Monitoring of different metabolic states of *Sacchoromyces cerevisiae* via NIR spectroscopy at low biomass concentration

A thesis presented for the degree of
Master of Science (MSc)

Vienna University of Technology
Faculty of Technical Chemistry
Institute of Chemical, Environmental and Bioscience Engineering
Research Group Integrated Bioprocess Development

Supervised by
Associate Prof. Dipl.-Ing. Dr.nat.techn. Oliver SPADIUT

Author
Markus HADERER, BSc

Submitted
28$^{\text{th}}$ June 2020

Declaration of Academic Integrity

Name: Markus HADERER, BSc

Birth Date: 13.05.1992

Matr. Num.:   01126247

I declare in lieu of oath, that I wrote this thesis and performed the associated research myself, using only literature cited in this volume. If text passages from sources are used literally, they are marked as such. I confirm that this work is original and has not been submitted elsewhere for any examination, nor is it currently under consideration for a thesis elsewhere.

Vienna, June, 2020

. . . .
Signature

Acknowledgements

First of all, I would like to thank Dr. Oliver Spadiut and Dr. Martin Joksch for their support. They were always at my side with their advice, knowledge and expertise and helped me from the start with all the adversities that occurred. Thank you for the chance that I was allowed to write my thesis with you.

I would also like to thank Johanna Hausjell and Benedikt Hufnagl for their support with the evaluation and interesting discussions for further analysis.

I would also like to thank my colleagues in the research group for their help with occasional problems, the good cooperation and a great working environment in general. I will think back on individual conversations and experiences forever.

Last but not least, I would like to thank my family and friends!
Thanks to my parents for the generous support of my career since childhood and the never-ending care. Thanks to my brother Michael, who always had good advice with his technical knowledge. Thanks to my girlfriend Claudia, who compassionate stood by my side and supported me in the best way imaginable.

Zusammenfassung

Mikroorganismen, wie die Hefe *Saccharomyces cerevisiae*, werden in der Biotechnologie in Fermentationen eingesetzt, um unterschiedlichste Produkte - vom Bier bis hin zu Medikamenten - herzustellen. Diese industriellen Bioprozesse unterliegen strengen gesetzlichen Richtlinien, die eine Überwachung der Qualität und Quantität der erzeugten Produkte verlangen. Diese Kontrolle wird mittels Proben aus dem Prozess durchgeführt.

Off-line Probenahmen stellen jedoch einen Verlust an Biomasse und/oder Produkt dar und die Gefahr der Kontamination des Prozesses ist gegeben, ebenso ist die Aufbereitung der Proben sehr zeitaufwendig. *On-line* Messungen durch spektroskopische Methoden können viele Daten in Echtzeit liefern, ohne dass der Bioreaktorinhalt berührt oder gestört wird.

Mithilfe von NIR-Daten von Fermentationen aus 2016 und 2019 wurden Analysen zum Metabolismus der Hefen bei einer Biomassekonzentration von $\leq 2.5 g/L$ gemacht. Dabei wurden die Fermentationen unterschiedlichen Prozessbedingungen ausgesetzt. Einerseits wurden Standard-Fermentationen durchgeführt, bei denen möglichst optimale Bedingungen für die Hefen eingestellt wurden. Diese dienten bei der folgenden Analyse als Sollwerte für einen ideal laufenden Prozess. Weiters wurden verschiedene Störprozesse aufgezeichnet, bei denen die Umgebung der Hefen (das Medium) kontrolliert in einen suboptimalen Bereich geregelt wurde. So wurden zu viel oder zu wenig Base zugegeben, sodass der pH-Wert den Metabolismus beeinflusst, oder die Sauerstoffversorgung unterbrochen, um einen Wechsel zum anaeroben Metabolismus zu erzwingen. Diese Störungen sollten realitätsnahe Ausfälle von einzelnen Parametern in Industrieprozessen simulieren. Die erhaltenen Ergebnisse wurden mithilfe der Software *R* aufbereitet und mittels PCA analysiert. Als Kontrolle für den vorherrschenden Metabolismus dienten *off-line* Proben.

Die Ergebnisse aus den spektralen Daten zeigten dabei, dass eine Aussage über den jeweiligen Metabolismus getroffen werden kann. Es sind sowohl die Standard-Fermentationen von den Stör-Fermentationen unterscheidbar als auch die gestörten Prozesse untereinander. Ein Vergleich mit Daten, die in 2016 aufgezeichnet wurden, lieferte ähnliche Ergebnisse in Bezug auf die physiologischen Daten. Die spektralen Daten von 2016 und 2019 lieferten unterschiedliche Ergebnisse, jedoch lagen nicht ausreichend Daten vor, um einen genauen Grund für den Ursprung dieses Unterschieds nennen zu können.

Sind die Ergebnisse der Daten von 2019 durch weitere Fermentationen reproduzierbar, so kann ein Modell für die Erkennung von Abweichungen erstellt werden. Weiters kann durch erhöhte zeitliche Probennahme festgestellt werden wie früh eine Abweichung vom Optimum erkannt werden kann. Abhängig vom Ergebnis des Trends über die Zeit, würde, bei einer signifikanten Abweichung im suboptimal-laufenden Prozess, ein früheres Eingreifen ermöglicht werden.

Abstract

Microorganisms, such as the yeast *Saccharomyces cerevisiae*, are used in fermentations in biotechnology to produce a wide variety of products - from beer to medicines. These industrial bioprocesses are subject to strict legal guidelines that require monitoring of the quality and quantity of the individual products. These controls are followed up with samples from the process. However, *off-line* sampling is a loss of biomass and/or product, there is a risk of contamination of the process, and the preparation of the samples is very time-consuming. *On-line* measurements via spectroscopic methods can provide a lot of data in real time without touching or disturbing the contents of the bioreactor .

With the help of NIR data from fermentations from 2016 and 2019, analyses of the metabolism of the yeast were made at a biomass concentration of $\leq 2.5 g/L$.
The fermentations were exposed to different process conditions. On the one hand, standard fermentations were carried out in which the best possible conditions for the yeasts were set. In the following analyses, these served as target values for an ideally running process. Furthermore, various disturbance processes were recorded in which the environment of the yeast (the medium) was deviated into a sub-optimal range. Too much or too little base was added so that the pH affected the metabolism, or the oxygen supply was interrupted to force a change to the anaerobic metabolism. These disturbances should simulate realistic failures of individual parameters in the industry. The results obtained were processed using the *R* software and analysed using PCA. *Off-line* samples served as controls for the present metabolism.

The results from the spectral data showed that a statement about the respective metabolism can be made. The standard fermentations can be distinguished from the disturbed fermentations as well as the disturbed processes among each other. A comparison with data recorded in 2016 gave similar results with regard to the physiological data. The spectral data from 2016 and 2019 gave different results, but there was insufficient data to give an exact reason for the origin of this deviation.

If the results of the 2019 data are reproducible through further fermentations, a model for the detection of deviations can be created. Furthermore, it can be determined - by increased time sampling - how early a difference from the optimum can be recognized. Depending on the result of the trend over time, an earlier intervention would be possible if there was a significant deviation in the suboptimal running process.

Contents

Chapter 1 - Motivation

Since the FDA published an industrial guideline for process analytical monitoring of products manufactured by microorganisms in the early 2000s, the targeted quality control has constantly been improved and promoted. If a deviation in the process can be detected quickly, it is possible to work against this error at an early stage and avoid cost-intensive failures.

Especially in the biotechnological field, deviations from optimal conditions lead to irreversible changes in cells and reaction media. The better the microorganism is supplied with its optimal conditions for life and/or growth (or later in the process, the best conditions for the production and possible folding of proteins and enzymes), the better the quantity and quality of the final product. In general, the individual metabolic states of the yeast are used from food to medicinal industry, which is why a more accurate determination of the metabolic state can improve the handling of processes. *On-line* monitoring via spectrometers can give real time insights into the state of the media, but not necessarily on the cell's metabolic state.

For this thesis, fermentations operated in 2016 (fermentation number starting with 1; hereinafter "historic") and fermentations operated in 2019 (fermentation numbers starting with 2; hereinafter "new") have been compared. Using their respective spectral data of batch fermentation executed with *Saccharomyces cerevisiae*, an attempt was made to determine the metabolic states of the yeast. At minimal biomass concentration levels ($\leq 2.5g/L$) and at high stirring rates, the spectral data received from the *in-situ* installed NIR probe demonstrated that (after preprocessing and analysis using the software R) it can be used to make comprehensible predictions on the metabolic state of the yeast. Furthermore, workflows shall be established and developed within the laboratory the study was conducted in.

1

Chapter 2 - Novelty of this Thesis

Already in the 1990s, fed-batch processes using *S.cerevisiae* have been examined spectroscopically for its anaerobic ethanol fermentation with a wide concentration range of biomass (1-60 g/L) in a 5 L reactor. [1, 2] Although these processes were executed at high stirring rates (at around 300-500 rpm), the fibre optic probes were placed outside and measured through the glass-walls of the vessel. The focus of these measurements was to analyse the concentrations of water, ethanol and yeast in varying mixtures. Their results demonstrated that ethanol in complex medium can be estimated independently of the yeast concentration. Further research in water/ethanol models successfully predicted the ethanol concentration. Nevertheless, the present yeast biomass led to high scatter effects and therefore noise addition in the measurements. [1, 2] Those researches just had no or modest aeration rates in their experiments which resulted in spectra that had better quality due to no added influence by the aeration rate.

By considering high aeration rates in combination with yeast processes, attempts [3] have been made in fed-batch processes with *P.pastoris* for producing mammalian protein. Here, NIR spectroscopy was used for monitoring concentrations of biomass, glycerol, methanol and products establishing during fermentation. During the fermentation, the challenge in monitoring increased as the cell density made the non-viscous medium change to viscous and saturated the transmission monitoring.

The challenges occurring with aeration, viscosity changes of the media and agitation, are the reason why NIR is not as widely spread within industrial bioprocesses as it might seem. [4] Besides, other publications that are available in literature are most often related to at-line NIR spectroscopy or on-line measurements using simple fermentation matrices. [5] For NIR spectroscopy to be accepted as a standard monitoring device by the bioengineering and pharmaceutical industry, more on-line studies have to be made with challenging conditions and in-situ, as many processes require sterility and an immersed probe installed within the vessel for autoclaving. [5, 6]

In this study, the spectroscopic NIR probe was installed inside the reactor and exposed to vigorous agitation and aeration during batch fermentations. At the same time the biomass concentration was at $\leq 2.5 \, g/L$ to overcome problems with potential viscous media, to inspect if the metabolic changes are also visible at low concentrations and to reproduce the existing data as similar as possible. Successfully measuring at low concentrations would pave way for possible applications within early stages at small scale volumes to determine metabolism states prior inoculating the next scale size up. It could be shown that (after processing) the spectral data was able to differentiate between different metabolic states within the yeasts. Although no high-end equipment of spectroscopic devices was used, the data delivered a promising outcome for future applications. Being able to be independent of high-end equipment, cheaper materials can be used to monitor processes and therefore lead to a more economic use than the alternative wasteful off-line or at-line measurements.

Chapter 3 - Introduction

3.1 Bioprocess Engineering

Combining the new sub-areas since the 1990s, the bioprocess engineering understands itself as the basis of industrial process development and can gain insight into the running processes with their models - both in the processes within the microorganism itself, as well as those in which the organism is located (its environment). The process of cultivating microorganisms is usually referred to as fermentation, while the vessel in which the cultivation takes place is referred to as a bioreactor. The fermentation processes can in turn be divided into an *upstream*- and *downstream*-part. The *upstream*-part of a process deals with the growth of the organism, the induction (initiation of a production) and the harvest of the microorganism. The *downstream*-process involves the processing of the target products from the cell suspension or from the cells. At a cost of 30-50% of the total manufacturing cost, it plays a significant role in testing whether a process is economical. [7]

3.1.1 Mode of Operation

Carrying out bioprocesses, it is possible to differentiate between three different modes depending on the feed and removal of elements of the process. These are the batch, fed-batch and continuous operation.

In a batch process, the nutrient medium is placed in a reactor and after adjusting all environmental parameters (pressure, temperature, aeration rate, pH, etc.) to an optimum, the culture is inoculated (inoculation). In the following process, apart from the environmental parameters, no change is made to the process itself (no addition of substrate during the process).

In a fed-batch process, the construction of the reactor and the inoculation are performed identically to the batch process, but substrate is added during the process (feed), in addition to any substrate present at the beginning of the fermentation. On average, the process can be carried out longer because there is no limitation of the metabolism due to missing substrate. However, the feeding has the disadvantage that the reactor filling volume increases steadily and thus a volumetric limitation of the process occurs.

In a process in continuous operation (initial setup equal to batch mode), the nutrient medium is added continuously, but also product (and/or biomass) are taken from the process. Due to the fact that feed and outflow (possibly) balance each other, the problem with the level of the fermenter can be bypassed.

The decision of which of the three modes to choose for each process is usually linked to economic factors, which have to consider not only costs but also time schedules (duration of fermentations).

