import

○ Spectra with constituents (*.cfl) ● Spectra (*.spc, *.cgm)

Look in: C:\Users\chris\OneDrive\Desktop

	File name	Samples	X	Y	First X	Last X
□	p1-05mm1....	1	2046	0	15,42939	15791,99
☑	p1-05mm1....	1	2046	0	15,42939	15791,99
□	p1-1mm.Si...	1	2046	0	15,42939	15791,99
☑	p1-1mm.Tr...	1	2046	0	15,42939	15791,99

Number of files selected: 2

□ Auto-select matching spectra

☑ Preview spectra

□ Interpolate

Figure 27 Determination of NIR wavenumber range using Unscrambler. The wavenumber range is determined to be 15 - 15,792 cm^{-1} described by First X and Last X found by opening the raw spectra spc-files in Unscrambler. A data point spacing of 8 cm^{-1} between all measured data points was used.

A2 - Results

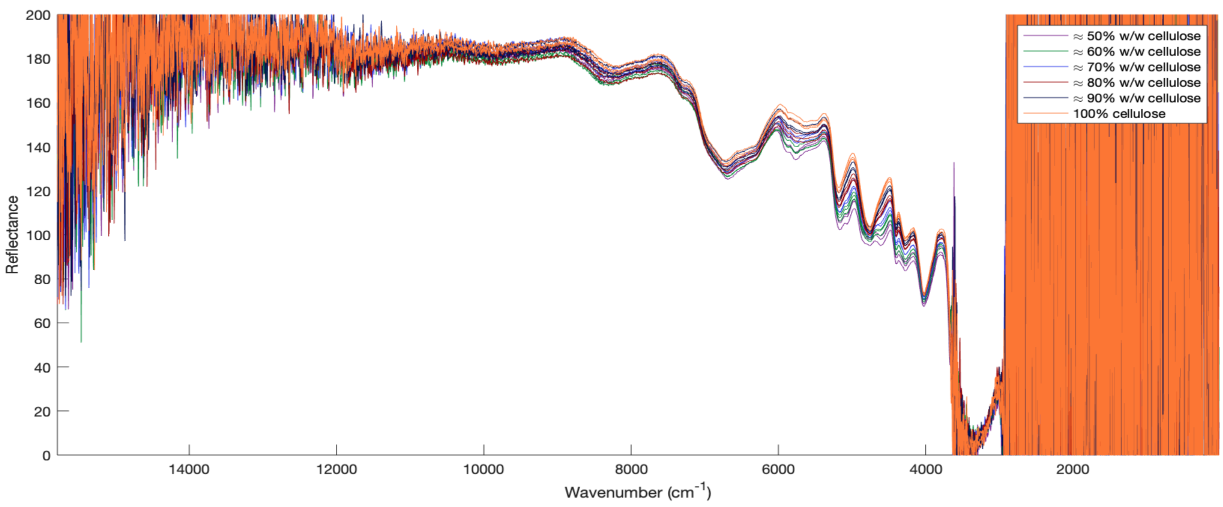


Figure 28 Raw NIR spectra of cellulose gluten powder. Reflectance as a function of wavenumber is presented for all 6 cellulose gluten powder triplicates.

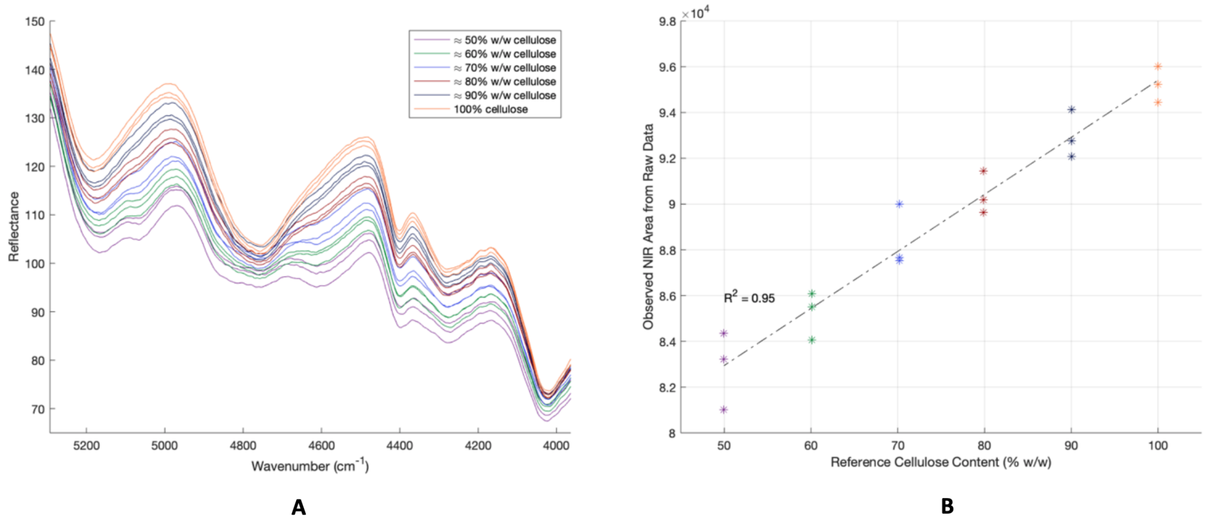


Figure 29 Spectral preprocessing of cellulose gluten powder presented for all 6 cellulose gluten powder triplicates. (A) Zoomed NIR spectra (5,290 - 3,960 cm^{-1}.) with reflectance as a function of wavenumber. (B) correlation between area below each NIR spectrum and reference cellulose content.

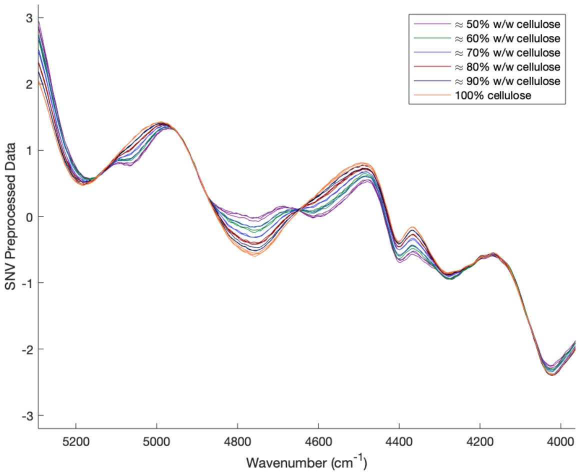


Figure 30 SNV preprocessed zoomed NIR spectra (5,290 - 3,960 cm^{-1}) of cellulose gluten powder. Reflectance as a function of wavenumber is presented for all 6 cellulose gluten powder triplicates.

Table 12 Predicted cellulose contents (% w/w) from the PLS model. For each of the 6 cellulose gluten powders the PLS model has predicted a cellulose content, which is presented related to the actual known reference values.

Dilutions	50_50_C_G	60_40_C_G	70_30_C_G	80_20_C_G	90_10_C_G	100_0_C_G
Predicted cellulose	49.85	60.28	70.10	80.30	89.36	100.28
Reference cellulose	49.96	60.09	70.20	79.87	90.05	100.0

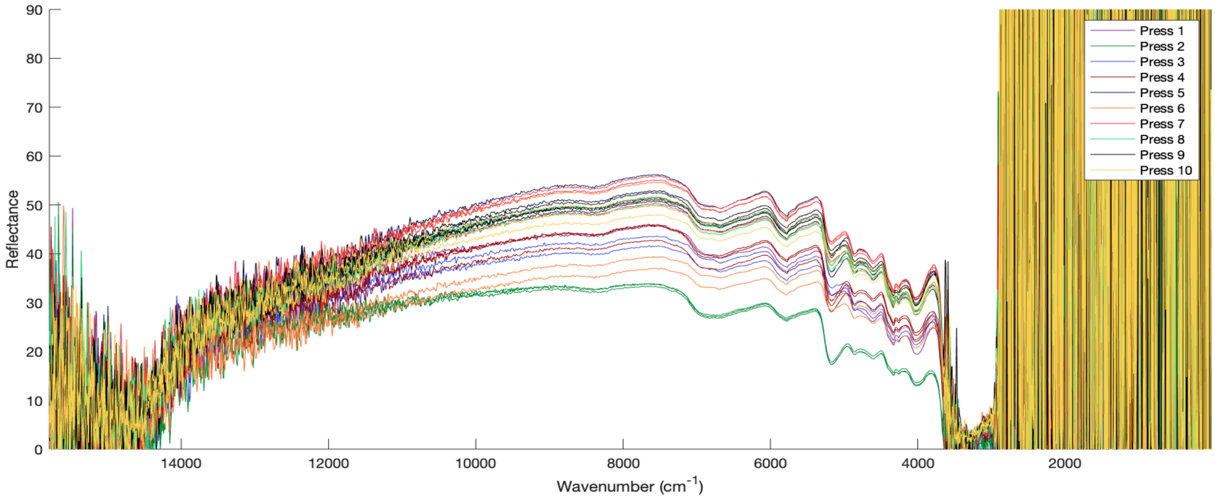


Figure 31 Raw NIR spectra of 1 mm particle size protein powder. Reflectance as a function of wavenumber is presented for all 10 presses in triplicates of rewetted alfalfa pulp, if sufficient powder was available.

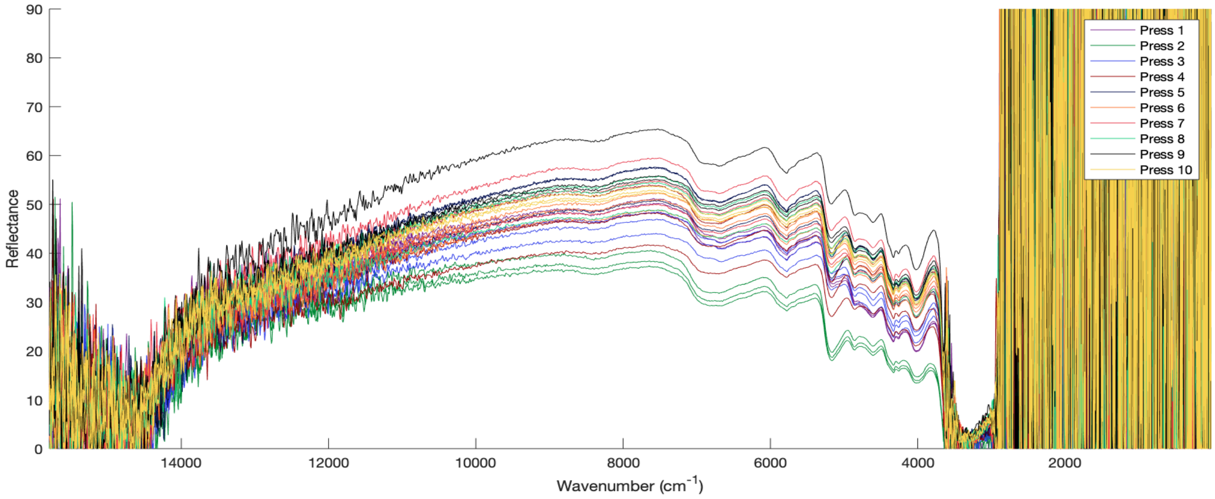


Figure 32 Raw NIR spectra of 0.5 mm particle size protein powder. Reflectance as a function of wavenumber is presented for all 10 presses in triplicates of rewetted alfalfa pulp, if sufficient powder was available.

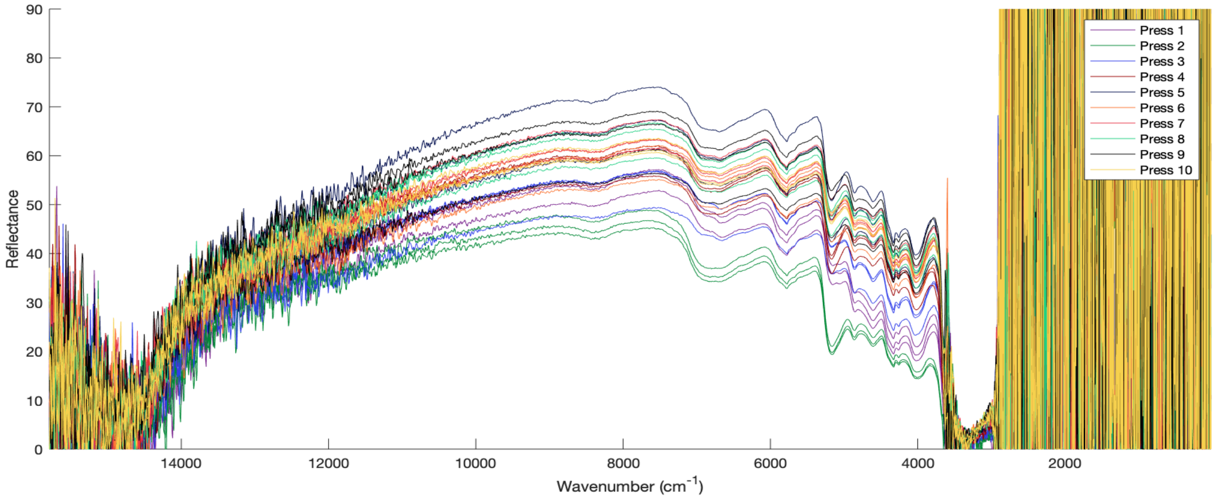


Figure 33 Raw NIR spectra of 0.25 mm particle size protein powder. Reflectance as a function of wavenumber is presented for all 10 presses in triplicates of rewetted alfalfa pulp, if sufficient powder was available.