3.1.2   Process Monitoring

Ongoing processes cannot only be monitored for their parameters by using different technologies, but also by different methods of taking samples. A distinction can be made between four possible approaches: off-line, at-line, on-line and in-line (Figure 3.1). The first two methods are mainly distinguished by the time factor. [8]

off-line and at-line monitoring

    The *off-* and *at-line* procedures consist of a manual sampling, the analysis and evaluation of the result in a laboratory and the resulting control of the running process afterwards by entering the obtained values. The location of the analysis lab depending on the running process gives the basis for the separation into *off-* (remote treatment) and *at-line* (close to the process) procedures. Examples of *at-line* methods are the measurements by HPLC (*high pressure liquid chromatography*), while the weighting of dry biomass or the cell count are classic *off-line* methods. [9]

in- and on-line monitoring

    The results of recorded data from *in-line* and *on-line* measurements are given at approximately the same time as their recording. While with *in-line* measurements, the values are taken directly in the process stream, the *on-line* samples are taken in a bypass of the process stream. The different application of the two methods is made depending on the process, but also according to construction requirements. Examples of both methods are the measurements using spectroscopic or fluorescence sensors, the impedance measurement or detection by NMR (*nuclear magnetic resonance*). [9]

Soft sensors

    Due to the lack of the time component in *in-line* and *on-line* methods, an implementation of so-called *soft sensors* is an essential part of these measurement methods. *Soft sensors* receive the directly measured data on process parameters and create (calculate) - by mathematical, deterministic or statistical models - new values that are incorporated into the control of individual parameters or the entire process. In this way any processes that are "disturbed" can be detected and corrected quickly. While in fermentation processes the main focus is on the possible cost reduction together with the better ability to control product quality, in time-critical processes the time of recognition of the situation deviating from the optimum is of enormous importance (like for example in waste water treatment plants). [9]
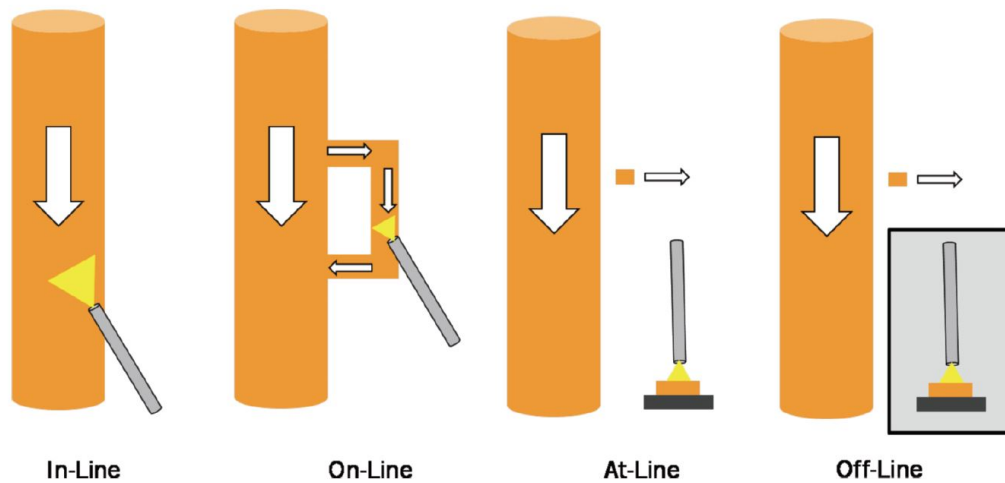
Figure 3.1 – Sampling locations in processes [10]

3.1.3 Process Analytical Technology - PAT

Since manual sampling risks contaminating the sample or even the entire process, an in-/on-line measurement is to be preferred to the at-/off-line measurement. Nevertheless, for process-technical reasons, it is not always possible to monitor in-/on-line. The most widely used methods for process monitoring are still the sample-based monitoring, as it distinguishes itself by the advantage of the high degree of automation of the analytical instruments used or the direct measurement of specific analysts. Furthermore, apart from a direct application of reference methods and a parallelization, the samples taken can be kept for a longer period of time so that a subsequent analysis can also be carried out. These samples are used to ensure safe products, as well as the recording and control of *critical process parameters* (CPP) and are therefore one of the central tasks of *process analytical technology* (PAT). Accordingly, the *US Food and Drug Administration* (FDA) issued a "Guide" to biotechnological industries in 2004. [11] Sample-based measurements are often used in combination with *Standard PAT Tools* (e.g., in-line sensors for temperature, pH-probes, pO_2-values). Even gas flow analysers or infrared measurements can be combined to a higher level of process analytical methods. These advanced PAT tools enable continuous process monitoring by linking the continuous signals of the sensors with mathematical models to soft sensors. The main reason that these sensors have not yet replaced sample-based process monitoring is their low specificity. [11, 12, 13] However, for modelling and implementation of a soft sensor, the data from single, one-dimensional (*univariate*) measurements, such as pH- or temperature-values alone, are not sufficient, as they only represent part of the overall process. It is necessary to link a large number of univariate quantities or to use the superimposed information from the multidimensional (*multivariate*) data acquired by a corresponding detector (for example: NIR spectra; see 3.3.2). [9]

3.2 Microorganism

The yeast used in this thesis[1] is commonly referred to as "baker's yeast" and is the genus of *Saccharomyces* (= ancient Greek: "sugar mushroom") which has been used for millennia for the production of food such as bread, wine or beer. The *Saccharomyces cerevisiae* is a eukaryotic microorganism whose genome consists of over 12 billion base pairs and comprises approximately 6275 genes. A single cell has a diameter of 5 to 10 μm. The reproduction happens vegetative via budding or pullutation, but it can also occur in hyphae form [14] or produce ascospores [15]. The yeast cells have glycoproteins (adhesins) on their cell walls, which lead to the formation of flakes. This flocculation is caused by stress during transcription, e.g. under nutrient limitation, and plays an important role in the separation of the cells from the fermentation medium. Since the harvest step of a process is done via centrifugation and filtration and it is very cost-intensive. The flocculation is used to improve and support the harvest. Via agglomeration of thousands of yeast cells, which leads to an increased particle diameter (increase from 10 to 500μm), the sedimentation rate can be increased by a factor of 2500. [7]

3.2.1 Metabolism

Baker's yeast can metabolize the energy that is needed to its household in both anaerobic and aerobic ways. If the organism is able to produce energy in both anaerobic (via fermentation; reductive) and aerobic (via respiration; oxidative) conditions, the organism is designated as *facultative anaerobic.* [7] For a more detailed explanation of the metabolic energy production of yeast, the following assumptions (1-7) are made (based on the definition of the model description from *Sonnleitner and Käppeli* [16]):

1. Yeast growth in batch mode with glucose and ethanol as nutrients follows nearly ideal Monod kinetics.

2. There are no by-products in significant quantities during the process; the resulting main products in diluted cultures are biomass, CO_2, H_2O, and, under appropriate conditions, ethanol. The recovery of carbon is always nearly 100%.

3. The specific oxygen uptake rate in oxidative metabolism is linearly correlated with the specific glucose uptake rate such as subcritical glucose flux or ethanol growth. Under oxidoreductive conditions of supracritical glucose flux (conditions in which ethanol is produced by the cell), the specific oxygen uptake rate remains constant at its maximum regardless of the growth rate.

4. Glucose inhibits the uptake of ethanol as a substrate insofar as it is present in a measurable amount.

---

[1]see 9.1.1 for more details on the yeast

5. Glucose can be metabolized both aerobically and anaerobically, but at a different rate and efficiency.

6. Ethanol can only be metabolized anaerobically.

7. The composition of the biomass does not change, regardless of whether ethanol is produced or not. The composition of the biomass changes slightly as it grows on ethanol as the primary substrate compared to glucose, but the differences are within an analytical error.

Based on these assumptions, metabolic energy production based on glucose and ethanol as substrates can be subdivided into two (respectively three) types: the oxidative degradation of glucose (Formula 3.1) and ethanol (Formula 3.2) and the reductive degradation of glucose (Formula 3.3).

$$C_6H_{12}O_6 + a\,O_2 + b\,NX\,[NH_3] \longrightarrow b\,C_1H_{HX}O_{OX}N_{NX} + c\,CO_2 + d\,H_2O \qquad (3.1)$$

$$C_2H_6O + k\,O_2 + l\,NX\,[NH_3] \longrightarrow l\,C_1H_{HX}O_{OX}N_{NX} + m\,CO_2 + n\,H_2O \qquad (3.2)$$

$$C_6H_{12}O_6 + g\,NX\,[NH_3] \longrightarrow g\,C_1H_{HX}O_{OX}N_{NX} + h\,CO_2 + i\,H_2O + j\,C_2H_6O \qquad (3.3)$$

The molecular weight of the resulting biomass with the chemical formula $C_1H_{HX}O_{OX}N_{NX}$ is calculated from elementary analysis of the biomass itself. The nitrogen source can be neglected in terms of nitrogen atom balance because the nitrogen atom(s) will not change the oxidative state. However, consideration must be given to the number of hydrogen atoms present in the nitrogen source. [16]

It is believed that the cell's respiratory capacity controls the glucose and ethanol metabolism in growing cells as well as product formation. It represents a bottleneck for oxidative substrate utilization. Figure 3.2 shows the flow of substrates and the respiratory capacity, shown as a bright green ring (bottleneck), for the respective substrate combinations of glucose and ethanol. A column-wise distinction can be made between the three types of substrate usage - *subcritical*, *critical* and *supracritical*.
In the case of subcritical metabolism, there are insufficient amounts of substrate to fully utilize the respiratory capacity (left column). If more substrate is added to the medium, it can be oxidatively consumed up to the critical stage (the bottleneck is completely filled; middle column). If the amount of substrate exceeds the maximum possible oxidative conversion (more substrate available than fits through the bottleneck), the supracritical state occurs (right column).

The top row of Figure 3.2 describes the case in which only glucose (orange arrows) is present as a substrate. The glucose is oxidatively consumed until the supracritical state is reached. In this state, the maximum possible amount is oxidatively metabolized, while other existing glucose is reductively converted to ethanol (fermentation) (striped orange arrow; A). If both substrates, glucose and ethanol (blue arrows), are present in the medium (second and third row) with glucose in an insufficient quantity to reach a supracritical state, all the glucose is oxidatively metabolized and the remaining respiratory capacity is used to oxidatively metabolize the available ethanol (blue arrow; B).

If the amount of substrates exceeds the respiratory capacity, the existing ethanol that is present in the medium during a supracritical state cannot be consumed reductively (third row; C). It can only be metabolized when the respiration capacity for oxidative degradation is available again (second row, far right; B).
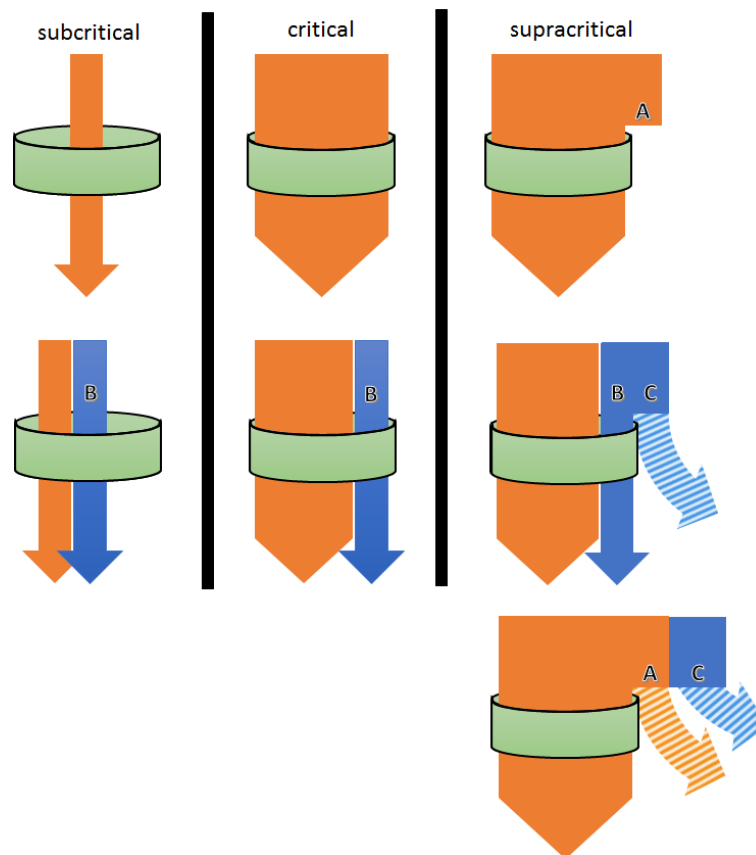
Figure 3.2 – Respiratory capacity in *bottleneck*-representation (green ring); adapted from [16] The arrows indicate substrates glucose (orange) and ethanol (blue) in three metabolic states

3.2.2   *Saccharomyces cerevisiae* in bioprocessing

One of the oldest food-manufacturing processes is bread making and especially *S.cerevisiae* strains are chosen for its dough-leavening characteristics. Since the 1990's, the recombinant-DNA technology has led to new formulation, ingredients and process conditions and now the role of baker's yeast changed into the possibility of using it as a cell factory for expression of heterologous genes. [17, 18] *S.cerevisiae* is nowadays used for production of several large volume products such as insulin and insulin analogues, but also human serum albumin or hepatitis vaccines. In 2012, 20% from those so-called "biopharmaceuticals" were produced by *S.cerevisiae*[2]. The yeast's eukaryotic model system enables production and proper folding of many human proteins. In addition, the products can be secreted into the extracellular medium, making it possible to optimize subsequent purification and in many cases, yeast can perform *post-translational modifications* (PTM) of the produced protein, including acylation, glycosylation and numerous other. [19, 20]

The secretory pathway in yeast is quite complex and has traditionally focused on transcriptional regulation of the protein production. The goal to further improve this protein production can be studied by engineering the pathway [20], and via metabolic engineering cycles where systems biology tools are implemented for the design of improved cell factories. These approaches need detailed analysis of the cellular metabolism and physiology. A major limitation in using those approaches of model engineering is the difficulty to obtain information about the metabolic state of the microorganism (yeast) within the process. [19]
This information can be obtained by using process parameters to develop models allowing a rough insight into the process (see 3.1.3).

---

[2]40% of production uses mammalian cell lines, mainly using chinese hamster ovarian cells (CHO), and for another 30%, *E.coli* cells are used. [19]

3.3 Infrared Spectroscopy

3.3.1 Near-Infrared Spectroscopy

For a long time, near-infrared spectroscopy (NIR) has been treated as an addition to analysis done via ultraviolet/visible- (UV/Vis) and mid-infrared (MIR) laboratory spectrometers (Figure 3.3). This changed at around 1980 and nowadays NIR spectroscopy is used in a variety of sectors for its availability of efficient chemometric evaluation routines. Especially the increasing costs and legal requirements for quality control, conclude to the fact, that the NIR spectroscopy is the method of choice for industrial on-line control [21]. Currently used off-line methods, such as HPLC, NMR, mass spectroscopy and wet chemicals methods, combined with diverse in- and on-line control and measurement techniques, are time consuming, need solvents for their application and/or cannot be applied remotely.

NIR has the potential to be, or already is, the new instrumental accessory for industrial process monitoring in a wide field of applications (see 3.3.2) [4, 21]. For bioprocesses with a high cell density or processes, where the biomass has a complex structure such as filamentous fungal or Saccharomycete culture and can disturb the transmission, reflectance spectroscopy mode is used. Besides off-line and at-line approaches the spectroscopy probe can be placed on-line and *ex-situ* where the device is physically outside of the reactor being placed on the glass wall or using a flow-through cell or *in-situ* where the device is placed into the reactor like a pH-probe. The direct approach of the *in-situ* method was chosen for this thesis, since, on the one hand, the set-up of the used bioreactor did not allow *ex-situ* application and on the other hand the direct measurement method allows for monitoring both the chemical and biological changes in the complex broth composition. [21]
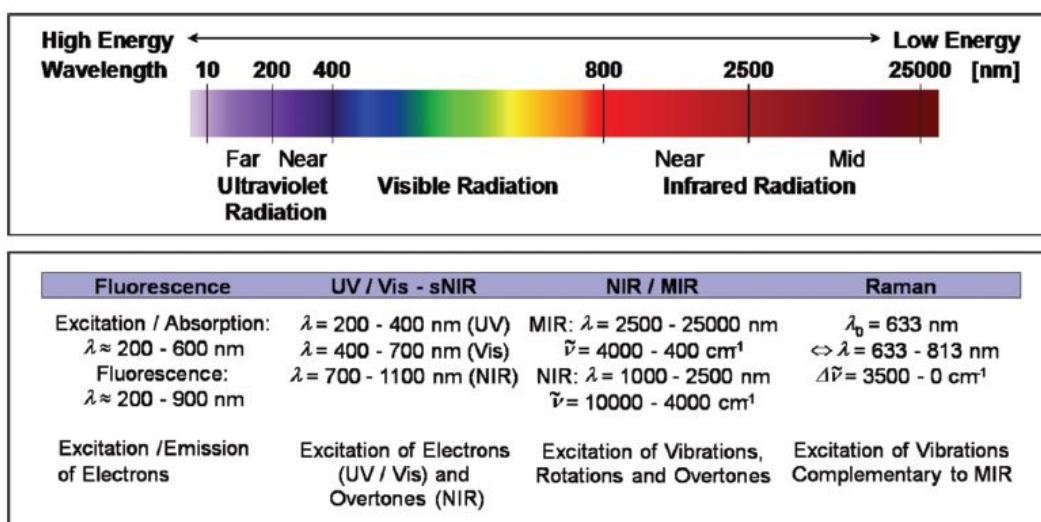

Figure 3.3 – Electromagnetic spectrum with its respective influence on electrons [10]

3.3.2   Application of NIR

Mainly used for quality control of raw materials, intermediates or finished products, NIR spectroscopy can be found in different technology fields like agricultural, food, chemical and pharmaceutical industry. Being a cost efficient and fast alternative to other control methods, the NIR spectroscopy is used for trend analysis of manufacturing processes to support the development of an automated control system. In the food industry, the measuring of protein or fat concentration in beverages like milk or cereals is performed, or the level of moisture in wheat, barley and other grain materials inflicting their quality are tested. Also, the determination of alcohol content in beer, wine and other alcoholic beverages, and additionally to alcohol content, tests for specific amino acids, acidity and total sugar were performed.

The pharmaceutical industry is increasingly using NIR for their in-process routines to analyse the drying and granulating process. Further, tablets are tested for their hardness and conformity as well as powder blending on their grain sizes. Beside the mostly mentioned qualitative measurement performed with NIR spectroscopy, also quantitative analysis can be executed. Especially the agricultural industry examines the composition of the foul and recovered organic solvents. The analysis of the composition of soil, the uptake of nitrogen and the quality control of insecticides are primarily using NIR methods. [4, 21]

While all these applications of NIR spectroscopy in diverse industrial fields have been established over many decades, the use of NIR spectroscopy in bioprocesses offers the potential of (near) real time monitoring. The different stages of where NIR spectroscopy can be (or is already) used in bioprocesses can be seen in Figure 3.4. The proven use in fermentations at lab scale (Figure 3.4, 2) arises from the fact that fermentations must be divided into two major subgroups regarding monitoring by NIR spectroscopy.
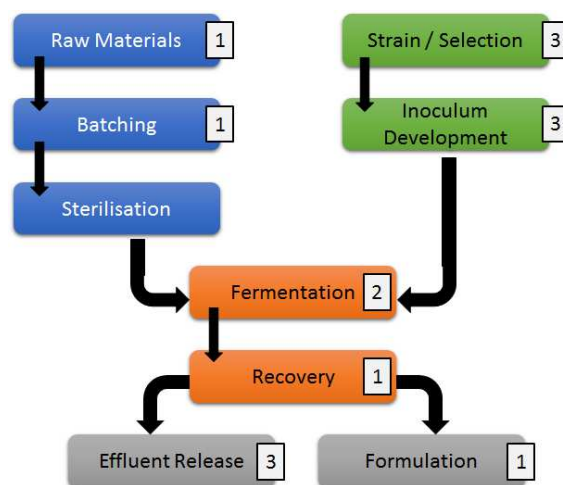
Figure 3.4 – Areas of NIR spectroscopy application in bioprocesses; adapted from [4]
(1- commonly used; 2- utility proven at lab scale; 3- utility under investigation)

Working with mammalian cells, sparsely aerated and lightly agitation is used which are considered spectroscopically simple and "clean" (following Newtonian rheological behaviour), these processes have been studied both at-line [22, 23, 24] and *in-situ* [25, 26] already.

The conditions of agitation and aeration in cell cultures using fungal fermentation or bacteria is much more vigorous [27, 28], presenting significant challenges to the use of NIR. The high cell density cultures of bacteria (such as *E.coli*) or yeasts (like *Pichia Pastoris* or *Saccharomyces cerevisiae*) present great levels of complexity. Although these different fermentations differ in their physical and chemical nature of their matrix (bioprocess fluid), NIR spectroscopy can be used to successfully monitor them (partially). [4]

3.3.3   NIR in Bioprocesses - State of the Art

Industrial processes and its variables can be monitored via different sensors and measurement methods. Those include measurements of process variables in gas and liquid phase as well as biomass and other solids. Spectroscopic sensors are just a small part of the available sensors for determining components within the liquid phase. Since 1999, a great number of studies have explored the use of NIR spectroscopy for monitoring bioprocesses as it can be useful in providing several applications, for example within automatic control systems, process optimization and on-line quality assessments. [5]

As they enable multiparametric, non-invasive measurements of process variables and trajectories, the robust free beam NIR process analysers are indispensable tools within the PAT and *quality by design* (QbD) framework. Already in 2013, the data acquired by NIR sensors could be used together with quantitative and multivariate analysis to provide information on-line. Being easy to interpret, they led to better process understanding. This made it possible for an operator to intervene earlier and improved overall performance. [29]

In [30], the advantage of real time measurements and weaker absorption in contrast to MIR are mentioned. Regarding the first advantage, "real time" can be interpreted variously in terms of process sampling used in comparison to the process variable to be measured, as some sampling methods need around 30 minutes of evaluation, but depending on the measured variable, for instance an metabolic change that can have an hour-long adjustment period, the measurement is fast enough for the purpose of process control. Nevertheless - speaking of NIR sensors - measurements are available immediately, independent of the monitored process variable . Regarding the second mentioned advantage in [30], weaker absorption bands result in lower sensitivity but higher penetration depth (than MIR) and are suitable for highly absorbing samples [31]. As NIR is used invasively and non-invasively in pharmaceutical industry - for different measurements such as dissolution rates, blending homogeneity, water content during drying -, it is becoming an effective tool for the PAT initiative in pharmaceutical processing. Due to the fact that bioprocesses most often consist of aqueous solutions that change their dilution and contain varying cell concentrations, bioprocesses are much harder to monitor via NIR than chemical drug manufacturing. [30]

However, NIR spectroscopy is able to cover a wide spectrum of upstream applications in biotechnology. Several studies monitored biomass concentrations of different microorganisms - amongst others also *S.cerevisiae* in 2007 [32] [3], resulting in limited information on biomass if the media is very turbid. Despite its many possibilities of application, NIR spectroscopy is not routinely used in upstream and downstream processes. Only few studies show results of using spectroscopic NIR approaches with bacilli, yeasts (*P.pastoris*) or mammalian cell fermentations. [3, 33] Just few applications of NIR spectroscopy in downstream units of mammalian process can be found. [4]

As far as the NIR spectroscopy can be used to monitor both - chemical and biological - process variables in upstream units as described above and in scale-up as described in [35], one has to bear in mind that the specific NIR absorption bands of process media and biomass alone do not provide enough information. Often multivariate analysis and preprocessing is needed to extract the detailed information of interest. [36] The technology is further limited by its dependence on calibrations (derived from single reactor at given time) [6], and the immersed probes are subject to a chemical broth and can therefore be blocked during measurements by solids or air bubbles. [3]

In general, raw NIR spectra have the disadvantage that even the slightest particle influences the outcome, maybe even at more than one wavelength, which results in noise peaks. The more diffuse the measured sample, the more noise is added to the spectra and agitation and aeration in bioprocesses exacerbate these circumstances.

To prevent these noises from overlaying essential information, the aforementioned preprocessing is required.

NIR shows a great potential for being used as the main sensor technique within industrial processes. It appears that NIR is more often used in research at lab-scale than in industry. Reasons for that could be the huge amounts of documentations that are needed for implementation and the possibility that industrial applications often underlie secrecy. [37]

---

[3]For comparison of two different sensor methods (infrared and dielectric spectroscopy) within fed-batch processes at high concentration levels.

[4]Applications of freeze-drying and crystallization are described in [34].

3.3.4   Spectral Preprocessing

Signal processing is performed to transform spectral data prior to calibration and further multivariate data analysis. In modern NIR spectroscopy of complex composite samples it is quite common that effects appear which cannot be easily quantified with reference analysis. Those noises arise from the interaction of compounds, light scattering, effect on the baseline and many others. Such interference can manipulate the outcome of following calculations and model predictions. [21] Therefore, the noises have to be removed as good as possible. This signal processing is also referred to as data pretreatment and aims at increasing the signal-to-noise ratio via different methods resulting in more distinctive peaks.

Although data preprocessing is almost always necessary, it should be kept in mind that every peak in a spectrum contains information. Applying too many methods or trying too hard to remove noise from the spectrum may result in the loss of essential information about the state of the system at the time of recording.

Smoothing

The basic approach of smoothing is the reduction of noise by averaging multiple measurements. The drawback is a sub-optimal cutting of frequencies in which frequencies that include essential information may be cut off together with those being just noise. Two more advanced approaches are the *moving average method* and the *Savitzky-Golay-Filter*.

The Savitzky-Golay method fits a low-degree polynomial through the data points and can be varied for its window size (number of data points) and order of derivative, resulting in a local convolution filter. [21]

Baseline Corrections

The baseline correction occurs when systematic deviations distort the baseline. Corrections are made by using vertices, by functions as described above at the smoothing approach or by applying derivatives. A positive side effect is the enhancement of the spectral resolution by the application of the derivatives, but the original appearance of the spectrum is lost which can complicate subsequent calculations by PCA (*principal component analysis*) or PLS (*partial least squares*). [38]

Multiplicative Corrections

Although many methods on multiplicative corrections exist, such as "Norris Regression" and MSC, the *standard normal variate* (SNV) was chosen as it gives nearly equivalent results as MSC without the required additional information on the product samples. SNV calculates the mean and standard deviation of an entire spectrum. Using these quantities, the individual spectra can be centred around zero. If a correlation between the pathlength changes of the wavelength and the standard deviation exists, the method delivers multiplicative standardisation. [21, 38]

3.4 Multivariate Data Analysis

The recorded data sets are multidimensional information that cannot always be displayed, which is why multivariate data analysis is used to help re-render the data or reduce dimensions. Using a variety of methods, such as PCA, the overview of multidimensional problems can be recaptured, and any relationships of data become visible.