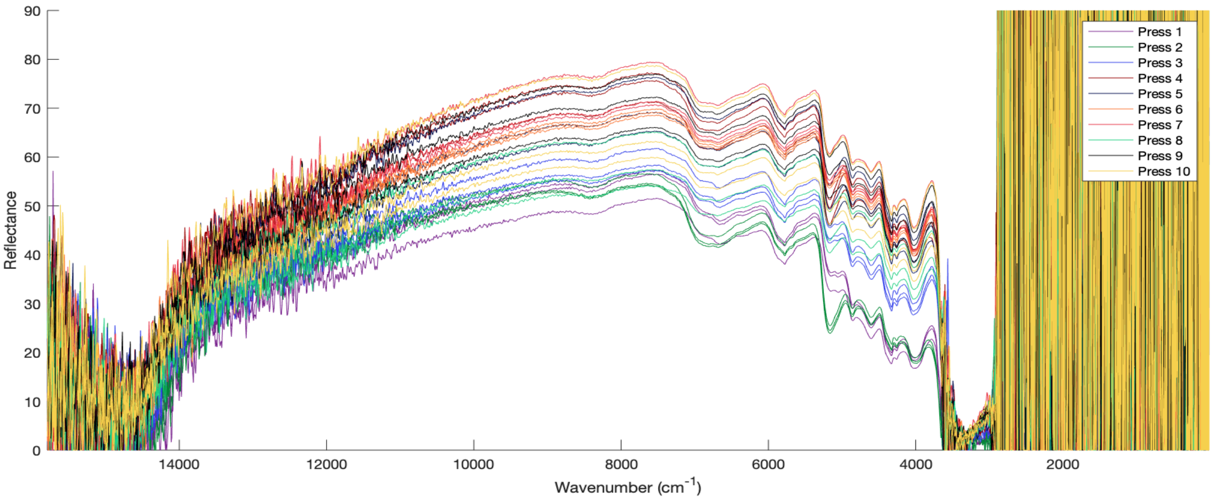


Figure 34 Raw NIR spectra of 0.125 mm particle size protein powder. Reflectance as a function of wavenumber is presented for all 10 presses in triplicates of rewetted alfalfa pulp, if sufficient powder was available.

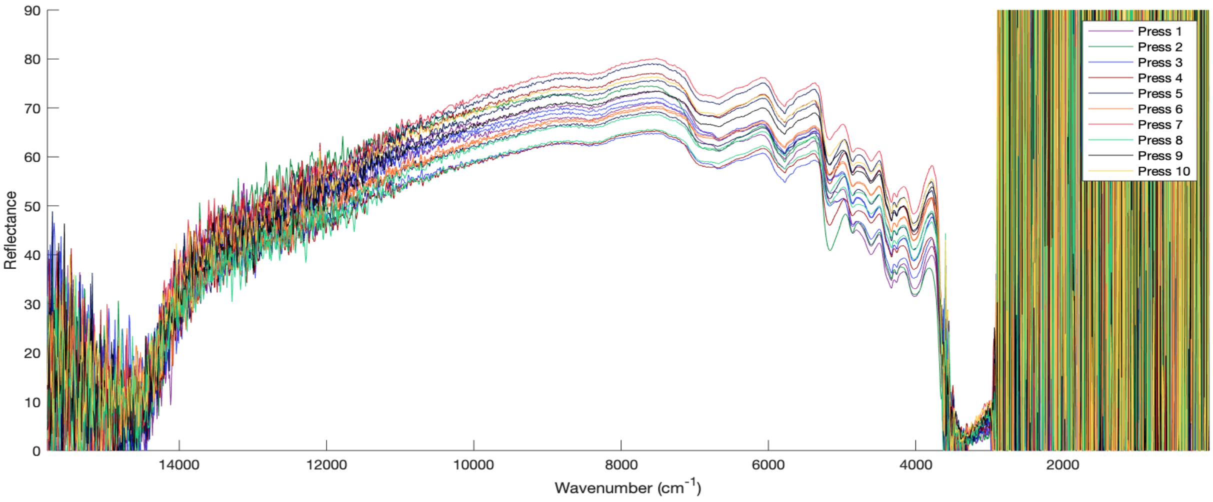


Figure 35 Raw NIR spectra of $<$ 0.125 mm particle size protein powder. Reflectance as a function of wavenumber is presented for all 10 presses in triplicates of rewetted alfalfa pulp, if sufficient powder was available.

Table 13 Determined locations of clear spectral peaks (cm^{-1}) for respective protein powder particle size. Peak numbers are correlated to the spectral occurrence, at which wavenumber they are detected.

Peak Number	1	2	3	4	5	6	7
1 mm	6079	5369	4984	4791	4498	4297	4158
0.5 mm	6079	5369	4984	4791	4498	4297	4166
0.25 mm	6079	5369	4976	4791	4498	4297	4166
0.125 mm	6079	5369	4976	4783	4498	4297	4166
$<$ 0.125 mm	6079	5369	4976	4791	4498	4289	4166

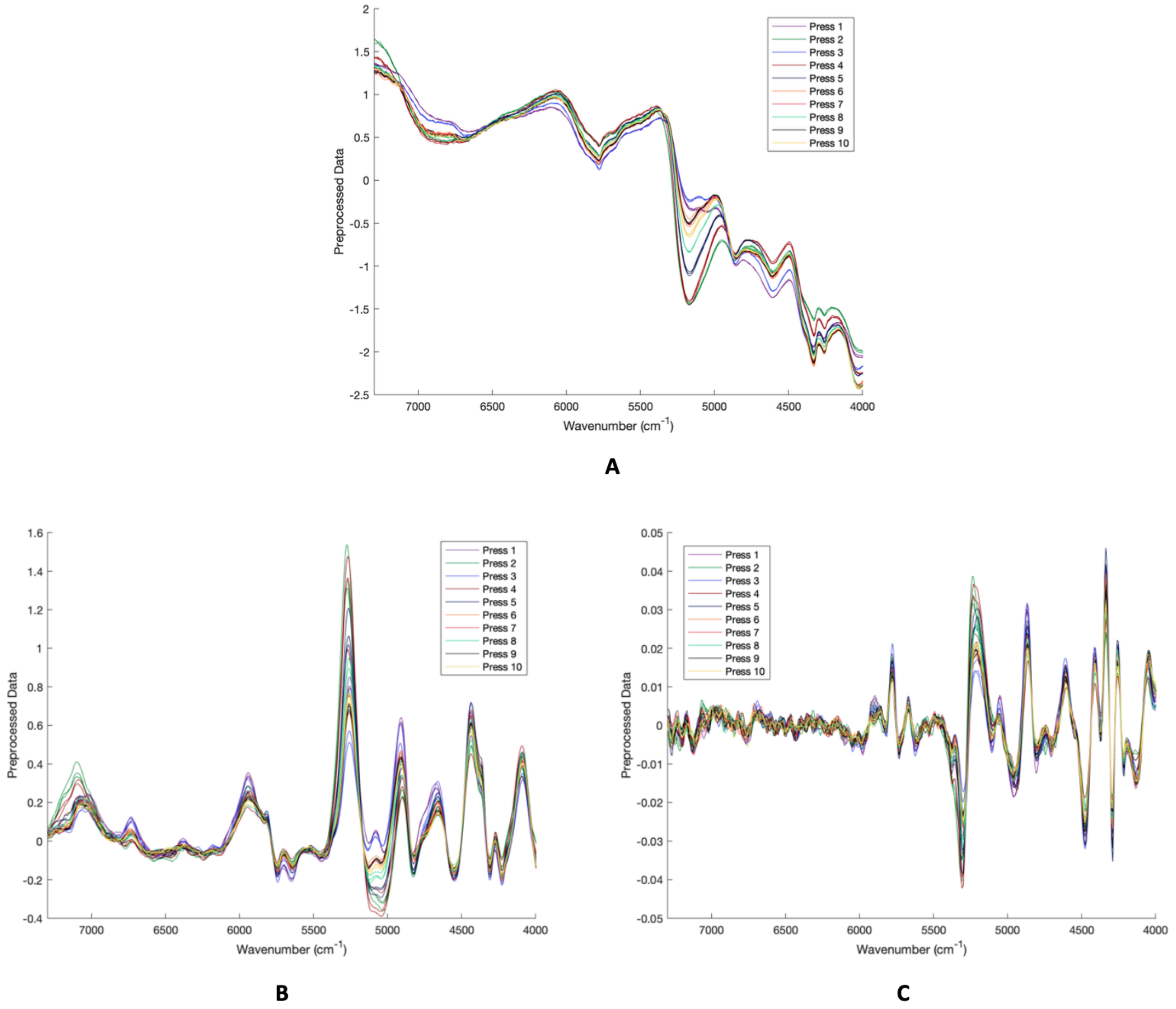


Figure 36 Preprocessing of zoomed NIR spectra (7,300 - 4,000 cm^{-1}) with reflectance as a function of wavenumber of original protein powder presented for all 10 presses in triplicates of rewetted alfalfa pulp. Original particle size refers to freeze dried protein powder milled down to 1 mm. (A) SNV preprocessing. (B) 1st derivative preprocessing. (C) 2nd derivative preprocessing.

A3 - MATLAB Cellulose Gluten Spectra

Own MATLAB code used for data handling of the cellulose gluten samples:

Main.m

```matlab
%% Cellulose gluten test sample data handling
clear all;
close all;
clc

% Loads raw data from Excel
run HarnessRaw.m

% Plots raw NIR data
figure, hold on
p501 = plot (wavelength, a(1,:), 'Color',[121/255, 35/255, 142/255]);
p502 = plot (wavelength, a(2,:), 'Color',[121/255, 35/255, 142/255]);
p503 = plot (wavelength, a(3,:), 'Color',[121/255, 35/255, 142/255]);
p601 = plot (wavelength, a(4,:), 'Color',[0/255, 136/255, 53/255]);
p602 = plot (wavelength, a(5,:), 'Color',[0/255, 136/255, 53/255]);
p603 = plot (wavelength, a(6,:), 'Color',[0/255, 136/255, 53/255]);
p701 = plot (wavelength, a(7,:), 'Color',[47/255, 62/255, 234/255]);
p702 = plot (wavelength, a(8,:), 'Color',[47/255, 62/255, 234/255]);
p703 = plot (wavelength, a(9,:), 'Color',[47/255, 62/255, 234/255]);
p801 = plot (wavelength, a(10,:), 'Color',[153/255, 0/255, 0/255]);
p802 = plot (wavelength, a(11,:), 'Color',[153/255, 0/255, 0/255]);
p803 = plot (wavelength, a(12,:), 'Color',[153/255, 0/255, 0/255]);
p901 = plot (wavelength, a(13,:), 'Color',[3/255, 15/255, 79/255]);
p902 = plot (wavelength, a(14,:), 'Color',[3/255, 15/255, 79/255]);
p903 = plot (wavelength, a(15,:), 'Color',[3/255, 15/255, 79/255]);
p1001 = plot (wavelength, a(16,:), 'Color',[252/255, 118/255, 52/255]);
p1002 = plot (wavelength, a(17,:), 'Color',[252/255, 118/255, 52/255]);
p1003 = plot (wavelength, a(18,:), 'Color',[252/255, 118/255, 52/255]);
xlabel('Wavenumber (cm^{-1})'), ylabel('Reflectance')
legend([p501 p601 p701 p801 p901 p1001],...
    {'\approx 50% w/w cellulose','\approx 60% w/w cellulose',...
    '\approx 70% w/w cellulose','\approx 80% w/w cellulose',...
    '\approx 90% w/w cellulose','100% cellulose'})
ylim([0 200])
xlim([min(wavelength) max(wavelength)])
x0=10;
y0=10;
width=1000;
height=400;
set(gcf,'units','points','position',[x0,y0,width,height])
set(gca,'xdir','reverse')

% Loads data for a specific wavenumber range and reference values
run HarnesData.m

% Plots zoomed NIR data
figure, hold on
p501 = plot (wavelength, a(1,:), 'Color',[121/255, 35/255, 142/255]);
p502 = plot (wavelength, a(2,:), 'Color',[121/255, 35/255, 142/255]);
p503 = plot (wavelength, a(3,:), 'Color',[121/255, 35/255, 142/255]);
p601 = plot (wavelength, a(4,:), 'Color',[0/255, 136/255, 53/255]);
p602 = plot (wavelength, a(5,:), 'Color',[0/255, 136/255, 53/255]);
p603 = plot (wavelength, a(6,:), 'Color',[0/255, 136/255, 53/255]);
p701 = plot (wavelength, a(7,:), 'Color',[47/255, 62/255, 234/255]);
p702 = plot (wavelength, a(8,:), 'Color',[47/255, 62/255, 234/255]);
p703 = plot (wavelength, a(9,:), 'Color',[47/255, 62/255, 234/255]);
p801 = plot (wavelength, a(10,:), 'Color',[153/255, 0/255, 0/255]);
p802 = plot (wavelength, a(11,:), 'Color',[153/255, 0/255, 0/255]);
p803 = plot (wavelength, a(12,:), 'Color',[153/255, 0/255, 0/255]);
p901 = plot (wavelength, a(13,:), 'Color',[3/255, 15/255, 79/255]);
p902 = plot (wavelength, a(14,:), 'Color',[3/255, 15/255, 79/255]);
p903 = plot (wavelength, a(15,:), 'Color',[3/255, 15/255, 79/255]);
p1001 = plot (wavelength, a(16,:), 'Color',[252/255, 118/255, 52/255]);
p1002 = plot (wavelength, a(17,:), 'Color',[252/255, 118/255, 52/255]);
p1003 = plot (wavelength, a(18,:), 'Color',[252/255, 118/255, 52/255]);
xlabel('Wavenumber (cm^{-1})'), ylabel('Reflectance')
legend([p501 p601 p701 p801 p901 p1001],...
    {'\approx 50% w/w cellulose','\approx 60% w/w cellulose',...
    '\approx 70% w/w cellulose','\approx 80% w/w cellulose',...
    '\approx 90% w/w cellulose','100% cellulose'})
ylim([65 150])
xlim([3964 5293])
set(gca,'xdir','reverse')