3.4.1 PCA

Principal component analysis has the mathematical advantage over methods such as *multiple linear regression* (MLR) that any uncorrelated data does not prevent the computation of linear models. Using PCA, multidimensional datasets can be placed on a - ideally two-dimensional - "lower" dimensional base. The factors that have the greatest impact (biggest variance) on the data are used to create new coordinate systems based on these so-called *principal components* (PCs). How this realignment of the data takes place so that maximum information can be obtained in the new representation depends on the direction of the maximum scatter of the data plotted on the new orthogonal axes. In the two-dimensional case, a system (diagram) of two axes is created, but it does not always make sense to display the data in two dimensions, since otherwise too much potentially important information could be lost. Regardless of how "n-dimensional" the new axis system should be, the following procedure is generally used: The dataset searches for a direction that describes the maximum scatter of the dataset. The first axis is placed in these. In the next step, a new axis which is orthogonal to the first axis, describing the maximum scatter in its direction, is put into the system. For the two-dimensional case, the resulting system would be the final result. For multidimensional systems, finding the new axes, assuming that they are always orthogonal to the previous axis and describe the maximum scatter in their direction, is repeated until the result satisfies the requirements or the maximum number of dimensions (depending on the data set) is reached. Due to the orthogonality of the axes to each other, the correlation between any two components is always zero. [38, 39]

Chapter 4   -   Goals and Scientific Questions

The aim of this diploma thesis was to provide a basis for the modelling of yeast batch fermentations (*S.cerevisiae*), with the requirement that this modelling is based on spectral process data and can be combined with process monitoring systems.

The fermentations performed during this study shall be used to establish and develop workflows at the laboratory. For this purpose, batch fermentations were carried out at lab-scale whose structure should be as identical as possible to the fermentations already carried out by the research group in 2016.

During the processes performed in 2016, the spectral data was recorded but was not used for further analysis or modelling. By comparing the new and the historic fermentations, it should be shown that already existing data can be used for modelling too, insofar as they are physiological similar to the new ones and reproducible.

1st Scientific Question

Can spectral data capture deviations in metabolic states in yeast at low biomass concentrations?

The historical fermentations were carried out at biomass concentration levels of $\leq 2.5\,g/L$ which could make it difficult to obtain adequate spectral data of the organism's metabolic states. Process-specific circumstances can lead to different behaviours of the yeasts in the solution which can have a tremendous impact on a final product in an industrial process.

2nd Scientific Question

Are the deviations distinguishable from each other within PCA analysis?

Being able to monitor a change in the metabolic state, the operator would be able to intervene at an early stage of the process which is not performing at an ideal level. If one is further able to differentiate between a disturbance of the pH and the oxygen supply, or even between different pH values (too basic or too acidic), the regulation back to the optimum would be easier.

Chapter 5 - Material and Methods

The historic as well as the new fermentations were part of a DoE (*design of experiment*) including fermentations that were disturbed during process, by shift of pH and aeration, and ones that were performed in as optimal conditions as possible (so called "Golden Batches (GB)"; hereinafter "standard fermentation"). Those disturbances were chosen because of their possibility to occur in industrial processes if one or more control mechanisms stopped working.

The disturbances on the pH included two modes: *base off* and *base on*. Both modes shall resemble the failure of tubing between reactor and base, emptying of stock solution and/or failure of pH-probe, resulting in insufficient to no base addition (*base off* mode) or extensive addition of base to suboptimal conditions (*base on* mode). In the *base off* mode fermentation, the base supply was terminated after 120 minutes of process. In the *base on* mode fermentation, the pH control system was shut down after 120 minutes, resulting in continuous addition of base towards pH-values beyond 10.

The disturbance on the aeration rate shall resemble a failure in oxygen supply resulting in a change from respiration to fermentation in the metabolic system of the yeast. In the *air off* mode, the air supply was shut down after 120 minutes. Respectively to their individual disturbance, those fermentations are referred to as "base off-", "base on-" and "air off-" fermentations.

The DoE of the historic fermentations included 10 standard and 5 disturbed fermentations. While the standard fermentations were performed similarly to the new standard fermentations, the disturbances were chosen differently. The historic disturbances included a *base off* fermentation[1], an *air off* fermentation, a fermentation with low stirring speed, an addition of $CuSO_4$[2] during fermentation and a fermentation with more biomass in the initial setup of the fermentation.

---

[1]There was no *base on* fermentation performed. Therefore, only *base off* fermentation results will be compared with the historic data set.

[2]Copper-Sulfate shall simulate a yeast-toxic substance in the medium. Because yeasts are used for the production of enzymes and other products, in a process with genetically modified yeasts, they may produce and release a substance into the fermenter medium that can have a poor or toxic effect on the yeast.

5.1 Culture Medium and Chemicals

For all fermentations, a uniform medium with nutrients was weighed in to create an optimal environment for the microorganism at the beginning of the process. In Table 5.1, in the leftmost column, one can find the predetermined target amount of media components to be weighed, which were each dissolved in 4.5 L tap water. The columns to the right of it contain the weights for the corresponding fermentations, indicated by their process numbers.

Table 5.1 – Weighted amount of the ingredients of medium, substrate (glucose) and biomass for the six standard fermentations (F295 - F299)

Chemicals	Target-Amount [g]	Actual-Amount [g]				
		F-295	F-296	F-297	F-298	F-299
$(NH_4)_2SO_4$	72.00	71.935	71.988	72.001	72.026	72.018
$MgSO_4 \cdot 7H_2O$	5.20	5.203	5.198	5.197	5.209	5.198
$(NH_4)H_2PO_4$	5.60	5.616	5.601	5.594	5.604	5.590
KCl	22.50	22.502	22.496	22.501	22.498	22.502
$CaCl_2 \cdot 2H_2O$	1.89	1.882	1.886	1.893	1.894	1.885
Glucose	9.50	9.508	9.526	9.502	9.494	9.495
Yeast	10.00	10.000	10.000	10.014	9.814	10.028

After the preparation of the medium, the reactor was assembled (the installed instruments are listed in chapter 9.1.2). The stirring was set to 800 rpm and 11 ml of trace element solution were added. For this, a single volume of 200 mL was prepared (stock solution) and 11 mL were taken for each fermentation. The composition for the trace element stock solution[3] is shown in Table 5.2.

After the trace element solution, 2 drops of the antifoam agent *Struktol J673* were added. This agent should prevent too much foaming, since the foam could cause an inhomogeneous broth.

After the medium in the reactor was completely assembled, the recording was started in the operating system. After around two minutes, the initial biomass was added to the fermenter. The exact amount of yeast for each fermentation can also be found in Table 5.1. The fermentations were monitored by the operator via process control system and terminated after indication of gassing factors (CO_2 and O_2).

---

[3]The data in Table 5.2 refers to a final volume of 50 mL.

Table 5.2 – Ingredients of the trace elements stock solution

Chemicals	Target-Amount [g]	Concentration [mg/L]
$CuSO_4 \cdot 5H_2O$	48	2.4
$MnSO_4 \cdot H_2O$	210	11.6
$ZnSO_4 \cdot 7H_2O$	180	9.9
Ca-D(+)-pantothenat	200	11.0
Myo-inositol	2100	115.5
Biotin	20	1.1
$FeCl_3 \cdot 6H_2O$	300	16.5

5.2 Processing of the Data

During the fermentations, the data from all instruments and the set process parameters were set up via *SIEMENS PCS7* and the data was collected electronically via the process analytical software *SIEMENS SIPAT*. In addition to electronic monitoring, samples were drawn off-line at different time points within the processes.

All process parameters of the historic and new fermentation were set as identical as possible, the same applies to the used instruments[4], chemicals[5] and processing operations of the off-line samples[6]. For the processing of the on-line data, a more identical preparation could be performed. The following sections shall explain the process of data collection and analysis of off-line (5.2.1) and on-line data (5.2.3).

---

[4]In 2016 other gassing probes were used to measure and collect the aeration data of CO_2 and O_2. Fortunately, there exists data in which the processes were monitored by the "old" and the "new" aeration probes simultaneously. Those data sets were used to convert the historic aeration data into values similar to the "new" instruments, making it possible to compare the data sets.

[5]As described in the section regarding the yeast strain (see 9.1.1), no statement can be made on strain consistency.

[6]In 2016 no measurements regarding the supernatant were performed in any kind.

5.2.1   Off-line Data Analysis - Collection and Pretreatment

At different points in time during the bioprocesses, small volumes (at around 20 mL per sample) of broth were collected via flexible tubing. Six test tubes were filled with 3 mL of broth solution respectively. Those tubes were centrifuged, and the supernatant was collected prior to further preparation of the cell pellets. This supernatant was frozen and stored at a temperature of $-4\,°C$ until transportation for HPLC measurements. The cell pellets were washed with water and centrifuged again. The newly supernatant was withdrawn and the cell pellets were put in the heating chamber at a temperature of $65\,°C$ for at least three days[7]. The dried samples were weighted and their dry cell weight (DCW) was used for further evaluation of the physiological data.

HPLC Measurement

The supernatant samples were transported frozen and thawed just before the HPLC measurement. Prior to the measurements, standard solutions of ethanol and pyruvic acid were made. These standard solutions and water (blank test) were analysed simultaneously to the measurement of the samples. The sugar concentrations were analysed via HPLC (Thermo Scientifc) on a Supelcogel column (Supelco Inc.) at a constant flow of 0.5 ml/min at $30\,°C$. The mobile phase consisted of 0.1% H_3PO_4 (85%) and ethanol and pyruvic acid were detected with a Shodex RI-101 refractive index detector (DataApex).[8] [40]

The supernatant was especially analysed for the amount of ethanol in the broth at the specific time stamp, but also pyruvic acid (and possible other by-products) of the yeast's individual metabolism.

---

[7]Due to limitations in laboratory accessibility on weekends, the dried samples could not always be further processed and analysed after exactly 72 hours. All samples were dried at least 72 hours.

[8]Analysis of the chromatograms was performed by Johanna Hausjell, MSc using *Chromeleon Software* (Dionex).

5.2.2   Offline Data Analysis - Physiological Data

The results from the DCW and the HPLC measurements were inserted into an *.xlsx*-file.
Six parameters were analysed to get an insight into the metabolic state of the yeast as good
as possible and to be able to compare the historic and new fermentations on a physiological
level.

Respiratory Quotient

Dividing the metabolism of *S.cerevisiae* into anaerobic fermentation and aerobic respir-
ation, the respiratory quotient, describing the ratio between CO_2 and O_2 in the gassing
rates, is an important tool to determine the metabolic state. Monitoring the CO_2- and
O_2-values in the off-gas, this factor was also used to determine the end of fermentation
during process.

C-Balance

As glucose is the only carbon source (in the beginning) the distribution of the C-atoms
can be used to determine the metabolic state. The C-Balance estimation is used to
check robustness of experimental information.

Volumetric Growth Rate of Biomass r_X

As substrate is metabolised, biomass grows in addition to self-sustainment. This growth
rate is defined by the increase of biomass over time and expressed by r_X.

Volumetric Substrate Uptake Rate r_S

The substrate is metabolised for self-sustainment and growth. It is defined by the
uptake of substrate over time and expressed by r_S.

$\frac{X}{S}$ Yield

The amount of substrate used for the formation of new biomass can be calculated using
the $\frac{X}{S}$ yield ($Y_{\frac{X}{S}}$). If the necessary substrate consumption for cellular maintenance is
neglected (3.2.1), the $\frac{X}{S}$ yield corresponds to the ratio between the formed biomass and
the substrate uptake.

$\frac{CO_2}{S}$ Yield

The $\frac{CO_2}{S}$ yield ($Y_{\frac{CO_2}{S}}$) describes the ratio of produced CO_2 regarding to the substrate
uptake.

5.2.3   On-line Data Analysis - Collection and Transfer

The collected data was transferred in **.csv*-data format from the process control unit to data storage. Due to the fact that the process control software stores all data equally, the order of rows and columns were not changed in order to perform each processing step identically for each data file. The data was loaded into *R*. There was no treatment done within the files prior to processing with *R*.

5.2.4   Online Data Analysis - Preprocessing of the Data

The code used for processing the spectral data (see 9.2) consists of three major parts.
In the first one, the data is inserted into the program and unified. The further process steps after this part can be further divided into individual and collective treatment. The second part deals with the smoothing and correction of the baseline. The third part deals with the statistical analysis of the data.

Unifying of the Data

Prior to processing the data with statistical methods, the spectral data was cut from the whole data set. The spectral data was plotted in 3D plots for first optical examination. It was found that the data points next to the limits of the wavelength range (1350nm and 1650nm) were extremely noisy. Failure of the connection of the NIR probe and its software to the process monitoring software led to the fact that in the beginning of each fermentation some time stamps had collected no spectral data but only process data. The noises and partially missing data points in the beginning led to the cutting of the data set. Respectively 10nm of wavelength range from the limits and the first 45 minutes of each fermentation were omitted. The data was then examined for missing NIR data during the process and the corresponding entries were removed from the record. Thus, it was given that all data sets did not contain missing spectral values. After this step, the spectral data were identical in size (number of variables), except for the time length. The individual wavelengths (281) are found as variables in the columns, the absorptions recorded every minute as objects in the rows of the data matrix.