```

```
% Calculates and plots areas under data curves
Xa=wavelength;
Ya=a';
run NIRarea.m

%% Pretreatment PCA
% Loads preprocessings
run Preprocessing.m

% Plots SNV preprocessed spectra
figure, hold on
s501 = plot (wavelength, Xsnv18(1,:), 'Color',[121/255, 35/255, 142/255]);
s502 = plot (wavelength, Xsnv18(2,:), 'Color',[121/255, 35/255, 142/255]);
s503 = plot (wavelength, Xsnv18(3,:), 'Color',[121/255, 35/255, 142/255]);
s601 = plot (wavelength, Xsnv18(4,:), 'Color',[0/255, 136/255, 53/255]);
s602 = plot (wavelength, Xsnv18(5,:), 'Color',[0/255, 136/255, 53/255]);
s603 = plot (wavelength, Xsnv18(6,:), 'Color',[0/255, 136/255, 53/255]);
s701 = plot (wavelength, Xsnv18(7,:), 'Color',[47/255, 62/255, 234/255]);
s702 = plot (wavelength, Xsnv18(8,:), 'Color',[47/255, 62/255, 234/255]);
s703 = plot (wavelength, Xsnv18(9,:), 'Color',[47/255, 62/255, 234/255]);
s801 = plot (wavelength, Xsnv18(10,:), 'Color',[153/255, 0/255, 0/255]);
s802 = plot (wavelength, Xsnv18(11,:), 'Color',[153/255, 0/255, 0/255]);
s803 = plot (wavelength, Xsnv18(12,:), 'Color',[153/255, 0/255, 0/255]);
s901 = plot (wavelength, Xsnv18(13,:), 'Color',[3/255, 15/255, 79/255]);
s902 = plot (wavelength, Xsnv18(14,:), 'Color',[3/255, 15/255, 79/255]);
s903 = plot (wavelength, Xsnv18(15,:), 'Color',[3/255, 15/255, 79/255]);
s1001 = plot (wavelength, Xsnv18(16,:), 'Color',[252/255, 118/255, 52/255]);
s1002 = plot (wavelength, Xsnv18(17,:), 'Color',[252/255, 118/255, 52/255]);
s1003 = plot (wavelength, Xsnv18(18,:), 'Color',[252/255, 118/255, 52/255]);
xlabel('Wavenumber (cm^{-1})'), ylabel('SNV Preprocessed Data')
legend([s501 s601 s701 s801 s901 s1001],...
    {'\approx 50% w/w cellulose','\approx 60% w/w cellulose',...
     '\approx 70% w/w cellulose','\approx 80% w/w cellulose',...
     '\approx 90% w/w cellulose','100% cellulose'})
xlim([3964 5293])
ylim([-3.2 3.2])
set(gca,'xdir','reverse')

% Shows PCA plot with reference values vs PC1
a = Xsnv18;
lookatrefpc1 = true;
run PCAmodel.m

% Calculates mean spectra from preprocessed spectra
run MeanSpectra.m
X(1,:) = []; % X=mean spectra matrix
[Xsnv6]=snv(X);
a = Xsnv6;

%% PLS
X = Xsnv6;
Y = refmean;
run PLSmodel.m

% Presents PLS fitted response vs reference values
PLSResult = [yfitPLS Y]
```

HarnessRaw.m

```
%% Loads data file
X_AL = xlsread('Spectra.xlsx','C_G_all','B3:T4096');
X_ALL = X_AL.';

[t,r]=size(X_ALL);
% t is the number of samples, r is the number of variables

% Array created for wavenumbers
wavelength = X_ALL(1,1:r);

% Array created for dataset
a=X_ALL(2:t,1:r);

% Initial preprocessing - negative values in import to zero
a(a<0)=0;
```

HarnessData.m

```
%% Loads data file
X_AL = xlsread('Spectra.xlsx','C_G_all','B3:T4096');
X_ALL = X_AL.';

[t,r]=size(X_ALL);
% t is the number of samples, r is the number of variables

% Choose wavenumber range
chooselow = 5288;
choosehigh = 3959;

% Array created for wavenumber range
wavelength1=X_ALL(1,2:r);
low = mink(find(abs(wavelength1-chooselow) < 2),1);
high = mink(find(abs(wavelength1-choosehigh) < 2),1);
wavelength = X_ALL(1,low:high);

% Array created for corresponding dataset
a=X_ALL(2:t,low:high);

% Initial preprocessing - negative values in import to zero
a(a<0)=0;

% Harness all reference values
ref = xlsread('Spectra.xlsx','C_G_all','C2:T2').';
refmean=[];
i=1;
while i<(t-2)
    refmean=[refmean,ref(i+1,1)];
    i=i+3;
end
refmean = refmean';
```

NIRarea.m

```
%% Calculates areas under raw data curves
areas = trapz(fliplr(wavelength), fliplr(Ya));

% Plots correlation between area and reference values
% with a linear regression line
[h,g]=size(Ya);
figure, hold on
for i=1:g
if g == 18
 C = {[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],...
     [0/255, 136/255, 53/255],[0/255, 136/255, 53/255],[0/255, 136/255, 53/255],...
     [47/255, 62/255, 234/255],[47/255, 62/255, 234/255],[47/255, 62/255, 234/255],...
     [153/255, 0/255, 0/255],[153/255, 0/255, 0/255],[153/255, 0/255, 0/255],...
     [3/255, 15/255, 79/255],[3/255, 15/255, 79/255],[3/255, 15/255, 79/255],...
     [252/255, 118/255, 52/255],[252/255, 118/255, 52/255],[252/255, 118/255, 52/255]};
 elseif g == 6
 C = {[121/255, 35/255, 142/255],[0/255, 136/255, 53/255],[47/255, 62/255, 234/255],...
     [153/255, 0/255, 0/255],[3/255, 15/255, 79/255],[252/255, 118/255, 52/255]};
end
    plot(ref(i,1),areas(1,g+1-i),'*','Color',C{i});
end
[P18,S18] = polyfit(ref,flip(areas'),1);
yfit18 = P18(1)*ref+P18(2);  % P(1)=slope, P(2)=intercept
hold on
plot(ref,yfit18,'k-.')
xlim([45 105])
xlabel('Reference Cellulose Content (% w/w)')
ylabel('Observed NIR Area from Raw Data')
grid on
Rsquarea18 = 1 - (S18.normr/norm(areas - mean(areas)))^2
text(50, 86000 , ['R^2 = 0.95'])

% Plots mean NIR area vs reference values with a linear regression line
Y = refmean;
figure, hold on
for i=1:6
    C = {[121/255, 35/255, 142/255],[0/255, 136/255, 53/255],[47/255, 62/255, 234/255],...
         [153/255, 0/255, 0/255],[3/255, 15/255, 79/255],[252/255, 118/255, 52/255]};
    amean = [...
```

```
        (areas(1,1)+areas(1,2)+areas(1,3))./3,...
        (areas(1,4)+areas(1,5)+areas(1,6))./3,...
        (areas(1,7)+areas(1,8)+areas(1,9))./3,...
        (areas(1,10)+areas(1,11)+areas(1,12))./3,...
        (areas(1,13)+areas(1,14)+areas(1,15))./3,...
        (areas(1,16)+areas(1,17)+areas(1,18))./3];
    amean = amean.';
    plot(Y(i,1),amean(7-i,1),'*','Color',C{i});
end
[P,S] = polyfit(Y,flip(amean),1);
slope = P(1);
intercept = P(2);
yfit = P(1)*Y+P(2);  % P(1)=slope, P(2)=intercept
hold on
plot(Y,yfit,'k-.')
xlim([45 105])
xlabel('Reference Values')
ylabel('Observed Response from Raw Data')
grid on
Rsqarea = 1 - (S.normr/norm(amean - mean(amean)))^2
text(50, 86000 , ['R^2 = 0.9975'])
```

Preprocessing.m

```
% SNV (Standard Normal Variate transformation)
[Xsnv18]=snv(a);

% MSC (Multiplicative Scatter Correction)
[xmsc18]=msc(a,1,size(a,2));

% S/G 1st der (Savitzky-Golay 1st derivative)
[Xde118]=deriv(a,1,11,2);

% S/G 2nd der (Savitzky-Golay 2nd derivative)
[Xde218]=deriv(a,2,25,2);

% S/G 2nd der (Savitzky-Golay 2nd derivative) incl. MSC
[Xde2msc18]=deriv(xmsc18,2,25,2);

% S/G 2nd der (Savitzky-Golay 2nd derivative) incl. SNV
[Xde2snv18]=deriv(Xsnv18,2,25,2);
```

PCAmodel.m

```
% PCA options
lookatPCs = false;
lookatrefpc2 = false;
lookatloading = false;
lookatareapc1 = false;

%% Plots PC1 vs PC2
if lookatPCs == true
    [coeff,score,latent,tsquared,explained] = pca(a);
    [g,h]=size(a);
    figure, hold on
    for i=1:g
        if g == 18
            C = {[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],...
                [0/255, 136/255, 53/255],[0/255, 136/255, 53/255],[0/255, 136/255, 53/255],...
                [47/255, 62/255, 234/255],[47/255, 62/255, 234/255],[47/255, 62/255, 234/255],...
                [153/255, 0/255, 0/255],[153/255, 0/255, 0/255],[153/255, 0/255, 0/255],...
                [3/255, 15/255, 79/255],[3/255, 15/255, 79/255],[3/255, 15/255, 79/255],...
                [252/255, 118/255, 52/255],[252/255, 118/255, 52/255],[252/255, 118/255, 52/255]};
            T = {'  50','  50','  50','  60','  60','  60',...
                  '  70','  70','  70','  80','  80','  80',...
                  '  90','  90','  90','  100','  100','  100'};
        elseif g == 6
            C = {[121/255, 35/255, 142/255],[0/255, 136/255, 53/255],[47/255, 62/255, 234/255],...
                [153/255, 0/255, 0/255],[3/255, 15/255, 79/255],[252/255, 118/255, 52/255]};
            T = {'  50','  60','  70','  80','  90','  100'};
        end
        scatter(score(i,1),score(i,2),75,'*','MarkerFaceColor',C{i});
        text(score(i,1),score(i,2),T{i})
    end
    xline(0,':k');
    yline(0,':k');
    expraw  = explained;
    xlabel('PC1'), ylabel('PC2')

    if isequal(a,Xsnv18)
        expsnv = explained;
    end

    if (exist ('X') == true)
        if isequal(a,X)
        expmean = explained;
        end

        figure, hold on
        for i=1:length(explained)+1
            G = cumsum(latent/sum(latent));
            G = [0;G];
            plot(i-1,G(i,1),'-bo')
            title('Explained Variance')
            xlabel('Number of PCs')
            ylabel('Percent Variance Explained in X')
        end
    end


%% Plots Ref vs PC1
elseif lookatrefpc1 == true
    [coeff,score,latent,tsquared,explained] = pca(a);
    [g,h]=size(a);
    figure, hold on
    for i=1:g
        if g == 18
            C = {[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],...
                 [0/255, 136/255, 53/255],[0/255, 136/255, 53/255],[0/255, 136/255, 53/255],...
                 [47/255, 62/255, 234/255],[47/255, 62/255, 234/255],[47/255, 62/255, 234/255],...
                 [153/255, 0/255, 0/255],[153/255, 0/255, 0/255],[153/255, 0/255, 0/255],...
                 [3/255, 15/255, 79/255],[3/255, 15/255, 79/255],[3/255, 15/255, 79/255],...
                 [252/255, 118/255, 52/255],[252/255, 118/255, 52/255],[252/255, 118/255, 52/255]};
            plot(ref(i,1),score(i,1),'*','Color',C{i});
        elseif g == 6
            C = {[121/255, 35/255, 142/255],[0/255, 136/255, 53/255],[47/255, 62/255, 234/255],...
                 [153/255, 0/255, 0/255],[3/255, 15/255, 79/255],[252/255, 118/255, 52/255]};
            plot(refmean(i,1),score(i,1),'*','Color',C{i});
        end
    end
    xlabel('Reference Cellulose Content (% w/w)'), ylabel('PC1')
    if isequal(a,Xsnv18)
        ylabel('PC1'), xlabel('Reference Cellulose Content (% w/w)')
        xlim([45 105])
        ylim([-3.5 3.5])
        grid on
    end
    if (exist ('X') == true)
        if isequal(a,X)
        ylabel('PC1'), xlabel('Reference Cellulose Content (% w/w)')
        xlim([45 105])
        grid on
        end
    end


%% Plots Ref vs PC2
elseif lookatrefpc2 == true
    [coeff,score,latent,tsquared,explained] = pca(a);
    [g,h]=size(a);
    figure, hold on
    for i=1:g
        if g == 18
            C = {[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],...
                 [0/255, 136/255, 53/255],[0/255, 136/255, 53/255],[0/255, 136/255, 53/255],...
                 [47/255, 62/255, 234/255],[47/255, 62/255, 234/255],[47/255, 62/255, 234/255],...
                 [153/255, 0/255, 0/255],[153/255, 0/255, 0/255],[153/255, 0/255, 0/255],...
                 [3/255, 15/255, 79/255],[3/255, 15/255, 79/255],[3/255, 15/255, 79/255],...
                 [252/255, 118/255, 52/255],[252/255, 118/255, 52/255],[252/255, 118/255, 52/255]};
            T = {'   50','   50','   50','   60','   60','   60',...
                        '   70','   70','   70','   80','   80','   80',...
                        '   90','   90','   90','   100','   100','   100'};
            scatter(score(i,2),ref(i,1),75,'*','MarkerFaceColor',C{i});
            text(score(i,2),ref(i,1),T{i})
        elseif g == 6
            C = {[121/255, 35/255, 142/255],[0/255, 136/255, 53/255],[47/255, 62/255, 234/255],...
                 [153/255, 0/255, 0/255],[3/255, 15/255, 79/255],[252/255, 118/255, 52/255]};
            T = {'   50','   60','   70','   80','   90','   100'};
            scatter(score(i,2),refmean(i,1),75,'*','MarkerFaceColor',C{i});
            text(score(i,2),refmean(i,1),T{i})
        end
    end
    xlabel('PC2'), ylabel('Reference %Cellulose'), title('Raw Data PCA')
    if isequal(a,Xsnv18)
        xlabel('PC2'), ylabel('Reference %Cellulose'), title('SNV Preprocessed Data PCA')
    end
    if (exist ('X') == true)
        if isequal(a,X)
           xlabel('PC2'), ylabel('Reference %Cellulose'), title('Mean Spectra from SNV Data PCA')
        end
    end


%% Plots loadings (coeff1) vs Wavenumber
elseif lookatloading == true
    [coeff,score,latent,tsquared,explained] = pca(a);
    [g,h]=size(a);
    wavelength = wavelength';
    figure, hold on
    plot(wavelength,coeff(:,1));
    xlabel('Wavenumber [cm-1]'), ylabel('Loading 1'), title('Raw Data PCA')
    if isequal(a,Xsnv18)
        xlabel('Wavenumber [cm-1]'), ylabel('Loading 1'), title('SNV Preprocessed Data PCA')
    end
    if (exist ('X') == true)
        if isequal(a,X)
        xlabel('Wavenumber [cm-1]'), ylabel('Loading 1'), title('Mean Spectra from SNV Data PCA')
        end
    end


%% Plots area vs PC1
elseif lookatareapc1 == true
    [coeff,score,latent,tsquared,explained] = pca(a);
    [g,h]=size(a);
    figure, hold on
    for i=1:g
        if g == 18
            C = {[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],...
                 [0/255, 136/255, 53/255],[0/255, 136/255, 53/255],[0/255, 136/255, 53/255],...
                 [47/255, 62/255, 234/255],[47/255, 62/255, 234/255],[47/255, 62/255, 234/255],...
                 [153/255, 0/255, 0/255],[153/255, 0/255, 0/255],[153/255, 0/255, 0/255],...
                 [3/255, 15/255, 79/255],[3/255, 15/255, 79/255],[3/255, 15/255, 79/255],...
                 [252/255, 118/255, 52/255],[252/255, 118/255, 52/255],[252/255, 118/255, 52/255]};
            T = {'   50','   50','   50','   60','   60','   60',...
                        '   70','   70','   70','   80','   80','   80',...
                        '   90','   90','   90','   100','   100','   100'};
            scatter(score(i,1),areas(1,i),75,'*','MarkerFaceColor',C{i});
            text(score(i,1),areas(1,i),T{i})
        elseif g == 6
            C = {[121/255, 35/255, 142/255],[0/255, 136/255, 53/255],[47/255, 62/255, 234/255],...
                 [153/255, 0/255, 0/255],[3/255, 15/255, 79/255],[252/255, 118/255, 52/255]};
            T = {'   50','   60','   70','   80','   90','   100'};
            scatter(score(i,1),amean(1,i),75,'*','MarkerFaceColor',C{i});
            text(score(i,1),amean(1,i),T{i})
        end
    end
    xlabel('PC1'), ylabel('Area'), title('Raw Data PCA')
    if (exist ('Xsnv6') == true)
        if isequal(a,Xsnv6)
            xlabel('PC1'), ylabel('Area'), title('Mean Raw Data PCA')
        end
    end

else
% do nothing
end

wavelength = wavelength;
```

PCAmodel.m

```
% PCA options
lookatPCs = false;
lookatrefpc2 = false;
lookatloading = false;
lookatareapc1 = false;

%% Plots PC1 vs PC2
if lookatPCs == true
    [coeff,score,latent,tsquared,explained] = pca(a);
    [g,h]=size(a);
    figure, hold on
    for i=1:g
        if g == 18
            C = {[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],...
                [0/255, 136/255, 53/255],[0/255, 136/255, 53/255],[0/255, 136/255, 53/255],...
                [47/255, 62/255, 234/255],[47/255, 62/255, 234/255],[47/255, 62/255, 234/255],...
                [153/255, 0/255, 0/255],[153/255, 0/255, 0/255],[153/255, 0/255, 0/255],...
                [3/255, 15/255, 79/255],[3/255, 15/255, 79/255],[3/255, 15/255, 79/255],...
                [252/255, 118/255, 52/255],[252/255, 118/255, 52/255],[252/255, 118/255, 52/255]};
            T = {'   50','   50','   50','   60','   60','   60',...
                 '   70','   70','   70','   80','   80','   80',...
                 '   90','   90','   90','   100','   100','   100'};
        elseif g == 6
            C = {[121/255, 35/255, 142/255],[0/255, 136/255, 53/255],[47/255, 62/255, 234/255],...
                [153/255, 0/255, 0/255],[3/255, 15/255, 79/255],[252/255, 118/255, 52/255]};
            T = {'   50','   60','   70','   80','   90','   100'};
        end
        scatter(score(i,1),score(i,2),75,'*','MarkerFaceColor',C{i});
        text(score(i,1),score(i,2),T{i})
    end
    xline(0,':k');
    yline(0,':k');
    expraw  = explained;
    xlabel('PC1'), ylabel('PC2')

    if isequal(a,Xsnv18)
        expsnv = explained;
    end

    if (exist ('X') == true)
        if isequal(a,X)
        expmean = explained;
        end

        figure, hold on
        for i=1:length(explained)+1
            G = cumsum(latent/sum(latent));
            G = [0;G];
            plot(i-1,G(i,1),'-bo')
            title('Explained Variance')
            xlabel('Number of PCs')
            ylabel('Percent Variance Explained in X')
        end
    end


%% Plots Ref vs PC1
elseif lookatrefpc1 == true
    [coeff,score,latent,tsquared,explained] = pca(a);
    [g,h]=size(a);
    figure, hold on
    for i=1:g
        if g == 18
            C = {[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],...
                [0/255, 136/255, 53/255],[0/255, 136/255, 53/255],[0/255, 136/255, 53/255],...
                [47/255, 62/255, 234/255],[47/255, 62/255, 234/255],[47/255, 62/255, 234/255],...
                [153/255, 0/255, 0/255],[153/255, 0/255, 0/255],[153/255, 0/255, 0/255],...
                [3/255, 15/255, 79/255],[3/255, 15/255, 79/255],[3/255, 15/255, 79/255],...
                [252/255, 118/255, 52/255],[252/255, 118/255, 52/255],[252/255, 118/255, 52/255]};
            plot(ref(i,1),score(i,1),'*','Color',C{i});
        elseif g == 6
            C = {[121/255, 35/255, 142/255],[0/255, 136/255, 53/255],[47/255, 62/255, 234/255],...
                [153/255, 0/255, 0/255],[3/255, 15/255, 79/255],[252/255, 118/255, 52/255]};
            plot(refmean(i,1),score(i,1),'*','Color',C{i});
        end
    end
    xlabel('Reference Cellulose Content (% w/w)'), ylabel('PC1')
    if isequal(a,Xsnv18)
```

```
        ylabel('PC1'), xlabel('Reference Cellulose Content (% w/w)')
        xlim([45 105])
        ylim([-3.5 3.5])
        grid on
    end
    if (exist ('X') == true)
        if isequal(a,X)
        ylabel('PC1'), xlabel('Reference Cellulose Content (% w/w)')
        xlim([45 105])
        grid on
        end
    end


%% Plots Ref vs PC2
elseif lookatrefpc2 == true
    [coeff,score,latent,tsquared,explained] = pca(a);
    [g,h]=size(a);
    figure, hold on
    for i=1:g
        if g == 18
            C = {[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],...
                 [0/255, 136/255, 53/255],[0/255, 136/255, 53/255],[0/255, 136/255, 53/255],...
                 [47/255, 62/255, 234/255],[47/255, 62/255, 234/255],[47/255, 62/255, 234/255],...
                 [153/255, 0/255, 0/255],[153/255, 0/255, 0/255],[153/255, 0/255, 0/255],...
                 [3/255, 15/255, 79/255],[3/255, 15/255, 79/255],[3/255, 15/255, 79/255],...
                 [252/255, 118/255, 52/255],[252/255, 118/255, 52/255],[252/255, 118/255, 52/255]};
            T = {'   50','   50','   50','   60','   60','   60',...
                        '   70','   70','   70','   80','   80','   80',...
                        '   90','   90','   90','   100','   100','   100'};
            scatter(score(i,2),ref(i,1),75,'*','MarkerFaceColor',C{i});
            text(score(i,2),ref(i,1),T{i})
        elseif g == 6
            C = {[121/255, 35/255, 142/255],[0/255, 136/255, 53/255],[47/255, 62/255, 234/255],...
                 [153/255, 0/255, 0/255],[3/255, 15/255, 79/255],[252/255, 118/255, 52/255]};
            T = {'   50','   60','   70','   80','   90','   100'};
            scatter(score(i,2),refmean(i,1),75,'*','MarkerFaceColor',C{i});
            text(score(i,2),refmean(i,1),T{i})
        end
    end
    xlabel('PC2'), ylabel('Reference %Cellulose'), title('Raw Data PCA')
    if isequal(a,Xsnv18)
        xlabel('PC2'), ylabel('Reference %Cellulose'), title('SNV Preprocessed Data PCA')
    end
    if (exist ('X') == true)
        if isequal(a,X)
           xlabel('PC2'), ylabel('Reference %Cellulose'), title('Mean Spectra from SNV Data PCA')
        end
    end


%% Plots loadings (coeff1) vs Wavenumber
elseif lookatloading == true
    [coeff,score,latent,tsquared,explained] = pca(a);
    [g,h]=size(a);
    wavelength = wavelength';
    figure, hold on
    plot(wavelength,coeff(:,1));
    xlabel('Wavenumber [cm-1]'), ylabel('Loading 1'), title('Raw Data PCA')
    if isequal(a,Xsnv18)
        xlabel('Wavenumber [cm-1]'), ylabel('Loading 1'), title('SNV Preprocessed Data PCA')
    end
    if (exist ('X') == true)
        if isequal(a,X)
        xlabel('Wavenumber [cm-1]'), ylabel('Loading 1'), title('Mean Spectra from SNV Data PCA')
        end
    end


%% Plots area vs PC1
elseif lookatareapc1 == true
    [coeff,score,latent,tsquared,explained] = pca(a);
    [g,h]=size(a);
    figure, hold on
    for i=1:g
        if g == 18
            C = {[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],...
                 [0/255, 136/255, 53/255],[0/255, 136/255, 53/255],[0/255, 136/255, 53/255],...
                 [47/255, 62/255, 234/255],[47/255, 62/255, 234/255],[47/255, 62/255, 234/255],...
                 [153/255, 0/255, 0/255],[153/255, 0/255, 0/255],[153/255, 0/255, 0/255],...
                 [3/255, 15/255, 79/255],[3/255, 15/255, 79/255],[3/255, 15/255, 79/255],...
                 [252/255, 118/255, 52/255],[252/255, 118/255, 52/255],[252/255, 118/255, 52/255]};
            T = {'   50','   50','   50','   60','   60','   60',...
                        '   70','   70','   70','   80','   80','   80',...
                        '   90','   90','   90','   100','   100','   100'};
            scatter(score(i,1),areas(1,i),75,'*','MarkerFaceColor',C{i});
            text(score(i,1),areas(1,i),T{i})
        elseif g == 6
            C = {[121/255, 35/255, 142/255],[0/255, 136/255, 53/255],[47/255, 62/255, 234/255],...
                 [153/255, 0/255, 0/255],[3/255, 15/255, 79/255],[252/255, 118/255, 52/255]};
            T = {'   50','   60','   70','   80','   90','   100'};
            scatter(score(i,1),amean(1,i),75,'*','MarkerFaceColor',C{i});
            text(score(i,1),amean(1,i),T{i})
        end
    end
    xlabel('PC1'), ylabel('Area'), title('Raw Data PCA')
    if (exist ('Xsnv6') == true)
        if isequal(a,Xsnv6)
            xlabel('PC1'), ylabel('Area'), title('Mean Raw Data PCA')
        end
    end

else
% do nothing
end

wavelength = wavelength;
```

MeanSpectra.m

```
% Creates mean spectra
% Code is made for triplicates

%Size of dataset
[m,n]=size(a);

o=n-n+1;
x1=[1:m/3]';
while o<=n
y1=[];
i=1;
j=2;
k=3;
for i=1:m/3
    l=nnz(a(i,o))+ nnz(a(j,o))+ nnz(a(k,o));
    % nnz = number of nonzero matrix elements
    y1=[y1 ; ((a(i,o)+ a(j,o)+ a(k,o))/l)];
    i=i+3;
    j=k+3;
    k=k+3;
end
x1=[x1,y1];
o=o+1;
end
x1(:,1)=[];

X=[wavelength;x1];
X(1,:)=round(X(1,:));
```

PLSmodel.m

```
% PLS model

% INPUT:
% X         matrix of independent variables (e.g. spectra) (n x p)
% Y         vector of y reference values (n x 1)
% A         number of PLS factors to consider

if isequal(X,X)
    X = Xsnv18;
    Y = ref;
    ncomp = length(Y)-1;
    [n,p] = size(X);
    [Xloadings,Yloadings,Xscores,Yscores,beta,PLSPctVar] = plsregress(X,Y,ncomp);
    PLSPctVarplot = [zeros(2,1),PLSPctVar];
    figure
    plot(1:ncomp,cumsum(100*PLSPctVar(2,:)),'-bo');
    ylim([-inf 100])
    xlabel('Number of PLS components');
    ylabel('Percent Variance Explained in Y');
    title('Model Quality by Number of Components in Y')
    % Shows percentage of Y-variance explained by each PLS factor

    A = 4; % Has manually been chosen from the figure above

    % Compute the fitted response values for the model
    [Xloadings,Yloadings,Xscores,Yscores,betaPLS] = plsregress(X,Y,A);
     yfitPLSall = [ones(n,1) X]*betaPLS;

    % Shows histogram with yfit vs error bars
    format bank
    yfitPLSmean = [(yfitPLSall(1,:)+yfitPLSall(2,:)+yfitPLSall(3,:))/3; ...
        (yfitPLSall(4,:)+yfitPLSall(5,:)+yfitPLSall(6,:))/3; ...
        (yfitPLSall(7,:)+yfitPLSall(8,:)+yfitPLSall(9,:))/3; ...
        (yfitPLSall(10,:)+yfitPLSall(11,:)+yfitPLSall(12,:))/3; ...
        (yfitPLSall(13,:)+yfitPLSall(14,:)+yfitPLSall(15,:))/3; ...
        (yfitPLSall(16,:)+yfitPLSall(17,:)+yfitPLSall(18,:))/3];
    errhigh = [maxk(yfitPLSall(1:3),1)-yfitPLSmean(1,:);...
        maxk(yfitPLSall(4:6),1)-yfitPLSmean(2,:);...
        maxk(yfitPLSall(7:9),1)-yfitPLSmean(3,:);...
        maxk(yfitPLSall(10:12),1)-yfitPLSmean(4,:);...
        maxk(yfitPLSall(13:15),1)-yfitPLSmean(5,:);...
        maxk(yfitPLSall(16:18),1)-yfitPLSmean(6,:)];
    errlow = [yfitPLSmean(1,:)-mink(yfitPLSall(1:3),1);...
        yfitPLSmean(2,:)-mink(yfitPLSall(4:6),1);...
        yfitPLSmean(3,:)-mink(yfitPLSall(7:9),1);...
        yfitPLSmean(4,:)-mink(yfitPLSall(10:12),1);...
        yfitPLSmean(5,:)-mink(yfitPLSall(13:15),1);...
        yfitPLSmean(6,:)-mink(yfitPLSall(16:18),1)];
    Xname = categorical({'49.96';'60.09';'70.20';'79.87';'90.05';'100.00'});
    Xname = reordercats(Xname,{'49.96';'60.09';'70.20';'79.87';'90.05';'100.00'});
    ylabel('NIR Fitted Response');
    xlabel('Reference Values');
    title('Fitted Response Variables')
    bar1 = bar(Xname,yfitPLSmean);
    bar1.FaceColor = 'flat';
    bar1.CData(1,:) =   [1 1 1];
    bar1.CData(2,:) =   [1 1 1];
    bar1.CData(3,:) =   [1 1 1];
    bar1.CData(4,:) =   [1 1 1];
    bar1.CData(5,:) =   [1 1 1];
    bar1.CData(6,:) =   [1 1 1];
    ylim([45 105])
    hold on
    er = errorbar(Xname,yfitPLSmean,errlow,errhigh);
    er.Color = [0 0 0];
    er.LineStyle = 'none';
    xtips = bar1.XEndPoints;
    ytips = yfitPLSmean+errhigh+0.5;
    format bank
    plotlabels = string(bar1.YData);
    text(xtips,ytips,plotlabels,'HorizontalAlignment','center',...
        'VerticalAlignment','bottom')
    % The error bars display the minimum and max values of the datasets
end

X = Xsnv6;
Y = refmean;
ncomp = length(Y)-1;
[n,p] = size(X);
[Xloadings,Yloadings,Xscores,Yscores,beta,PLSPctVar] = plsregress(X,Y,ncomp);
PLSPctVarplot = [zeros(2,1),PLSPctVar];
figure
plot(1:ncomp,cumsum(100*PLSPctVar(2,:)),'-bo');
ylim([-inf 100])
xlabel('Number of PLS components');
ylabel('Percent Variance Explained in Y');
title('Model Quality by Number of Components in Y')
% Shows percentage of Y-variance explained by each PLS factor

A = 4; % Has manually been chosen from the figure above