Smoothing and Baseline Correction

Taking the data from the previous preprocesses, a standard normal variate method (SNV) was carried out at first. This was done to correct the baseline row-wise. The data sets were reduced from noisy peaks by smoothing. For this, a Savitzky-Golay (SG) polynomial fit was applied. The window size was 13, the polynomial order was 1 and the first derivative was calculated. The Savitzky-Golay fit was applied on the data that has already been smoothed via SNV. Prior to further analysis the results of unifying, SNV- and SG-processing were analysed optically via 3D-plots. These initial checks by visual inspection were used to investigate the effects on the functions for correctness. Likewise, the best settings for the SG-filter (polynomial order, window size and derivation) could be found. This entire process was carried out twice: once just with the data from the first processing step and once with the data from the first processing step with additional application of the "MeanInt" function. The self-written function "MeanInt" should serve to form the arithmetic mean over the data, in the interval $+i$ and $-i$ around the sample x. The sample x was taken every 20 minutes and the interval i was set to 7. The two resulting data sets were further analyzed by PCA.

5.2.5   On-line Data Analysis - Principal Component Analysis

There were two analyses performed via principal component analysis (PCA). In the first place, on each fermentation data a PCA was performed individually. This was done to determine the biggest variance for each fermentation and to look for possible outliers. For the second PCA analysis an additional column was inserted including the number of fermentations in each row for the respective fermentation data. The resulting data sets were bound together row-wise, resulting in one big data set. This part of analysis was done so that the axis of rotation is the same for the largest variance in each process. The results and conclusions on those analyses can be found in the respective chapters (Chapter 6.4.3 and Chapter 7).

Chapter 6 - Results and Discussion

In the following sections (6.1-6.3), the physiological data, required from off-line samples, and the spectral on-line data of the process were compared and analysed in accordance with their specific fermentations (standard vs. disturbance). The fermentation data sets were analysed using the six parameters mentioned in chapter 5.2.2. The fermentations were correlated with each other, divided according to their fermentation types (e.g. historical standard with new standard or specific historical disturbance with corresponding new disturbance). Furthermore, the metabolic states of the new fermentations were compared regarding the six parameters and if their spectral fingerprint can give an insight into the metabolic state. In the diagrams, the error bars represent the standard deviation and the historic data is represented by dotted bars. The colour code shall help to distinguish between processes. The supernatant analyses via HPLC were not included in the data analysis since for the metabolite ethanol as well as for pyruvic acid, no concentrations above the analysis limit could be determined. Since it was assumed that ethanol and pyruvic acid were the most frequently present metabolites in terms of quantity, and even no ethanol was found above analysis limit in the supernatant of the anaerobic fermentation, possible further metabolites in the broth were neglected.

6.1 Standard Fermentations

From the 10 historically performed standard fermentations, 5 were chosen to be analysed in comparison to the new standard fermentations regarding their spectral data. From those 5, 2 were compared physiologically with the 2 new standard fermentations due to deviations in the historic physiological data. Overall, an error of under 18% could be achieved for all six parameters (Table 6.1), thus the reproduction of the historic fermentations could be achieved quite well. Comparing the specific growth rate (μ [h^{-1}]) in Figure 6.1 and the volumetric growth rate of biomass r_X in Figure 6.5, the difference is at around 5% in between the historic (F134, F137) and new (F295, F296) standard fermentations respectively.

**Table 6.1** – Differences between the standard fermentations

|	Growth Rate μ	C-Balance	$\frac{CO_2}{S}$ Yield	$\frac{X}{S}$ Yield	Volumetric Growth Rate r_X	Substrate Uptake Rate r_S
	[1/h]	[−]	[C − mol/C − mol]	[C − mol/C − mol]	[C − mmol/h]	[C − mmol/h]
Average	0.01	0.71	0.32	0.39	21.30	-55.22
Standard Deviation	0.00	0.11	0.06	0.06	1.08	8.08
Difference [%]	5.11	15.92	17.17	15.04	5.08	14.63
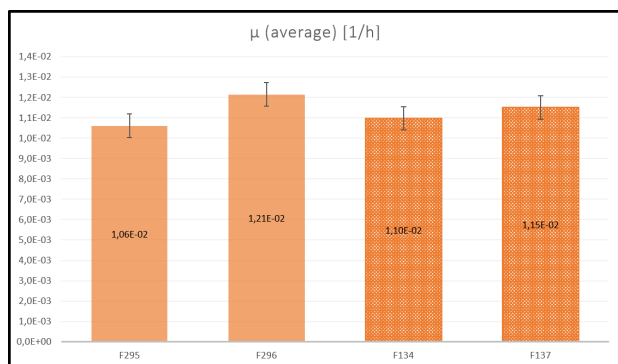
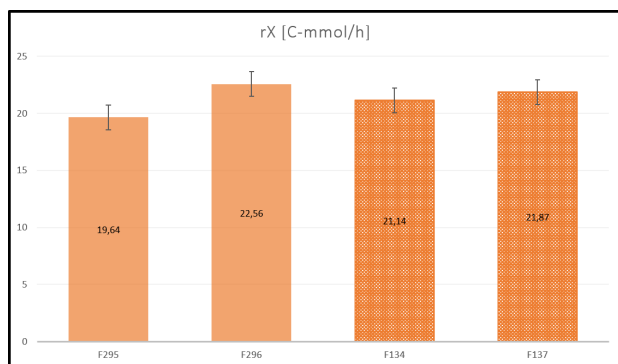**Figure 6.1** –
Comparison of the growth rate μ
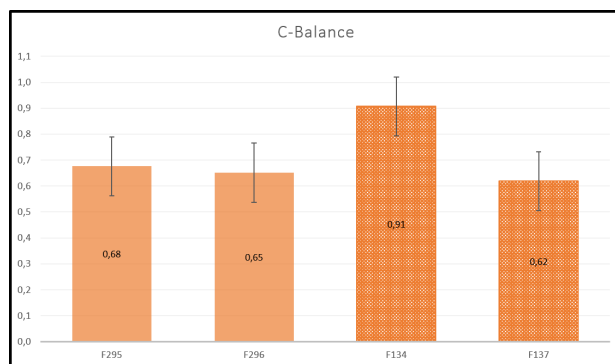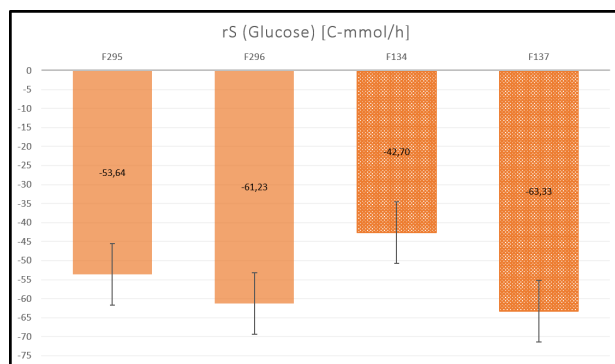

Figure 6.2 –
Comparison of the C-balance
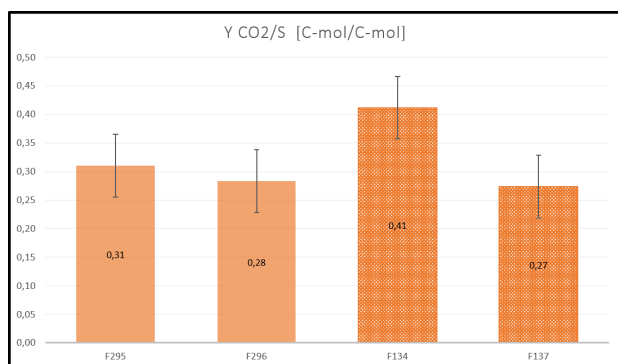
Figure 6.3 –
Comparison of the carbondioxide yield
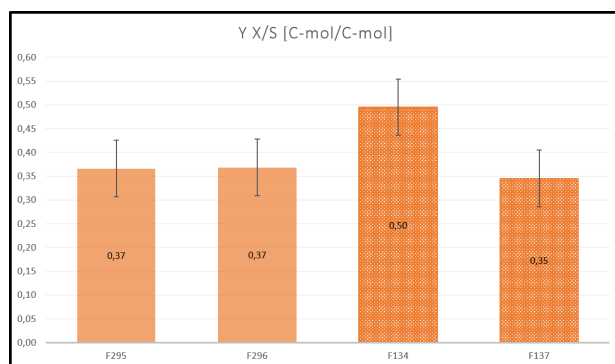
Figure 6.4 –
Comparison of the biomass yield

Figure 6.5 –
Comparison of the volumetric growth rate of
biomass r_X

Figure 6.6 –
Comparison of the volumetric substrate uptake
rate r_S

6.2   *Base* Disturbance Fermentations

In Figures 6.7-6.12, the fermentations disturbed via missing or extensive base input (green bars, right-hand side) can be seen next to the standard fermentations (orange bars, left-hand side). Regardless of which parameter is considered, even though the fermentations were disrupted by the addition of base, they reach (almost) the same level as the ideally performed fermentations (F295, F296).

Since yeast cells must be able to adjust to the most diverse environmental conditions, a pH value that deviates from the optimal range does not automatically lead to cell death. Specifically, the enzyme $H^+\text{-}ATPase$, along with several systems, help to regulate the internal pH to balance out external stress. [41] To achieve this, the yeast cells in parallel must adjust their metabolism to the new environmental conditions. The cells can thus survive under these conditions and continue to process substrate, yet much of it is used on survival itself or on the metabolic change. This leads to a declining growth rate or even to a complete halt (regarding growth). Experiments showed that a pH in the strong acidic [42] and basic range [43] inhibits the metabolism of the yeast in terms of metabolic adjustments but is not necessarily lethal.

Furthermore, in experiments with a minimal substrate and biomass approach, the performed increase in base addition cannot be directly assumed to be the cause of the end of the fermentation[1]. In addition, the storage bottle of the base was empty just before the termination of fermentation[2].

Although the two disturbances have a similar level in terms of C-balance and volumetric substrate uptake rate r_S, bigger differences can be seen in specific growth rate μ, the $\frac{CO_2}{S}$ *Yield* and the volumetric growth rate of biomass r_X. As the specific and volumetric growth rate are related, it explains their similar appearance. The lower values of *base off* (light green) compared to *base on* (dark green) could be explained by the carbon dioxide yield. It is known in literature [44, 45] that high concentrations of dissolved CO_2 have an inversely proportional effect on the growth rate of yeasts. Likewise in *S.cerevisiae* an inhibition by CO_2 could be determined. [46]

The increased uptake of carbon dioxide in the *base off* fermentation is most likely related to the metabolism of yeast that releases hydrogen protons into the medium when substrates are broken down. If the base that balances these protons is missing, it results in an acidic medium which may increase the interaction between the gas and the solution. The error bars represent the standard deviations in between the separate groups of fermentations (see also Table 6.2).

---

[1] Regarding the experiments in this thesis, the substrates were metabolised long before the pH levels hit a critical level concerning survival, in addition to the pH and the base consumption, the *off-gas* curves of O_2 and CO_2 were observed, which were used as indicators for the end of the metabolic phases.
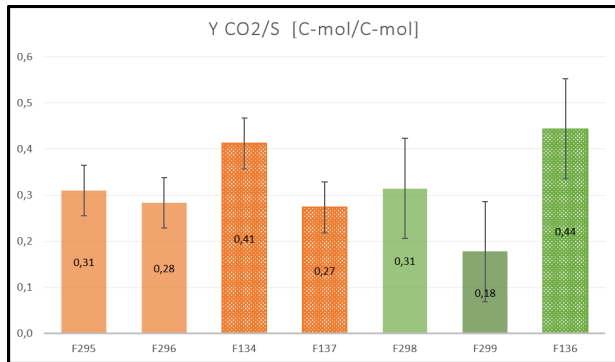
[2] A replacement to a new storage bottle was not possible because of the equipment provided, as this second "spare bottle" could not have been monitored by the process control system (calibration of the scales).
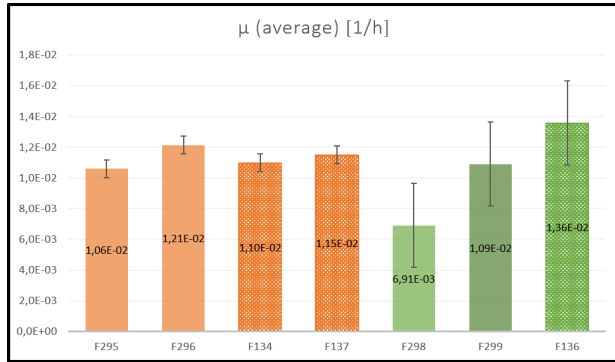
Figure 6.7 – *Base* disturbances: Comparison of the growth rate μ
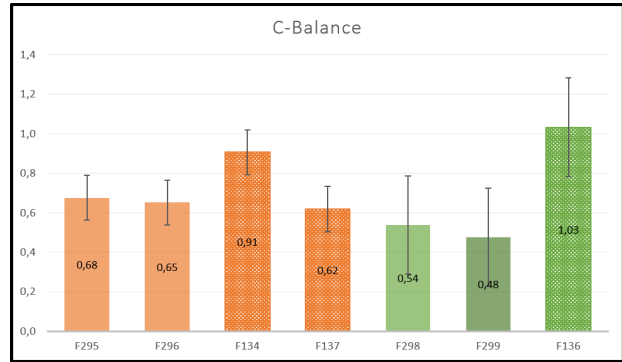
Figure 6.8 – *Base* disturbances: Comparison of the C-balance