```

```
% Computes fitted response values for the model
[Xloadings,Yloadings,Xscores,Yscores,betaPLS] = plsregress(X,Y,A);
yfitPLS = [ones(n,1) X]*betaPLS;


%% Plots fitted vs observed response for the PLS fits
[h,g]=size(Y');
figure, hold on
for i=1:g
    if g == 18
        C = {[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],...
             [0/255, 136/255, 53/255],[0/255, 136/255, 53/255],[0/255, 136/255, 53/255],...
             [47/255, 62/255, 234/255],[47/255, 62/255, 234/255],[47/255, 62/255, 234/255],...
             [153/255, 0/255, 0/255],[153/255, 0/255, 0/255],[153/255, 0/255, 0/255],...
             [3/255, 15/255, 79/255],[3/255, 15/255, 79/255],[3/255, 15/255, 79/255],...
             [252/255, 118/255, 52/255],[252/255, 118/255, 52/255],[252/255, 118/255, 52/255]};
        % Makes X=Y line for reference and r2 value
        xline = [50; 50; 50; 60; 60; 60; 70; 70; 70; ...
            80; 80; 80; 90; 90; 90; 100; 100; 100];
        yline = [50; 50; 50; 60; 60; 60; 70; 70; 70; ...
            80; 80; 80; 90; 90; 90; 100; 100; 100];
     elseif g == 6
        C = {[121/255, 35/255, 142/255],[0/255, 136/255, 53/255],[47/255, 62/255, 234/255],...
             [153/255, 0/255, 0/255],[3/255, 15/255, 79/255],[252/255, 118/255, 52/255]};
        % Makes X=Y line for reference and r2 value
        xline = [50; 60; 70; 80; 90; 100];
        yline = [50; 60; 70; 80; 90; 100];
    end
    plot(Y(i,1),yfitPLS(i,1),'*','MarkerSize', 10,'Color',C{i});
    e = errorbar(Y,yfitPLS,errlow,errhigh,'o');
    e.Marker = 'none';
    e.Color = 'k';
end

plot(xline,yline,'k-.');
xlabel('Reference Cellulose Content (% w/w)');
ylabel('Predicted Cellulose Content (% w/w)');
grid on
xlim([45 105])
ylim([45 105])
text(52, 95 , ['R^2 = 1.00'])
x0=10;
y0=10;
width=1000;
height=400;
set(gcf,'units','points','position',[x0,y0,width,height])

Rsqyline = 1 - sum((yfitPLS - yline).^2)/sum((yfitPLS - mean(yfitPLS)).^2)
```

PeakLocs.m

```
%% Separate program for peak locations
clear all;
close all;
clc

%% Load data file
X_AL = xlsread('Spectra.xlsx','C_G_all','B3:T4096');
X_ALL = X_AL.';

[t,r]=size(X_ALL);
% t is the number of samples, r is the number of variables

% Choose wavenumbers
chooselow = 7300;
choosehigh = 4000;

%Array created for wavenumber
wavelength1=X_ALL(1,2:r);
low = mink(find(abs(wavelength1-chooselow) < 2),1);
high = mink(find(abs(wavelength1-choosehigh) < 2),1);
wavelength = X_ALL(1,low:high);

%Array created for dataset
a=X_ALL(2:t,low:high);

```

```
% Initial preprocessing — negative values in import to zero
a(a<0)=0;

% Plots zoomed NIR data
figure, hold on
p501 = plot (wavelength, a(1,:), 'Color',[121/255, 35/255, 142/255]);
p502 = plot (wavelength, a(2,:), 'Color',[121/255, 35/255, 142/255]);
p503 = plot (wavelength, a(3,:), 'Color',[121/255, 35/255, 142/255]);
p601 = plot (wavelength, a(4,:), 'Color',[0/255, 136/255, 53/255]);
p602 = plot (wavelength, a(5,:), 'Color',[0/255, 136/255, 53/255]);
p603 = plot (wavelength, a(6,:), 'Color',[0/255, 136/255, 53/255]);
p701 = plot (wavelength, a(7,:), 'Color',[47/255, 62/255, 234/255]);
p702 = plot (wavelength, a(8,:), 'Color',[47/255, 62/255, 234/255]);
p703 = plot (wavelength, a(9,:), 'Color',[47/255, 62/255, 234/255]);
p801 = plot (wavelength, a(10,:), 'Color',[153/255, 0/255, 0/255]);
p802 = plot (wavelength, a(11,:), 'Color',[153/255, 0/255, 0/255]);
p803 = plot (wavelength, a(12,:), 'Color',[153/255, 0/255, 0/255]);
p901 = plot (wavelength, a(13,:), 'Color',[3/255, 15/255, 79/255]);
p902 = plot (wavelength, a(14,:), 'Color',[3/255, 15/255, 79/255]);
p903 = plot (wavelength, a(15,:), 'Color',[3/255, 15/255, 79/255]);
p1001 = plot (wavelength, a(16,:), 'Color',[252/255, 118/255, 52/255]);
p1002 = plot (wavelength, a(17,:), 'Color',[252/255, 118/255, 52/255]);
p1003 = plot (wavelength, a(18,:), 'Color',[252/255, 118/255, 52/255]);
xlabel('Wavenumber (cm^{-1})'), ylabel('Reflectance')
legend([p501 p601 p701 p801 p901 p1001],...
    {'\approx 50% w/w cellulose','\approx 60% w/w cellulose',...
    '\approx 70% w/w cellulose','\approx 80% w/w cellulose',...
    '\approx 90% w/w cellulose','100% cellulose'})
xlim([4000 7300])
set(gca,'xdir','reverse')

% Calculating and plotting peak locations
nonan = rmmissing(a);
meanvector = mean(nonan);
[pks,locs] = findpeaks(flip(meanvector),flip(wavelength));
figure
findpeaks(flip(meanvector),flip(wavelength))
set(gca,'xdir','reverse')
```

A4 - MATLAB Protein Powder Spectra

Own MATLAB code used for data handling of the protein powder samples:

Main.m

```
%% Start
clear all;
close all;
clc

% Loads raw data from excel and makes you choose spectra
run HarnesData.m

% Plots raw NIR data
run PlotRawNIR.m

% Creates raw zoomed data with initial outsorting of noise
% Makes you choose new wavenumbers
run Wavenumber.m

% Plots zoomed raw NIR data and peak locations
run PlotZoom.m

% Calculates and plots chosen area under zoomed data curves vs ref value
run RefValues.m
run NIRarea.m

% Loads preprocessing posibillities
run Preprocessings.m

% Initial PCA plots
run PCAmodel.m

%% Preprocessing of data

continueornot = input('Do you want to preprocess data? 1 = Yes, 2 = No --> ');
if continueornot == 1
    pretreat = input(...
        'Which pretreatment? 1 = SNV, 2 = MSC, 3 = S/G 1st der, 4 = S/G 2nd der --> ');
    if pretreat == 1
        a = snv(a);
    elseif pretreat == 2
        a = msc(a,1,size(a,2));
    elseif pretreat == 3
        a=X_ALL(:,high-5:low+5);
        ainitial = polydif(11,2,1,a');
        ainitial2 = ainitial(:,:,1+1)';
        a = ainitial2(:,6:length(a)-5);
    elseif pretreat == 4
        a=X_ALL(:,high-5:low+5);
        ainitial = polydif(11,2,2,a');
        ainitial2 = ainitial(:,:,2+1)';
        a = ainitial2(:,6:length(a)-5);
    end
    run pretreatplot.m
    % Looks at PCA again to see if the clustering is now better
    run PCAmodel.m
    elseif continueornot == 2
        % continue program without preprocessing
end

%% Initial PLS to choose number of PLS components and look at plotted b coefficients
run PLSinitial.m
% Makes you choose zoomed spectral area again after looking at plotted b coefficients
run Wavenumber.m

%% Pretreatment of data in the same way as before if new wavenumber is chosen
if continueornot == 1
    if pretreat == 1
        a = snv(a);
    elseif pretreat == 2
        a = msc(a,1,size(a,2));
    elseif pretreat == 3
        a=X_ALL(:,high-5:low+5);
        ainitial = polydif(11,2,1,a');
        ainitial2 = ainitial(:,:,1+1)';
        a = ainitial2(:,6:length(a)-5);
    elseif pretreat == 4
        a=X_ALL(:,high-5:low+5);
        ainitial = polydif(11,2,2,a');
        ainitial2 = ainitial(:,:,2+1)';
        a = ainitial2(:,6:length(a)-5);
    end
end

%% Final PLS plot
run PLSfinal.m

% Result: Matrix with fitted reference values vs actual reference values
PLSResult = [yfitPLS Y]

% Manual outlier detection: 1 = outlier, 0 = no outlier
% The mean predicted results are manually calculated outside of MATLAB
P1out = isoutlier(PLSResult(1:3,1))
P2out = isoutlier(PLSResult(4:6,1))
P3out = isoutlier(PLSResult(7:9,1))
P4out = isoutlier(PLSResult(10:12,1))
P5out = isoutlier(PLSResult(13:15,1))
P6out = isoutlier(PLSResult(16:18,1))
P7out = isoutlier(PLSResult(19:21,1))
P8out = isoutlier(PLSResult(22:24,1))
P9out = isoutlier(PLSResult(25:27,1))
P10out = isoutlier(PLSResult(28:30,1))
```

HarnessData.m

```
% Load wavenumber data
wavelength = xlsread('SpectraAll.xlsx','Original','A2:A2047').';

% Choose which spectra to look at
spectrachoice = input('Choose particle size: Enter 1 = Original, 2 = 1mm, 3 = 0.5mm, 4 = 0.25mm, 5 =
    0.125mm or 6 =< 0.125mm --> ');
if spectrachoice == 1
    % Load data file - Original
    X_ALL = xlsread('SpectraAll.xlsx','Original','B2:AE2047').';
elseif spectrachoice == 2
    % Load data file - 1mm
    X_ALL = xlsread('SpectraAll.xlsx','1mm','B2:AE2047').';
elseif spectrachoice == 3
    % Load data file - 0.5mm
    X_ALL = xlsread('SpectraAll.xlsx','0.5mm','B2:AE2047').';
elseif spectrachoice == 4
    % Load data file - 0.25mm
    X_ALL = xlsread('SpectraAll.xlsx','0.25mm','B2:AE2047').';
elseif spectrachoice == 5
    % Load data file - 0.125mm
    X_ALL = xlsread('SpectraAll.xlsx','0.125mm','B2:AE2047').';
elseif spectrachoice == 6
    % Load data file - <0.125mm
    X_ALL = xlsread('SpectraAll.xlsx','small0.125mm','B2:AE2047').';
end

[t,r]=size(X_ALL);
% t is the number of samples, r is the number of variables

% Initial preprocessing - negative values in import to zero
X_ALL(X_ALL<0)=0;

% Creates colour matrices
Col30 = {[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],[121/255, 35/255, 142/255],...
     [0/255, 136/255, 53/255],[0/255, 136/255, 53/255],[0/255, 136/255, 53/255],...
     [47/255, 62/255, 234/255],[47/255, 62/255, 234/255],[47/255, 62/255, 234/255],...
     [153/255, 0/255, 0/255],[153/255, 0/255, 0/255],[153/255, 0/255, 0/255],...
     [3/255, 15/255, 79/255],[3/255, 15/255, 79/255],[3/255, 15/255, 79/255],...
     [252/255, 118/255, 52/255],[252/255, 118/255, 52/255],[252/255, 118/255, 52/255],...
     [232/255, 53/255, 72/255],[232/255, 53/255, 72/255],[232/255, 53/255, 72/255],...
     [31/255, 208/255, 130/255],[31/255, 208/255, 130/255],[31/255, 208/255, 130/255],...
     [0/255, 0/255, 0/255],[0/255, 0/255, 0/255],[0/255, 0/255, 0/255],...
     [246/255, 208/255, 77/255],[246/255, 208/255, 77/255],[246/255, 208/255, 77/255]};
Col10 = {[121/255, 35/255, 142/255],[0/255, 136/255, 53/255],[47/255, 62/255, 234/255],...
     [153/255, 0/255, 0/255],[3/255, 15/255, 79/255],[252/255, 118/255, 52/255],...
     [232/255, 53/255, 72/255],[31/255, 208/255, 130/255],[0/255, 0/255, 0/255],...
     [246/255, 208/255, 77/255]};
```

PlotRawNIR.m

```
% Plots raw NIR data

figure, hold on
p1_1 = plot(wavelength, X_ALL(1,:),'Color',[121/255, 35/255, 142/255]);
p1_2 = plot(wavelength, X_ALL(2,:),'Color',[121/255, 35/255, 142/255]);
p1_3 = plot(wavelength, X_ALL(3,:),'Color',[121/255, 35/255, 142/255]);
p2_1 = plot(wavelength, X_ALL(4,:),'Color',[0/255, 136/255, 53/255]);
p2_2 = plot(wavelength, X_ALL(5,:),'Color',[0/255, 136/255, 53/255]);
p2_3 = plot(wavelength, X_ALL(6,:),'Color',[0/255, 136/255, 53/255]);
p3_1 = plot(wavelength, X_ALL(7,:),'Color',[47/255, 62/255, 234/255]);
p3_2 = plot(wavelength, X_ALL(8,:),'Color',[47/255, 62/255, 234/255]);
p3_3 = plot(wavelength, X_ALL(9,:),'Color',[47/255, 62/255, 234/255]);
p4_1 = plot(wavelength, X_ALL(10,:),'Color',[153/255, 0/255, 0/255]);
p4_2 = plot(wavelength, X_ALL(11,:),'Color',[153/255, 0/255, 0/255]);
p4_3 = plot(wavelength, X_ALL(12,:),'Color',[153/255, 0/255, 0/255]);
p5_1 = plot(wavelength, X_ALL(13,:),'Color',[3/255, 15/255, 79/255]);
p5_2 = plot(wavelength, X_ALL(14,:),'Color',[3/255, 15/255, 79/255]);
p5_3 = plot(wavelength, X_ALL(15,:),'Color',[3/255, 15/255, 79/255]);
p6_1 = plot(wavelength, X_ALL(16,:),'Color',[252/255, 118/255, 52/255]);
p6_2 = plot(wavelength, X_ALL(17,:),'Color',[252/255, 118/255, 52/255]);
p6_3 = plot(wavelength, X_ALL(18,:),'Color',[252/255, 118/255, 52/255]);
p7_1 = plot(wavelength, X_ALL(19,:),'Color',[232/255, 53/255, 72/255]);
p7_2 = plot(wavelength, X_ALL(20,:),'Color',[232/255, 53/255, 72/255]);
p7_3 = plot(wavelength, X_ALL(21,:),'Color',[232/255, 53/255, 72/255]);
p8_1 = plot(wavelength, X_ALL(22,:),'Color',[31/255, 208/255, 130/255]);
p8_2 = plot(wavelength, X_ALL(23,:),'Color',[31/255, 208/255, 130/255]);
p8_3 = plot(wavelength, X_ALL(24,:),'Color',[31/255, 208/255, 130/255]);
p9_1 = plot(wavelength, X_ALL(25,:),'Color',[0/255, 0/255, 0/255]);
p9_2 = plot(wavelength, X_ALL(26,:),'Color',[0/255, 0/255, 0/255]);
p9_3 = plot(wavelength, X_ALL(27,:),'Color',[0/255, 0/255, 0/255]);
p10_1 = plot(wavelength, X_ALL(28,:),'Color',[246/255, 208/255, 77/255]);
p10_2 = plot(wavelength, X_ALL(29,:),'Color',[246/255, 208/255, 77/255]);
p10_3 = plot(wavelength, X_ALL(30,:),'Color',[246/255, 208/255, 77/255]);
xlabel('Wavenumber (cm^{-1})'), ylabel('Reflectance')
legend([p1_1 p2_1 p3_1 p4_1 p5_1 p6_1 p7_1 p8_1 p9_1 p10_1],...
    {'Press 1','Press 2','Press 3','Press 4','Press 5','Press 6',...
    'Press 7','Press 8','Press 9','Press 10'})
xlim([min(wavelength) max(wavelength)])
ylim([0 90])
set(gcf,'units','points','position',[10,10,1000,400])
set(gca,'xdir','reverse')
```

Wavelength.m

```
% Choose which wavenumbers to look at
wavechoicelow = input('Choose wavenumber: Enter the lower limit, that is the value most to the left on the x axis --> ');
    if wavechoicelow > max(wavelength)
        disp('Choose a number between 15729 and 16 cm^(-1)');
        wavechoicelow = input('Enter the lower limit, that is the value most to the left on the x axis --> ');
    elseif wavechoicelow < min(wavelength)
        disp('Choose a number between 15729 and 16 cm^(-1)');
        wavechoicelow = input('Choose wavenumber: Enter the lower limit, that is the value most to the left on the x axis --> ');
    end

wavechoicehigh = input('Choose wavenumber: Enter the higher limit, that is the value most to the right on the x axis --> ');
    if wavechoicehigh > max(wavelength)
        disp('Choose a number between 15729 and 16 cm^(-1)');
        wavechoicehigh = input('Choose wavenumber: Enter the higher limit, that is the value most to the right on the x axis --> ');
    elseif wavechoicehigh < min(wavelength)
        disp('Choose a number between 15729 and 16 cm^(-1)');
        wavechoicehigh = input('Choose wavenumber: Enter the higher limit, that is the value most to the right On the x axis --> ');
    end

%Array created for zoomed wavenumbers
low = mink(find(abs(wavelength-wavechoicelow) < 5),1);
high = mink(find(abs(wavelength-wavechoicehigh) < 5),1);
wavelengthzoomed = wavelength(1,high:low);

```

```
%Array created for zoomed dataset;
a=X_ALL(:,high:low);
```

PlotZoom.m

```
% Plots zoomed raw NIR data

figure, hold on
p1_1 = plot(wavelengthzoomed, a(1,:),'Color',[121/255, 35/255, 142/255]);
p1_2 = plot(wavelengthzoomed, a(2,:),'Color',[121/255, 35/255, 142/255]);
p1_3 = plot(wavelengthzoomed, a(3,:),'Color',[121/255, 35/255, 142/255]);
p2_1 = plot(wavelengthzoomed, a(4,:),'Color',[0/255, 136/255, 53/255]);
p2_2 = plot(wavelengthzoomed, a(5,:),'Color',[0/255, 136/255, 53/255]);
p2_3 = plot(wavelengthzoomed, a(6,:),'Color',[0/255, 136/255, 53/255]);
p3_1 = plot(wavelengthzoomed, a(7,:),'Color',[47/255, 62/255, 234/255]);
p3_2 = plot(wavelengthzoomed, a(8,:),'Color',[47/255, 62/255, 234/255]);
p3_3 = plot(wavelengthzoomed, a(9,:),'Color',[47/255, 62/255, 234/255]);
p4_1 = plot(wavelengthzoomed, a(10,:),'Color',[153/255, 0/255, 0/255]);
p4_2 = plot(wavelengthzoomed, a(11,:),'Color',[153/255, 0/255, 0/255]);
p4_3 = plot(wavelengthzoomed, a(12,:),'Color',[153/255, 0/255, 0/255]);
p5_1 = plot(wavelengthzoomed, a(13,:),'Color',[3/255, 15/255, 79/255]);
p5_2 = plot(wavelengthzoomed, a(14,:),'Color',[3/255, 15/255, 79/255]);
p5_3 = plot(wavelengthzoomed, a(15,:),'Color',[3/255, 15/255, 79/255]);
p6_1 = plot(wavelengthzoomed, a(16,:),'Color',[252/255, 118/255, 52/255]);
p6_2 = plot(wavelengthzoomed, a(17,:),'Color',[252/255, 118/255, 52/255]);
p6_3 = plot(wavelengthzoomed, a(18,:),'Color',[252/255, 118/255, 52/255]);
p7_1 = plot(wavelengthzoomed, a(19,:),'Color',[232/255, 53/255, 72/255]);
p7_2 = plot(wavelengthzoomed, a(20,:),'Color',[232/255, 53/255, 72/255]);
p7_3 = plot(wavelengthzoomed, a(21,:),'Color',[232/255, 53/255, 72/255]);
p8_1 = plot(wavelengthzoomed, a(22,:),'Color',[31/255, 208/255, 130/255]);
p8_2 = plot(wavelengthzoomed, a(23,:),'Color',[31/255, 208/255, 130/255]);
p8_3 = plot(wavelengthzoomed, a(24,:),'Color',[31/255, 208/255, 130/255]);
p9_1 = plot(wavelengthzoomed, a(25,:),'Color',[0/255, 0/255, 0/255]);
p9_2 = plot(wavelengthzoomed, a(26,:),'Color',[0/255, 0/255, 0/255]);
p9_3 = plot(wavelengthzoomed, a(27,:),'Color',[0/255, 0/255, 0/255]);
p10_1 = plot(wavelengthzoomed, a(28,:),'Color',[246/255, 208/255, 77/255]);
p10_2 = plot(wavelengthzoomed, a(29,:),'Color',[246/255, 208/255, 77/255]);
p10_3 = plot(wavelengthzoomed, a(30,:),'Color',[246/255, 208/255, 77/255]);
xlabel('Wavenumber (cm^{-1})'), ylabel('Reflectance')
legend([p1_1 p2_1 p3_1 p4_1 p5_1 p6_1 p7_1 p8_1 p9_1 p10_1],...
    {'Press 1','Press 2','Press 3','Press 4','Press 5','Press 6',...
    'Press 7','Press 8','Press 9','Press 10'})
ylim([0 90])
xlim([min(wavelengthzoomed) max(wavelengthzoomed)])
set(gca,'xdir','reverse')

% Calculating and plotting peak locations
nonan = rmmissing(a);
meanvector = mean(nonan);
[pks,locs] = findpeaks(meanvector,wavelengthzoomed);
figure
findpeaks(meanvector,wavelengthzoomed)
set(gca,'xdir','reverse')
```

RefValues.m

```
% Choose which ref value to look at and harnes reference values
refchoice = input('Choose reference nutrient: Enter 1 = Protein, 2 = IDF, 3 = SDF, 4 = TDF, 5 = ACH, 6 = Ash --> ');
if refchoice == 1
    refmean = xlsread('SpectraAll.xlsx','RefValues','C2:L2').';
elseif refchoice == 2
    refmean = xlsread('SpectraAll.xlsx','RefValues','C3:L3').';
elseif refchoice == 3
    refmean = xlsread('SpectraAll.xlsx','RefValues','C4:L4').';
elseif refchoice == 4
    refmean = xlsread('SpectraAll.xlsx','RefValues','C5:L5').';
elseif refchoice == 5
    refmean = xlsread('SpectraAll.xlsx','RefValues','C6:L6').';
elseif refchoice == 6
    refmean = xlsread('SpectraAll.xlsx','RefValues','C7:L7').';
end
```

```

% Making ref matrix for all triplicate spectra
ref = [refmean(1,1); refmean(1,1); refmean(1,1);...
        refmean(2,1); refmean(2,1); refmean(2,1);...
        refmean(3,1); refmean(3,1); refmean(3,1);...
        refmean(4,1); refmean(4,1); refmean(4,1);...
        refmean(5,1); refmean(5,1); refmean(5,1);...
        refmean(6,1); refmean(6,1); refmean(6,1);...
        refmean(7,1); refmean(7,1); refmean(7,1);...
        refmean(8,1); refmean(8,1); refmean(8,1);...
        refmean(9,1); refmean(9,1); refmean(9,1);...
        refmean(10,1); refmean(10,1); refmean(10,1)];
```

NIRarea.m

```
% Calculates and plots area under raw data curves related to chosen refValue

areas = trapz(wavelengthzoomed, a');

figure, hold on
for i=1:30
    C = Col30;
    plot(ref(i,1),areas(1,i),'*','Color',C{i});
end

% Makes linear regression line
i = 1;
while i <= max(size(areas))
    if isnan(areas(1,i)) == 1
        areas(:,i) = [];
        ref(i,:) = [];
        i = i;
    end
    if isnan(areas(1,i)) == 0
        i = i + 1;
    end
end
maxindex = max(size(areas));
[P30,S30] = polyfit(ref,areas',1);
yfit30 = P30(1)*ref+P30(2);  % P(1)=slope and P(2)=intercept
hold on
plot(ref,yfit30,'k-.')
xlim([min(ref)-max(ref)*0.025 max(ref)+max(ref)*0.025])
ylim([min(areas)-2000 max(areas)+2000])
xlabel('Chosen Reference Content (% w/w)')
ylabel('Observed NIR Area from Zoomed Raw Data')
grid on
Rsqarea30 = 1 - (S30.normr/norm(areas - mean(areas)))^2;
text(min(ref),max(areas)-3000,['R^2 = ',num2str(Rsqarea30)])

%% Calculates and plots area under mean raw data curves related to chosen refValue
meanarea = input('Does the area show a good correlation (show mean area plot)? 1 = Yes, 2 = No --> ');
    if meanarea == 1

areas = trapz(wavelengthzoomed, a');

i = 1;
j = 0;
m = 1;
n = 0;
amean=[];
while i <= maxindex
    if isnan(areas(1,i)) == 1;
        areas(:,i) = [];
        i = i;
    else isnan(areas(1,i)) == 0;
        i = i+1;
        n = n+1;
    end
    j = j+1;
    if j == 3
        j = 0;
        if n == 1
            amean(1,m) = areas(1,i-1);
        end
        if n == 2
```

```
            amean(1,m) = (areas(1,i-1)+areas(1,i-2))/2;
        end
        if n == 3
            amean(1,m) = (areas(1,i-1)+areas(1,i-2)+areas(1,i-3))/3;
        end
        m = m+1;
        n = 0;
    end
end

figure, hold on
for i=1:10
C = Col10;
plot(refmean(i,1),amean(1,i),'*','Color',C{i});
end

% Make linear regression line
[P,S] = polyfit(refmean,amean',1);
yfit = P(1)*refmean+P(2);  % P(1)=slope and P(2)=intercept
hold on
plot(refmean,yfit,'k-.')
xlim([min(refmean)-max(ref)*0.025 max(refmean)+max(ref)*0.025])
ylim([min(areas)-2000 max(areas)+2000])
xlabel('Chosen Reference Content (% w/w)')
ylabel('Observed mean NIR Area from Raw Data')
grid on
Rsqarea = 1 - (S.normr/norm(amean - mean(amean)))^2;
text(min(ref),max(areas)-3000,['R^2 = ',num2str(Rsqarea)])

elseif meanarea == 2
    end

ref = [refmean(1,1); refmean(1,1); refmean(1,1);...
        refmean(2,1); refmean(2,1); refmean(2,1);...
        refmean(3,1); refmean(3,1); refmean(3,1);...
        refmean(4,1); refmean(4,1); refmean(4,1);...
        refmean(5,1); refmean(5,1); refmean(5,1);...
        refmean(6,1); refmean(6,1); refmean(6,1);...
        refmean(7,1); refmean(7,1); refmean(7,1);...
        refmean(8,1); refmean(8,1); refmean(8,1);...
        refmean(9,1); refmean(9,1); refmean(9,1);...
        refmean(10,1); refmean(10,1); refmean(10,1)];
```