Figure 6.9 – *Base* disturbances: Comparison of the carbondioxide yield

Figure 6.10 – *Base* disturbances: Comparison of the biomass yield

Figure 6.11 – *Base* disturbances: Comparison of the volumetric growth rate of biomass r_X

Figure 6.12 – *Base* disturbances: Comparison of the volumetric substrate uptake rate r_S

<div align="center">

Table 6.2 – Differences between the fermentations disturbed by base

</div>

	Growth Rate μ	C-Balance	$\frac{CO_2}{S}$ Yield	$\frac{X}{S}$ Yield	Volumetric Growth Rate r_X	Substrate Uptake Rate r_S
	$[1/h]$	$[-]$	$[C-mol/C-mol]$	$[C-mol/C-mol]$	$[C-mmol/h]$	$[C-mmol/h]$
Average	0.01	0.68	0.31	0.37	19.74	-56.35
Standard Deviation	0.00	0.25	0.11	0.16	5.56	9.51
Difference [%]	26.21	36.51	34.76	42.54	28.18	16.87

6.3 *Aeration* Disturbance Fermentations

While disturbances in base addition influence the yeast's operational rates, however, the yeast-preferred metabolic pathway of substrate degradation does not change fundamentally. By switching off the fumigation, the yeast changes to an anaerobic metabolism (reductive pathway), whereby the produced alcohol can no longer be processed (see 3.2.1).

Thus, the processing of a second substrate at the end of the glucose supply is not possible and the ethanol remains unprocessed in the reactor medium. This change in raw materials can be seen in Figures 6.13-6.18, where the values for the six parameters for the disturbed *air off*-fermentation (Table 6.3) are way below the values of the standards and the *base* fermentations.

<div align="center">

Table 6.3 – Differences between the fermentations disturbed by aeration

</div>

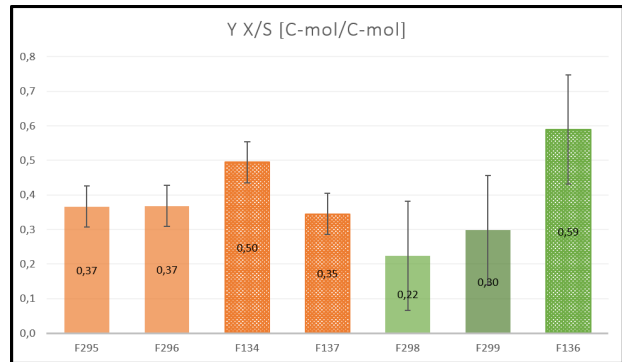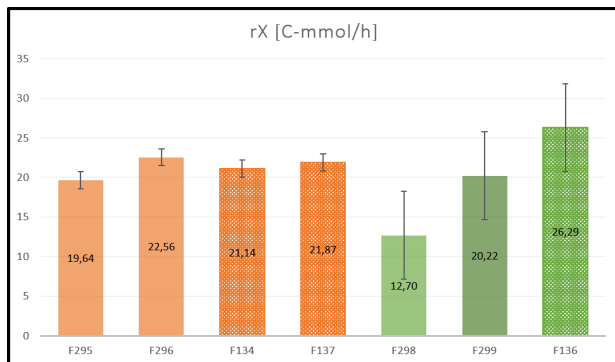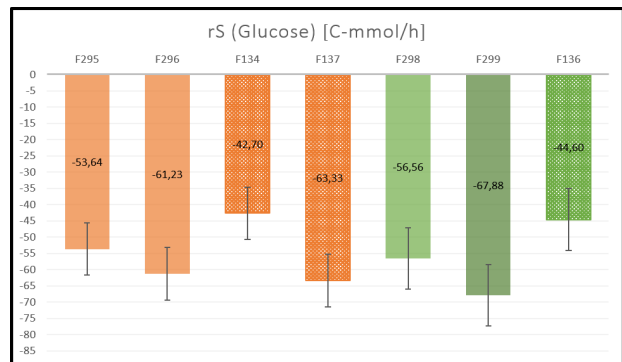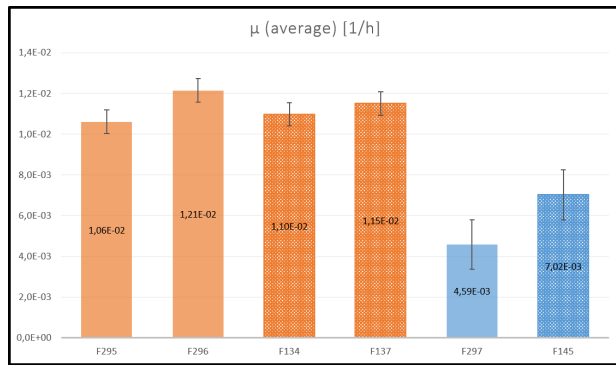|	Growth Rate μ	C-Balance	$\frac{CO_2}{S}$ Yield	$\frac{X}{S}$ Yield	Volumetric Growth Rate r_X	Substrate Uptake Rate r_S
	$[1/h]$	$[-]$	$[C-mol/C-mol]$	$[C-mol/C-mol]$	$[C-mmol/h]$	$[C-mmol/h]$
Average	0.01	0.45	0.248	0.21	10.73	-57.92
Standard Deviation	0.00	0.21	0.12	0.09	2.38	12.68
Difference [%]	21.00	46.22	49.68	42.01	22.16	21.90

Figure 6.13 – *Air off* disturbance: Comparison of the growth rate μ
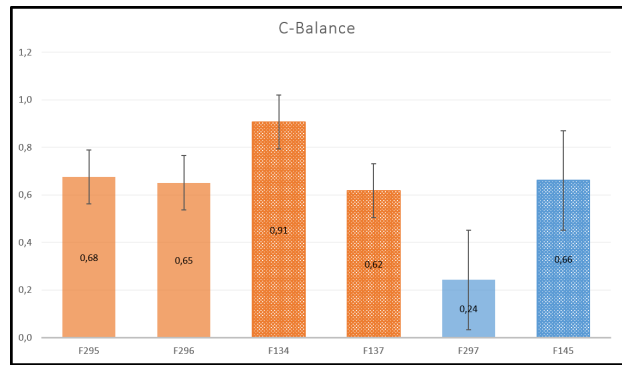
Figure 6.14 – *Air off* disturbance: Comparison of the C-balance
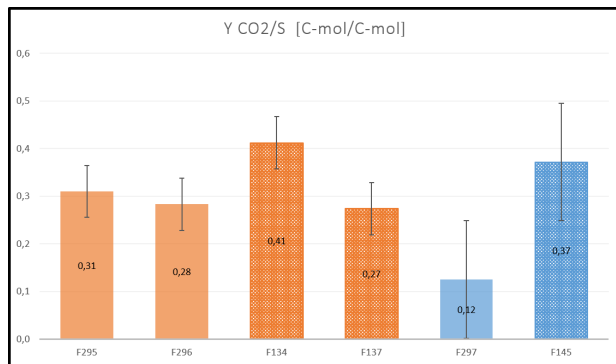
Figure 6.15 – *Air off* disturbance: Comparison of the carbondioxide yield
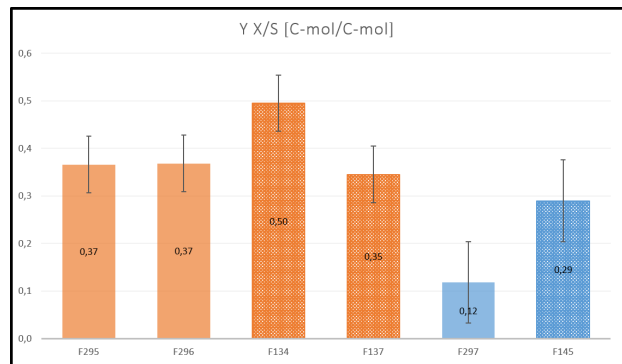
Figure 6.16 – *Air off* disturbance: Comparison of the biomass yield
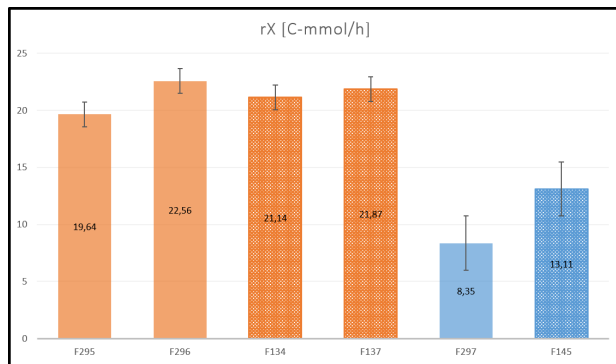
Figure 6.17 – *Air off* disturbance: Comparison of the volumetric growth rate of biomass r_X

Figure 6.18 – *Air off* disturbance: Comparison of the volumetric substrate uptake rate r_S

6.4 Evaluation of PCA Results

Via the software R, the spectral data was analysed. After the preprocessing and unifying of the data (see 5.2.3 *ff.*), the PCA results of the individual analysis and the linked data set were analysed. Those are shown in section 6.4.1 and section 6.4.2 respectively.

6.4.1 PCA - Individual Analysis

The new standard fermentations (Figures 6.19 and 6.20) had a steady time course of the data points over the variance of the second principal component. In t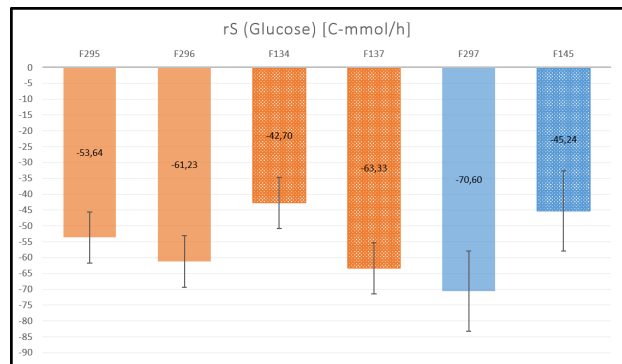he pictures, one can see a starting point in the lower right corner (dark blue), which moves into the upper right corner (light blue). The Figures 6.19-6.27 each have a dark data point in the left area. After cutting away the first 45 minutes in each record, this is the first data point to occur. Since the respective disturbances took place after 120 minutes, this data point was left in the figures to show that similar initial values were obtained at the beginning of each of the fermentations and that the trends are well visible. (All of the figures were also analysed with numbers representing sample time in order to prove the change over time.) This course of the data points is exactly the same in the data of the historical fermentations (Figures 6.21 and 6.22), although the percentage of total variance of first principal component varies significantly.
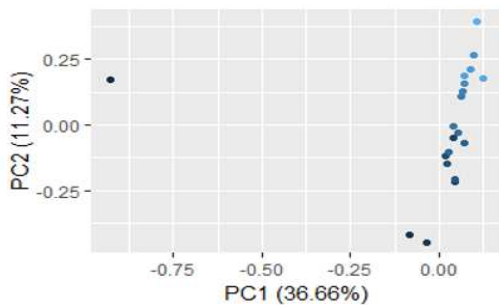
Figure 6.19 – New Standard - F295
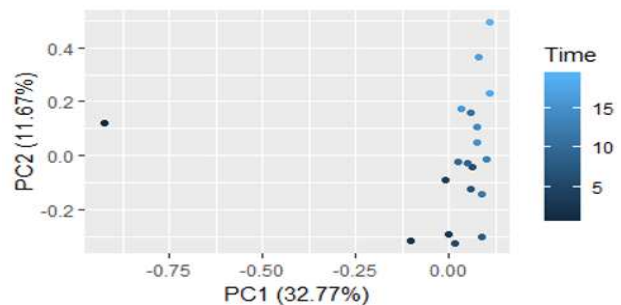
Figure 6.20 – New Standard - F296
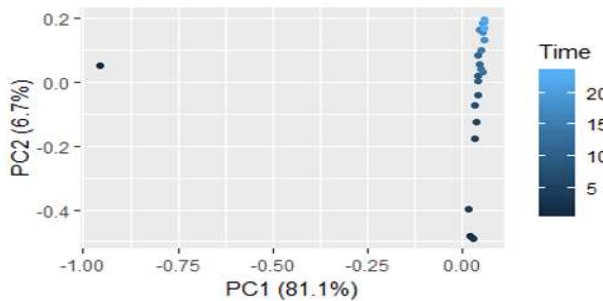
Figure 6.21 – Historic Standard - F134
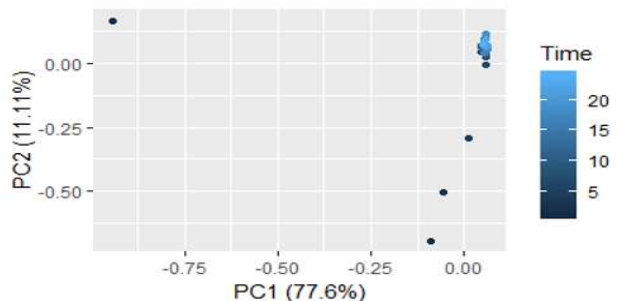
Figure 6.22 – Historic Standard - F137