Preprocessings.m

```
% SNV (Standard Normal Variate transformation)
[Xsnv30]=snv(a);

% MSC (Multiplicative Scatter Correction)
[xmsc30]=msc(a,1,size(a,2));

% 1st der (Savitzky-Golay 1st derivative)
[Xde130]=polydif(11,2,1,a');

% 2nd der (Savitzky-Golay 2nd derivative)
[Xde230]=polydif(11,2,2,a');
```

PCAmodel.m

```
% PCA model
[coeff,score,latent,tsquared,explained] = pca(a);
[g,h]=size(a);

% Plots PC1 vs PC2
figure, hold on
for i=1:g
    if g == 30
         C = Col30;
         plot(score(i,1),score(i,2),'*','Color',C{i});
    end
end
xline(0,':k');
```

```
yline(0,':k');
xlabel('PC1'), ylabel('PC2')
expraw = explained;

if isequal(a,Xsnv30)
    xlabel('PC1'), ylabel('PC2')
    expsnv = explained;
end

figure, hold on
for i=1:length(explained)+1
        G = cumsum(latent/sum(latent));
        G = [0;G];
        plot(i-1,G(i,1),'-bo')
        title('Explained Variance')
        xlabel('Number of PCs')
        ylabel('Percent Variance Explained in X')
end

% Plots PC1 vs RefValues
figure, hold on
p = max(size(ref));

for i=1:p
    if g == 30
        C = Col30;
        plot(ref(i,1),score(i,1),'*','Color',C{i});
    end
end
xline(0,':k');
yline(0,':k');
xlabel('Reference Content (% w/w)'), ylabel('PC1')
xlim([min(ref)-max(ref)*0.025 max(ref)+max(ref)*0.025])
grid on

% PC2 vs RefValues
figure, hold on
p = max(size(ref));

for i=1:p
    if g == 30
        C = Col30;
         plot(ref(i,1),score(i,2),'*','Color',C{i});
    end
end
xline(0,':k');
yline(0,':k');
ylabel('PC2'), xlabel('Reference Content (% w/w)')
xlim([min(ref)-max(ref)*0.025 max(ref)+max(ref)*0.025])
grid on
```

pretreatplot.m

```
% Plots pretreated spectre
figure, hold on
p1_1 = plot(wavelengthzoomed, a(1,:),'Color',[121/255, 35/255, 142/255]);
p1_2 = plot(wavelengthzoomed, a(2,:),'Color',[121/255, 35/255, 142/255]);
p1_3 = plot(wavelengthzoomed, a(3,:),'Color',[121/255, 35/255, 142/255]);
p2_1 = plot(wavelengthzoomed, a(4,:),'Color',[0/255, 136/255, 53/255]);
p2_2 = plot(wavelengthzoomed, a(5,:),'Color',[0/255, 136/255, 53/255]);
p2_3 = plot(wavelengthzoomed, a(6,:),'Color',[0/255, 136/255, 53/255]);
p3_1 = plot(wavelengthzoomed, a(7,:),'Color',[47/255, 62/255, 234/255]);
p3_2 = plot(wavelengthzoomed, a(8,:),'Color',[47/255, 62/255, 234/255]);
p3_3 = plot(wavelengthzoomed, a(9,:),'Color',[47/255, 62/255, 234/255]);
p4_1 = plot(wavelengthzoomed, a(10,:),'Color',[153/255, 0/255, 0/255]);
p4_2 = plot(wavelengthzoomed, a(11,:),'Color',[153/255, 0/255, 0/255]);
p4_3 = plot(wavelengthzoomed, a(12,:),'Color',[153/255, 0/255, 0/255]);
p5_1 = plot(wavelengthzoomed, a(13,:),'Color',[3/255, 15/255, 79/255]);
p5_2 = plot(wavelengthzoomed, a(14,:),'Color',[3/255, 15/255, 79/255]);
p5_3 = plot(wavelengthzoomed, a(15,:),'Color',[3/255, 15/255, 79/255]);
p6_1 = plot(wavelengthzoomed, a(16,:),'Color',[252/255, 118/255, 52/255]);
p6_2 = plot(wavelengthzoomed, a(17,:),'Color',[252/255, 118/255, 52/255]);
p6_3 = plot(wavelengthzoomed, a(18,:),'Color',[252/255, 118/255, 52/255]);
p7_1 = plot(wavelengthzoomed, a(19,:),'Color',[232/255, 53/255, 72/255]);
p7_2 = plot(wavelengthzoomed, a(20,:),'Color',[232/255, 53/255, 72/255]);
p7_3 = plot(wavelengthzoomed, a(21,:),'Color',[232/255, 53/255, 72/255]);
```

```
p8_1 = plot(wavelengthzoomed, a(22,:),'Color',[31/255, 208/255, 130/255]);
p8_2 = plot(wavelengthzoomed, a(23,:),'Color',[31/255, 208/255, 130/255]);
p8_3 = plot(wavelengthzoomed, a(24,:),'Color',[31/255, 208/255, 130/255]);
p9_1 = plot(wavelengthzoomed, a(25,:),'Color',[0/255, 0/255, 0/255]);
p9_2 = plot(wavelengthzoomed, a(26,:),'Color',[0/255, 0/255, 0/255]);
p9_3 = plot(wavelengthzoomed, a(27,:),'Color',[0/255, 0/255, 0/255]);
p10_1 = plot(wavelengthzoomed, a(28,:),'Color',[246/255, 208/255, 77/255]);
p10_2 = plot(wavelengthzoomed, a(29,:),'Color',[246/255, 208/255, 77/255]);
p10_3 = plot(wavelengthzoomed, a(30,:),'Color',[246/255, 208/255, 77/255]);
xlabel('Wavenumber (cm^{-1})'), ylabel('Preprocessed Data')
legend([p1_1 p2_1 p3_1 p4_1 p5_1 p6_1 p7_1 p8_1 p9_1 p10_1],...
    {'Press 1','Press 2','Press 3','Press 4','Press 5','Press 6',...
    'Press 7','Press 8','Press 9','Press 10'})
xlim([min(wavelengthzoomed) max(wavelengthzoomed)])
set(gca,'xdir','reverse')
```

PLSinitial.m

```
% Partial least squares (PLS) model building

% INPUT:
% X        matrix of independent variables (e.g. spectra) (n x p)
% Y        vector of y reference values (n x 1)
% A        number of PLS components to consider

X = a;
Y = ref;

ncomp = length(Y)-1; % Default initial value
[n,p] = size(X);
[Xloadings,Yloadings,Xscores,Yscores,beta,PLSPctVar] = plsregress(X,Y,ncomp);
PLSPctVarplot = [zeros(2,1),PLSPctVar];
figure
plot(1:ncomp,cumsum(100*PLSPctVar(2,:)),'-bo');
ylim([-inf 100])
xlabel('Number of PLS components');
ylabel('Percent Variance Explained in Y');
title('Model Quality by Number of Components in Y')

choosea = input('Choose number of PLS components: 1,2,3,4...,10 --> ');
if choosea == 1
    A = 1;
elseif choosea == 2
    A = 2;
elseif choosea == 3
    A = 3;
elseif choosea == 4
    A = 4;
elseif choosea == 5
    A = 5;
elseif choosea == 6
    A = 6;
elseif choosea == 7
    A = 7;
elseif choosea == 8
    A = 8;
elseif choosea == 9
    A = 9;
elseif choosea == 10
    A = 10;
end

PLScomponents = cumsum(100*PLSPctVar(2,:));
ExplainedVariance = PLScomponents(:,A)

% Plots b coefficients
[n,p] = size(X);
[Xloadings,Yloadings,Xscores,Yscores,betaPLS] = plsregress(X,Y,A);
yfitPLS = [ones(n,1) X]*betaPLS;
figure
plot(wavelengthzoomed,betaPLS(2:max(size(wavelengthzoomed))+1,:))
xlim([min(wavelengthzoomed) max(wavelengthzoomed)])
set(gca,'xdir','reverse')
xlabel('Wavenumber (cm^{-1})');
ylabel('b coefficients from PLS');
grid on
```

PLSfinal.m

```
% Final PLS plot

[n,p] = size(X);
[Xloadings,Yloadings,Xscores,Yscores,betaPLS] = plsregress(X,Y,A);
yfitPLS = [ones(n,1) X]*betaPLS;

figure
for i=1:1:30
    C = Col30;
    plot(Y(i,:),yfitPLS(i,:),'*','Color',C{i});
    hold on
end

% Make linear X=Y line for reference and r2 value
xline = [min(Y)-0.025*max(Y);max(Y)+0.025*max(Y)];
yline = [min(Y)-0.025*max(Y);max(Y)+0.025*max(Y)];
plot(xline,yline,'k-.');
xlabel('Reference Content (% w/w)');
ylabel('Predicted Reference Content (% w/w)');
grid on
slope = 1;
yCalc = slope*Y;
Rsqyline = 1 - sum((yfitPLS - yCalc).^2)/sum((yfitPLS - mean(yfitPLS)).^2)
text(min(Y)-0.025*max(Y),max(Y)+0.025*max(Y),['Number of PLS Components: ',num2str(A) ', R^2 = ',num2str
    (Rsqyline)])
```

Own MATLAB code used for data handling of the protein powder samples of different particle sizes:

Particlesmain.m

```
%% Start
clear all;
close all;
clc

% 1 mm
run Particlesize.m
peaks1 = locs;

% 0.5mm
run Particlesize.m
peaks05 = locs;

% 0.25mm
run Particlesize.m
peaks025 = locs;

% 0.125mm
run Particlesize.m
peaks0125 = locs;

% < 0.125mm
run Particlesize.m
peakss0125 = locs;
```