Base Disturbance

By looking at the fermentations that are disturbed by the base addition, their PCA results of the spectral data differ from the standards. At the same time, they look similar to each other (Figure 6.23 and 6.24). This may be due to the aforementioned differences in metabolism. As with the physiological data, they reach values similar to the standard fermentations, but not the same. The disturbed fermentations also start in the lower right corner and run into the upper right corner. There, however, the end is not reached as in the standards, but they move along the same axis back to their previous starting point. This course of the data points for the *base* fermentation is similar in the data of the historical fermentation (Figures 6.25)[3]. The changing of data over time also occurs along the second principal component and although the direction is inverted, the increasing and decreasing behaviour appears to be identically.
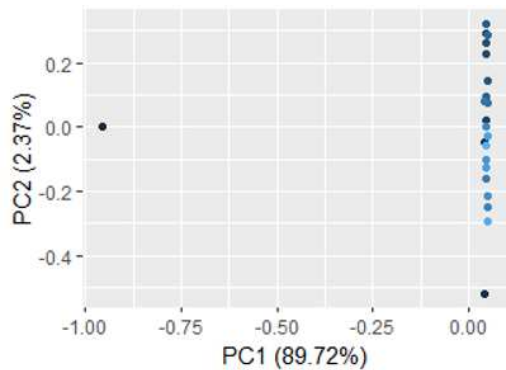
Figure 6.23 –
New *Base off* Fermentation - F298
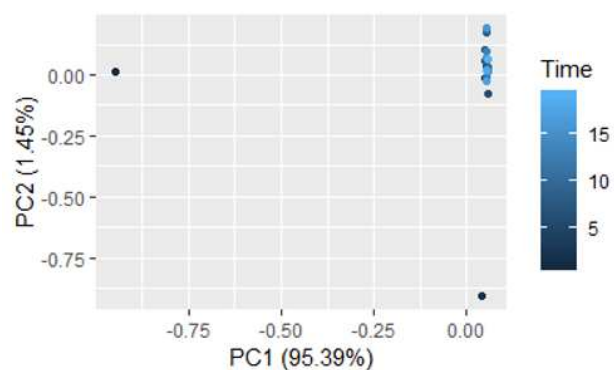
Figure 6.24 –
New *Base on* Fermentation - F299
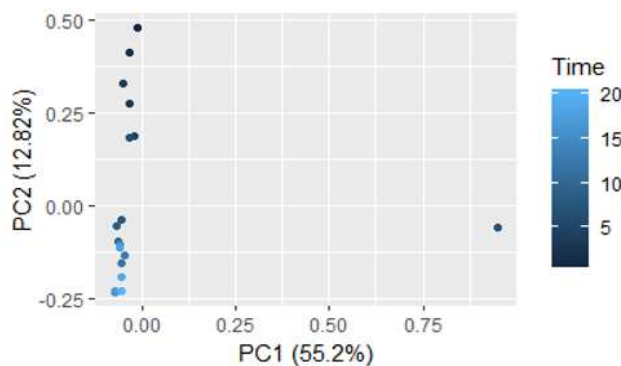
Figure 6.25 –
Historic *Base off* Fermentation - F136

---

[3]There does not exist a recording as to when exactly the base supply was cut off in the historic fermentation which makes a comparison considerably more difficult and can be the reason for deviations.

Aeration Disturbance

Since a change in the aeration is associated with a change in metabolism, which is fundamentally different from the standard and base fermentations, also the results of the PCA analysis look significantly different than the previous ones (Figure 6.26). As already shown with the physiological data, this difference has an effect on the yeasts and can also be seen by the evaluation of the spectral data. The data points begin at a completely different data value for the first principal component (PC1) and then develop along its axis rather than the axis of the second principal component (PC2), as in the previous figures.
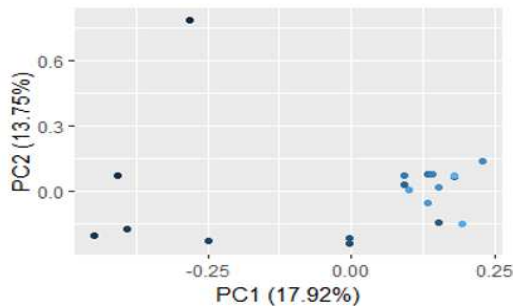
Figure 6.26 –
New *Air off* Fermentation - F297
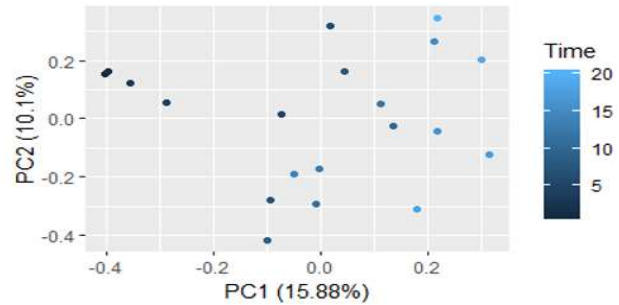
Figure 6.27 –
Historic *Air off* Fermentation - F145

32

6.4.2   PCA - Linked Analysis (New)

The data points aligned to the largest variance in PCA may differ significantly in their direction, depending on the origin of the data. In the previous chapter it was, for example, shown that the data from standard fermentations deviate from the disturbed ones. For the alignment of the largest variance to refer to the same properties of the data points, the next two figures show the results of this calculation (Figure 6.28 and 6.29).

As one can see in the representation by point form in Figure 6.28, all different fermentations are in a different area of the diagram. The two standard fermentations (yellow and brown) overlap each other, which represents an identical spectral recording. Each disturbed fermentation is in a section of the diagram that does not overlap with the other disturbances. Only the *air off* fermentation (green) shows a small area where it overlaps with the standard fermentations.
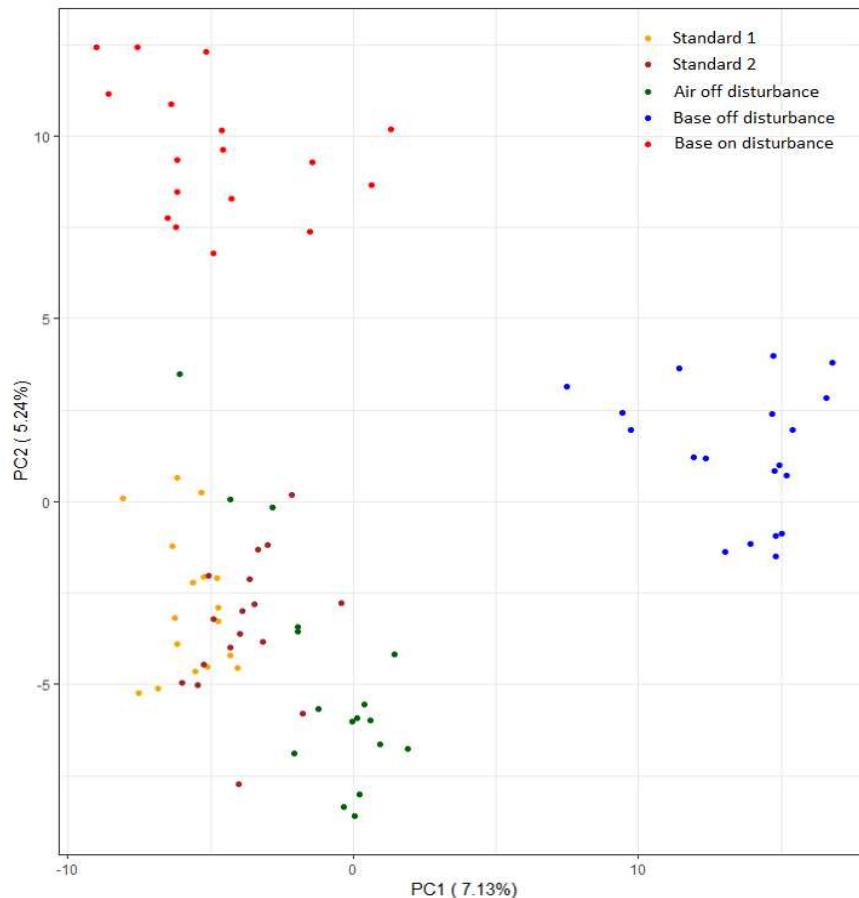
Figure 6.28 – Comparison of the distribution of the new fermentations regarding PCA

Using Figure 6.29, the timing of each sample shall be represented. The numbers after the letter "X" in the figure indicate the order of the data points for the respective fermentation. It can be seen that the *air off* fermentation starts with values above the standard fermentations, passes through its "cloud of data points" during the process and ends in an agglomeration below the standard fermentations. While in the "cloud of data points" in the *base off* fermentation a circular formation over time can be seen, the *base on* data points have no significant tendency into a specific direction.
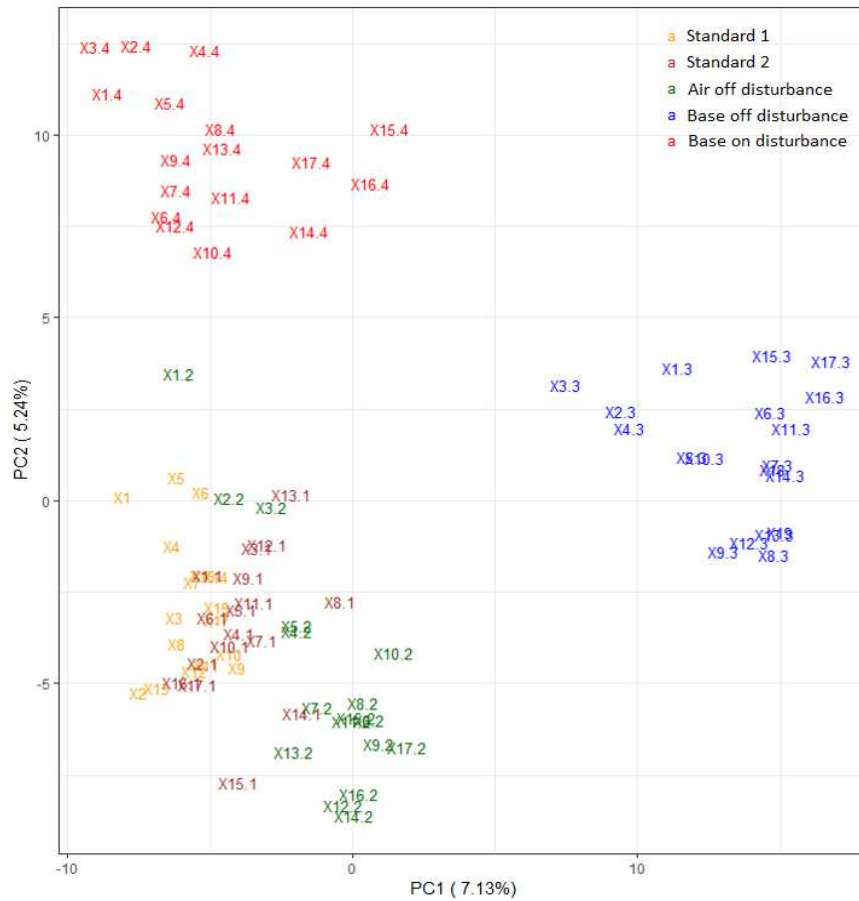
Figure 6.29 – Comparison of the timely distribution of the new fermentations regarding PCA

6.4.3   PCA - Linked Analysis (Historic)

The analysis and presentation of the historical fermentations was carried out analogously to Figure 6.28. It can be seen that the two standard fermentations are clearly separated and that the data points of the base disturbed fermentation are between the two standards.[4] The significance of the PCA in regard of the historical fermentations should therefore be viewed critically. The proportions of the total variance for both, PC1 and PC2, are higher than those of the new fermentations in Figure 6.28.
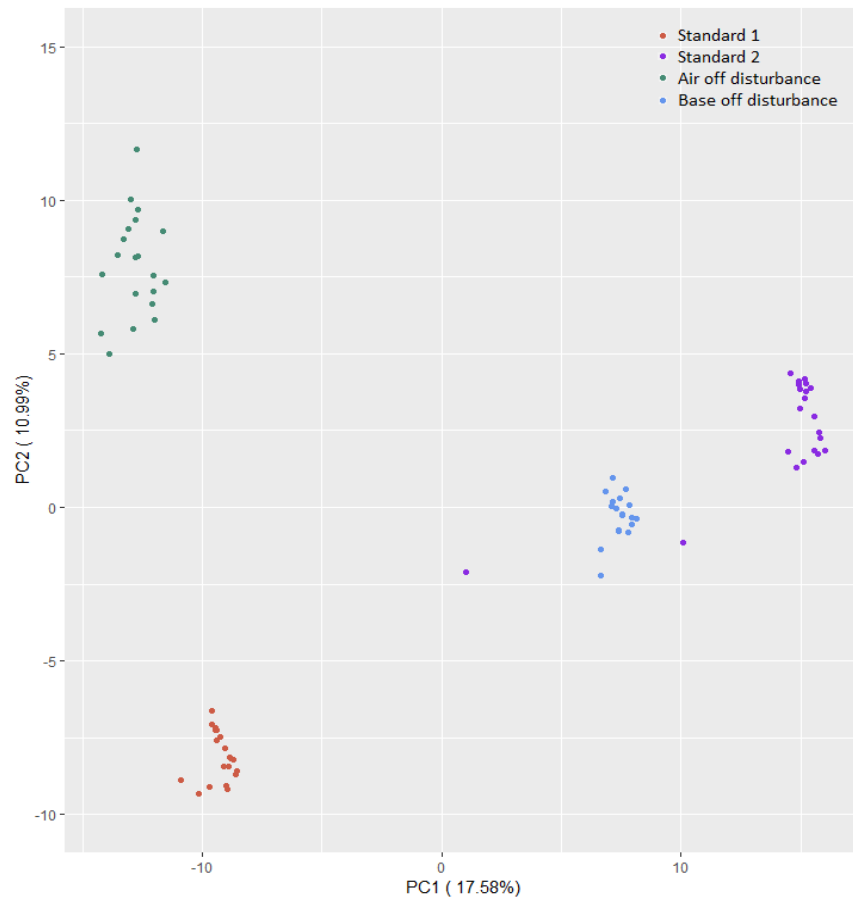
Figure 6.30 – Comparison of the distribution of the historic fermentations regarding PCA

---

[4]In order to be able to display the data points as good as possible, one data point of fermentation F137 (Standard 2) was removed (coordinates at about 10/-20).