Particlesize.m

```
% Loads raw data from excel and makes you choose spectra
run HarnesData.m

% Plots raw NIR data
figure, hold on
p1_1 = plot(wavelength, X_ALL(1,:),'Color',[121/255, 35/255, 142/255]);
p1_2 = plot(wavelength, X_ALL(2,:),'Color',[121/255, 35/255, 142/255]);
p1_3 = plot(wavelength, X_ALL(3,:),'Color',[121/255, 35/255, 142/255]);
p2_1 = plot(wavelength, X_ALL(4,:),'Color',[0/255, 136/255, 53/255]);
p2_2 = plot(wavelength, X_ALL(5,:),'Color',[0/255, 136/255, 53/255]);
p2_3 = plot(wavelength, X_ALL(6,:),'Color',[0/255, 136/255, 53/255]);
```

```
p3_1 = plot(wavelength, X_ALL(7,:),'Color',[47/255, 62/255, 234/255]);
p3_2 = plot(wavelength, X_ALL(8,:),'Color',[47/255, 62/255, 234/255]);
p3_3 = plot(wavelength, X_ALL(9,:),'Color',[47/255, 62/255, 234/255]);
p4_1 = plot(wavelength, X_ALL(10,:),'Color',[153/255, 0/255, 0/255]);
p4_2 = plot(wavelength, X_ALL(11,:),'Color',[153/255, 0/255, 0/255]);
p4_3 = plot(wavelength, X_ALL(12,:),'Color',[153/255, 0/255, 0/255]);
p5_1 = plot(wavelength, X_ALL(13,:),'Color',[3/255, 15/255, 79/255]);
p5_2 = plot(wavelength, X_ALL(14,:),'Color',[3/255, 15/255, 79/255]);
p5_3 = plot(wavelength, X_ALL(15,:),'Color',[3/255, 15/255, 79/255]);
p6_1 = plot(wavelength, X_ALL(16,:),'Color',[252/255, 118/255, 52/255]);
p6_2 = plot(wavelength, X_ALL(17,:),'Color',[252/255, 118/255, 52/255]);
p6_3 = plot(wavelength, X_ALL(18,:),'Color',[252/255, 118/255, 52/255]);
p7_1 = plot(wavelength, X_ALL(19,:),'Color',[232/255, 53/255, 72/255]);
p7_2 = plot(wavelength, X_ALL(20,:),'Color',[232/255, 53/255, 72/255]);
p7_3 = plot(wavelength, X_ALL(21,:),'Color',[232/255, 53/255, 72/255]);
p8_1 = plot(wavelength, X_ALL(22,:),'Color',[31/255, 208/255, 130/255]);
p8_2 = plot(wavelength, X_ALL(23,:),'Color',[31/255, 208/255, 130/255]);
p8_3 = plot(wavelength, X_ALL(24,:),'Color',[31/255, 208/255, 130/255]);
p9_1 = plot(wavelength, X_ALL(25,:),'Color',[0/255, 0/255, 0/255]);
p9_2 = plot(wavelength, X_ALL(26,:),'Color',[0/255, 0/255, 0/255]);
p9_3 = plot(wavelength, X_ALL(27,:),'Color',[0/255, 0/255, 0/255]);
p10_1 = plot(wavelength, X_ALL(28,:),'Color',[246/255, 208/255, 77/255]);
p10_2 = plot(wavelength, X_ALL(29,:),'Color',[246/255, 208/255, 77/255]);
p10_3 = plot(wavelength, X_ALL(30,:),'Color',[246/255, 208/255, 77/255]);
xlabel('Wavenumber (cm^{-1})'), ylabel('Reflectance')
legend([p1_1 p2_1 p3_1 p4_1 p5_1 p6_1 p7_1 p8_1 p9_1 p10_1],...
    {'Press 1','Press 2','Press 3','Press 4','Press 5','Press 6',...
     'Press 7','Press 8','Press 9','Press 10'})
xlim([min(wavelength) max(wavelength)])
ylim([0 90])
set(gcf,'units','points','position',[10,10,1000,400])
set(gca,'xdir','reverse')

% Raw data with initial outsorting of noise
% Makes you choose zoomed spectral area
run Wavenumber.m

% Plots zoomed raw NIR data
figure, hold on
p1_1 = plot(wavelengthzoomed, a(1,:),'Color',[121/255, 35/255, 142/255]);
p1_2 = plot(wavelengthzoomed, a(2,:),'Color',[121/255, 35/255, 142/255]);
p1_3 = plot(wavelengthzoomed, a(3,:),'Color',[121/255, 35/255, 142/255]);
p2_1 = plot(wavelengthzoomed, a(4,:),'Color',[0/255, 136/255, 53/255]);
p2_2 = plot(wavelengthzoomed, a(5,:),'Color',[0/255, 136/255, 53/255]);
p2_3 = plot(wavelengthzoomed, a(6,:),'Color',[0/255, 136/255, 53/255]);
p3_1 = plot(wavelengthzoomed, a(7,:),'Color',[47/255, 62/255, 234/255]);
p3_2 = plot(wavelengthzoomed, a(8,:),'Color',[47/255, 62/255, 234/255]);
p3_3 = plot(wavelengthzoomed, a(9,:),'Color',[47/255, 62/255, 234/255]);
p4_1 = plot(wavelengthzoomed, a(10,:),'Color',[153/255, 0/255, 0/255]);
p4_2 = plot(wavelengthzoomed, a(11,:),'Color',[153/255, 0/255, 0/255]);
p4_3 = plot(wavelengthzoomed, a(12,:),'Color',[153/255, 0/255, 0/255]);
p5_1 = plot(wavelengthzoomed, a(13,:),'Color',[3/255, 15/255, 79/255]);
p5_2 = plot(wavelengthzoomed, a(14,:),'Color',[3/255, 15/255, 79/255]);
p5_3 = plot(wavelengthzoomed, a(15,:),'Color',[3/255, 15/255, 79/255]);
p6_1 = plot(wavelengthzoomed, a(16,:),'Color',[252/255, 118/255, 52/255]);
p6_2 = plot(wavelengthzoomed, a(17,:),'Color',[252/255, 118/255, 52/255]);
p6_3 = plot(wavelengthzoomed, a(18,:),'Color',[252/255, 118/255, 52/255]);
p7_1 = plot(wavelengthzoomed, a(19,:),'Color',[232/255, 53/255, 72/255]);
p7_2 = plot(wavelengthzoomed, a(20,:),'Color',[232/255, 53/255, 72/255]);
p7_3 = plot(wavelengthzoomed, a(21,:),'Color',[232/255, 53/255, 72/255]);
p8_1 = plot(wavelengthzoomed, a(22,:),'Color',[31/255, 208/255, 130/255]);
p8_2 = plot(wavelengthzoomed, a(23,:),'Color',[31/255, 208/255, 130/255]);
p8_3 = plot(wavelengthzoomed, a(24,:),'Color',[31/255, 208/255, 130/255]);
p9_1 = plot(wavelengthzoomed, a(25,:),'Color',[0/255, 0/255, 0/255]);
p9_2 = plot(wavelengthzoomed, a(26,:),'Color',[0/255, 0/255, 0/255]);
p9_3 = plot(wavelengthzoomed, a(27,:),'Color',[0/255, 0/255, 0/255]);
p10_1 = plot(wavelengthzoomed, a(28,:),'Color',[246/255, 208/255, 77/255]);
p10_2 = plot(wavelengthzoomed, a(29,:),'Color',[246/255, 208/255, 77/255]);
p10_3 = plot(wavelengthzoomed, a(30,:),'Color',[246/255, 208/255, 77/255]);
xlabel('Wavenumber (cm^{-1})'), ylabel('Reflectance')
legend([p1_1 p2_1 p3_1 p4_1 p5_1 p6_1 p7_1 p8_1 p9_1 p10_1],...
    {'Press 1','Press 2','Press 3','Press 4','Press 5','Press 6',...
     'Press 7','Press 8','Press 9','Press 10'})
ylim([0 90])
xlim([min(wavelengthzoomed) max(wavelengthzoomed)])
set(gca,'xdir','reverse')

% Calculating mean spectra for matrix a and peak locations
nonan = rmmissing(a);
meanvector = mean(nonan);
```

```
[pks,locs] = findpeaks(meanvector,wavelengthzoomed);
figure
findpeaks(meanvector,wavelengthzoomed)
set(gca,'xdir','reverse')
```

A5 - MATLAB Copyright

Copyright MATLAB code prepared by others used for data preprocessing for both cellulose gluten and protein powder data:

snv.m

```

%#  function [xsnv]=snv(x)
%#
%#  AIM:         Standard Normal Variate Transformation
%#                   Row centering, followed by row scaling.
%#
%#  PRINCIPLE:   Removal of the row mean from each row, followed
%#               by division of the row by the respective row
%#                        standard deviation.
%#
%#  INPUT:       x: (m x n) matrix with m spectra and n variables
%#
%#  OUTPUT:      xsnv: (m x n) matrix containing snv transformed spectra
%#
%#  AUTHOR:      Andrea Candolfi
%#                   Copyright(c) 1997 for ChemoAC
%#               FABI, Vrije Universiteit Brussel
%#               Laarbeeklaan 103 1090 Jette
%#
%#  VERSION: 1.1 (28/02/1998)
%#
%#  TEST:        Roy de Maesschalck
%#

function [xsnv]=snv(x);

[m,n]=size(x);
xsnv=(x-mean(x')'*ones(1,n))./(std(x')'*ones(1,n));
```

msc.m

```

%#  function [xmsc,me,xtmsc]=msc(x,first,last,xt)
%#
%#  AIM:         Multiple Scatter Correction:
%#          To remove the effect of physical light scatter
%#          from the spectrum. (Compensation for particle size
%#          effects.)
%#
%#  PRINCIPLE:   Each spectrum is shifted and rotated so that it fits
%#               as closely as possible to the mean spectrum of the data.
%#               The fit is achieved by LS (first-degree polynomial).
%#               The correction depends on the mean spectrum of the
%#               training set.
%#
%#  INPUT:       x: (m x n) matrix with m spectra and n variables
%#               first: first variable used for correction
%#          last: last variable used for correction
%#               (A segment is selected which is representative for the
%#               baseline of the spectra.)
%#          xt: (mt x nt) matrix for new data (optional)
%#
%#  OUTPUT:      xmsc: (m x n) matrix containing the spectra after
%#                        correction with msc
%#          me: mean spectrum (1 x n) of x
%#          xtmsc: (mt x nt) matrix containing the new spectra after
%#               correction with msc
%#
%#  AUTHOR:      Andrea Candolfi
%#               Copyright(c) 1997 for ChemoAC
%#               FABI, Vrije Universiteit Brussel
%#               Laarbeeklaan 103 1090 Jette
%#
%#  VERSION: 1.1 (28/02/1998)
%#
%#  TEST:        Roy de Maesschalck
%#
```

```

function [xmsc,me,xtmsc]=msc(x,first,last,xt);

if nargin==1;
   first=input('The first variable for the correction: ');
   last=input('The last variables for the correction: ');
end

[m,n]=size(x);
me=mean(x);

for i=1:m,                                                      % for the x data
  p=polyfit(me(first:last),x(i,first:last),1);                  % least square fit between mean spectrum and each spectrum (first-degree polynomial)
  xmsc(i,:)=(x(i,:)-p(2)*ones(1,n))./(p(1)*ones(1,n));          % each spectrum is corrected
end

if nargin ==4;                                                  % correction of new data by using the mean spectrum from x.
[mt,nt]=size(xt);
  for i=1:mt,
    p=polyfit(me(first:last),xt(i,first:last),1);               % least square fit between mean spectrum and each new spectrum (first-degree polynomial)
    xtmsc(i,:)=(xt(i,:)-p(2)*ones(1,n))./(p(1)*ones(1,n));      % each new spectrum is corrected
  end
end

end
```

deriv.m

```

%#  function [dx] = deriv(x,der,window,order)
%#
%#  AIM:        Derivative computation by using the #Savitsky-Golay#
%#          algorithm.
%#
%#  PRINCIPLE:  Differentiation by convolution method.
%#
%#  INPUT:      x       - Data Matrix: (nxm) n spectra m variables
%#          der - (1x1) degree of the derivative;
%#                        it must be <= order
%#          window      - (optional), (1x1) the number of points
%#                        in filter, it must be >3 and odd
%#          order       - (optional), (1x1) the order of the polynomial
%#                        It must be <=5 and <= (window-1)

%#
%#  OUTPUT:     dx      - Matrix of differentiated function (nxm)
%#
%#  SUBROUTINE:
%#              weight.m
%#              genfact.m
%#              grampoly.m
%#
%#  AUTHOR:     Luisa Pasti
%#              Copyright(c) 1997 for ChemoAc
%#              FABI, Vrije Universiteit Brussel
%#              Laarbeeklaan 103 1090 Jette
%#              Modified program of
%#              Sijmen de Jong
%#              Unilever Research Laboratorium Vlaardingen
%#
%#  VERSION: 1.1 (28/02/1998)
%#
%#  TEST:       Kris De Braekeleer
%#

function dx = deriv(x,der,window,order)

[nr,nc]=size(x);
if (nargin<4)
  order = 2;
  disp(' Polynomial order set to 2')
end
if (nargin<3)
```

```
    window=min(17,floor(nc/2));
    disp(['  Windows size set to ',num2str(window)]);
end
if (nargin<2)
    disp(' function dx = deriv(x,der)')
end

m = fix(window/2);

p = round(window/2);

o=order;

for i=1:window
      i0=i-p;
      for j=1:window,
          j0=j-p;
          w(i,j)=weight(i0,j0,m,o,der);
      end
end
yr(:,1:m)=x(:,[1:window])*w(:,1:m);                % First window
for i=1:(nc-2*m)                                   % Middle
      yr(:,i+m)=x(:,[i:(i+2*m)])*w(:,p);
end
a=nc-2*m;                                          % Last window
yr(:,(nc-m+1):nc)=x(:,a:nc)*w(:,p+1:window);
dx=yr;

end
```

weight.m

```

%#  function [sum] = weight(i,t,m,n,s)
%#
%#  AIM:          Derivative computation by using the Savitsky-Golay
%#            algorithm: Weight computation.
%#
%#  PRINCIPLE:  Computation of the weight.
%#
%#  INPUT:        i        - index of the ith data point
%#            t    - index of the tth Leat Square point of the
%#                           s derivative
%#            m    - the number of points in filter
%#            n    - order of the polynomial
%#            s    - derivative order
%#
%#  OUTPUT:       sum      - Matrix of weight
%#
%#  SUBROUTINE:
%#                genfact
%#                grampoly
%#
%#  AUTHOR:       Luisa Pasti
%#                Copyright(c) 1997 for ChemoAc
%#                FABI, Vrije Universiteit Brussel
%#                Laarbeeklaan 103 1090 Jette
%#                Modified program of
%#                Sijmen de Jong
%#                Unilever Research Laboratorium Vlaardingen
%#
%#  VERSION: 1.1 (28/02/1998)
%#
%#  TEST:         Kris De Braekeleer
%#

function sum=weight(i,t,m,n,s)

sum=0;
for k=0:n;
sum=sum+(2*k+1)*(genfact(2*m,k)/genfact(2*m+k+1,k+1))*...
grampoly(i,m,k,0)*grampoly(t,m,k,s);
end
```

genfact.m

```

%#  function [gf] = genfact(a,b)
%#
%#  AIM:         Derivative computation by using the Savitsky-Golay
%#          algorithm: Weight computation.
%#
%#  PRINCIPLE:  Calculates the generalized factorial (a), (a-1)...
%#
%#  INPUT:       a        - equal to 2*m, m is the length of the filter
%#          b   - index of the data point
%#
%#  OUTPUT:      gf       - generalized factorial vector
%#
%#  AUTHOR:      Luisa Pasti
%#               Copyright(c) 1997 for ChemoAc
%#               FABI, Vrije Universiteit Brussel
%#               Laarbeeklaan 103 1090 Jette
%#               Modified program of
%#               Sijmen de Jong
%#               Unilever Research Laboratorium Vlaardingen
%#
%#  VERSION: 1.1 (28/02/1998)
%#
%#  TEST:        Kris De Braekeleer

%#

function gf=genfact(a,b)
gf=1;
for i=(a-b+1):a
  gf=gf*i;
end
```

grampoly.m

```

%#  function [y] = grampoly(i,m,k,s)
%#
%#  AIM:         Derivative computation by using the Savitsky-Golay
%#          algorithm: Weight computation.
%#
%#  PRINCIPLE:  Calculates the Gram Polynomial
%#
%#  INPUT:       i        - index of the data point
%#          m   - index of the filter length
%#          k   - order of the polinomial
%#          s   - order of the derivative
%#
%#  OUTPUT:      y        - Gram Polynomial vector
%#
%#  AUTHOR:      Luisa Pasti
%#               Copyright(c) 1997 for ChemoAc
%#               FABI, Vrije Universiteit Brussel
%#               Laarbeeklaan 103 1090 Jette
%#               Modified program of
%#               Sijmen de Jong
%#               Unilever Research Laboratorium Vlaardingen
%#
%# VERSION: 1.1 (28/02/1998)
%#
%#  TEST:        Kris De Braekeleer
%#

function y=grampoly(i,m,k,s)
if k>0
  r1=grampoly(i,m,k-1,s);
  r2=grampoly(i,m,k-1,s-1);
  r3=grampoly(i,m,k-2,s);
y=((4*k-2)/(k*(2*m-k+1)))*(i*r1+s*r2)-(((k-1)*(2*m+k))/(k*(2*m-k+1)))*r3;
else
  if ((k==0)&(s==0)),y=1;else y=0;end
end
```