In order to be able to compare the informational value of the spectral data of the analysis, a PCA analysis was carried out over the entire data (new and historic fermentations).

The data points of the new and historical fermentations are shown in Figure 6.31a and 6.31b. The new fermentations (F295 - F299) overlap on the left in those illustrations. The historical fermentations show a similar distribution as in Figure 6.30 and it can be seen that the axis - on which the two standard and the base disturbed fermentation are distributed on - is also present in this figure (although slightly turned). The air disturbed fermentation (F145, yellow) lies vertically above the standard fermentation F134 (rose). This becomes even clearer when looking at Figure 6.31b, where the PC1 is plotted against the PC3. The two fermentations, prior lying vertically one above the other, now overlap and in turn - again - form an axis with the other two historical fermentations. This leads to a further reason for the assumption that a better classification of the fermentations could be made on higher PCs. This assumption was checked up to the fifth principal component, but no improvement was shown. It can also be seen in Figure 6.31b that the "cloud of data points" of the new fermentations is now divided as the *base-off* fermentation lies "above" the others. Its data points are now at around the same values of PC3 as the historical *base-off* fermentation.
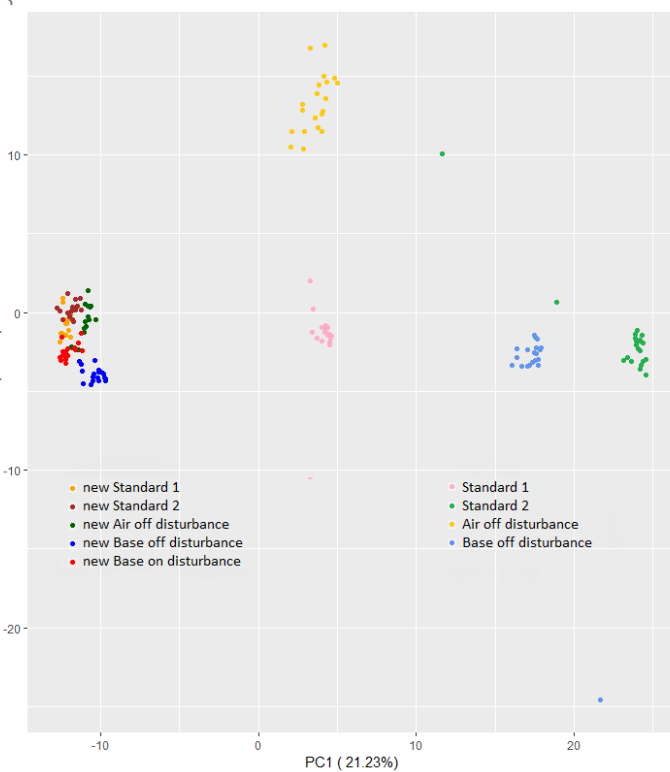
(a) Comparison of the new and historic fermentations regarding PC1 and PC2
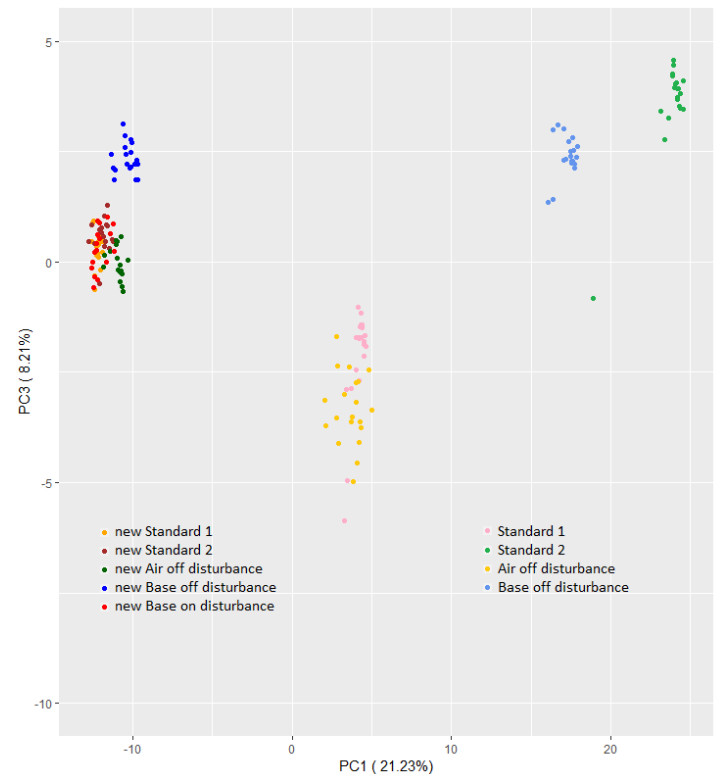
(b) Comparison of the new and historic fermentations regarding PC1 and PC3

Figure 6.31 – Results of the PCA analysis with linked fermentations (new and historic)

Chapter 7 - Conclusion

Offline Data

The physiological data - on which the comparison of the new with the historical data was based - showed great similarity between the individual fermentations (standard vs. standard, base vs. base, air vs. air). In particular, the standard fermentations showed great similarity when compared to their historical counterparts. The growth rate for instance showed a difference of just over five percent. As already described in the text above (see 6.1), only two of the ten fermentations carried out in 2016 were compared with the new standards. This was due to the standard fermentations containing certain temporal and operational fluctuations in their data. During the preparation of the spectral data of these fermentations, large outliers in the middle of the fermentations were found which is why only two fermentations were ultimately used as a comparison with the new ones.

With the fermentations disturbed by the base, it could be shown that a disturbance in the pH value seems to have only marginal effects on the metabolism, when considering the total batch time. Slightly different values could be found in the fermentations. In order to be able to make more precise statements for these analyses, however, further fermentations under the same conditions are necessary. Unfortunately, no fermentation was carried out in 2016 in which the base was added uncontrolled. As a result, no exact comparison could be made, which means that the standard deviations of the base fermentations are significantly higher than those of the standard fermentations.

The fermentations in which the air supply was cut off showed the difference in metabolism more clearly. The large deviations of the new fermentations from the historical fermentations are mainly due to the changing equipment with the different oxygen and exhaust gas sensors. Here, too, further fermentations would have to be carried out, so that a tendency can be inferred.

On-line Analysis

Individual PCA Analysis

When processing and analysing the spectral data with the software, similarities to the off-line data could be determined. The results of the PCA of the standard fermentations shown in Figure 6.19 - 6.22 are very similar both to each other and with their historical counterparts. Time dependent behaviour can be seen in all fermentations and a distinction can be made between the individual variants of the implementation. Nevertheless, the fermentation data was processed individually in this analysis and not on a combined level. Even if the same trends can be identified, they can occur differently (in their variance), as can also be seen in the base fermentations (see Figure 6.23 - 6.25).

Linked PCA Analysis

The analysis using PCA with simultaneous calculation of the variance shows that a clear distinction can be made between the new fermentations in the first two principal components (see Figure 6.28). The standard fermentations overlap, which can be seen as an argument for reproducibility and at the same time they clearly differentiate from the disturbances.

The historical fermentations can also be clearly distinguished, but the standards are not as close together to each other as the new ones. It can only be assumed what this lack of reproducibility of the historical standards can be attributed to. It is possible that there are clear differences between these two fermentations for PCA even if these were not recognized during the initial selection for physiological consideration and when the spectral data sets were first analysed. With regard to the distribution of the data points of the historical fermentations, it should be noted that:

i) When considering the first and third principal components, the two *base-off* fermentations are at approximately the same value for the third principal component.

ii) When comparing the first and third principal components, the four historic fermentations lie on one axis, which can be traced back to an origin that has not yet been found.

1st Scientific Question

Can spectral data capture deviations in metabolic states in yeast in low concentrations?

The spectral data associated with the physiological data showed a distinction between the two metabolic states of yeast (anaerobic / aerobic). It was also possible to differentiate between high and low pH of the medium using spectral data. Thus, yeast disorders are also recognizable, which are associated with a change in metabolism, but not due to changes from aerobic to anaerobic, but from an optimal pH value environment to suboptimal conditions. The monitoring of metabolic state could be useful while inoculating during scale up from a prior small volume (e.g. shaking flasks). Especially in early stages the yeast should be transferred within its exponential growth rate to adjust best to new surrounding media as the majority of the existing enzymes in the yeasts are used for growth. If the state of the microorganisms can be defined, the right time for transfer could be chosen more accurately. Although industrial processes work with a much higher cell densities, part of this study was to establish and develop workflows within the laboratory the study was performed. Additionally, the historic data had to be reconstructed as similar as possible and therefore the processes were conducted with lower cell density levels.

2$^{\text{nd}}$ Scientific Question

Are the deviations distinguishable from each other within PCA analysis?

Comparing new and historic fermentation data, no exact statement can be given. The historic data lets one differentiate between optimum and disturbance, but the standard fermentations do not overlap. If the standards are not reproducible, the significance of the deviation of the disturbances must be viewed critically. This deviation of standards may be due to the conversion of the old data, as this fermentation was carried out with other devices (see 5.2). In general, there are not enough measurements for the disturbed fermentations to test how they can be reproduced with each other. It can be shown by the combined PCA that the new fermentations and the historical ones differ greatly in terms of the first three principal components. Likewise, no reproduction between the individual counterparts (standard vs. standard, base vs. base, air vs. air) can be shown.

Nevertheless, the new standard fermentations are comparable with each other. The data points of the historic standard fermentations are "shifted" locally, but they have a very similar appearance. An analysis of the time dependency of the data points also showed a very large agreement with a counterclockwise course during the fermentation (see Figure 7.1) which can be seen when magnifying the two standards from Figure 6.30. The physiological comparison led to an error of maximum 18% in the standard fermentations based on the six criteria tested. Depending on spectral data, the deviations are much bigger and need more data to see a trend for reproducibility.
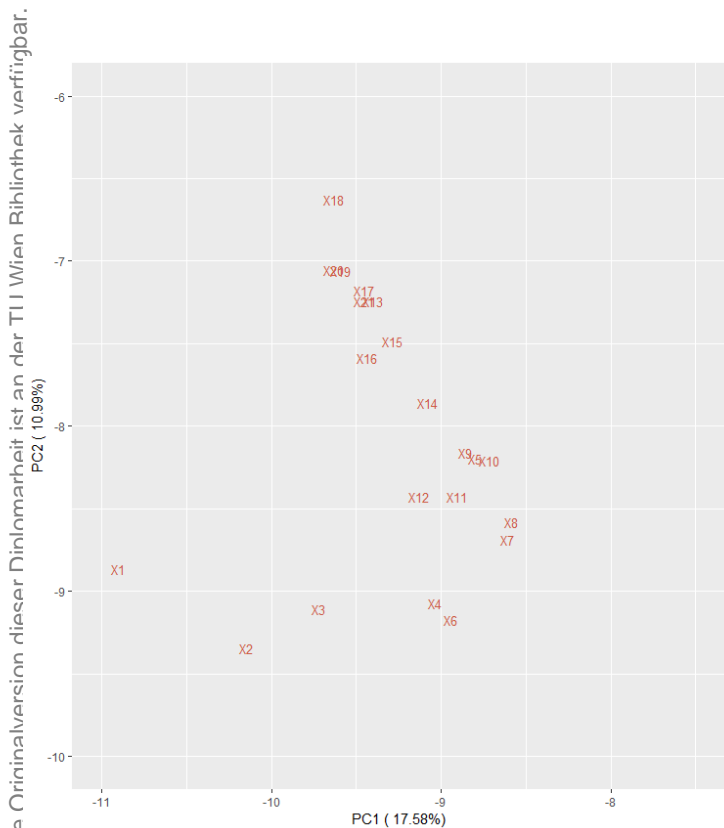
(a) Zoom on the data of F134
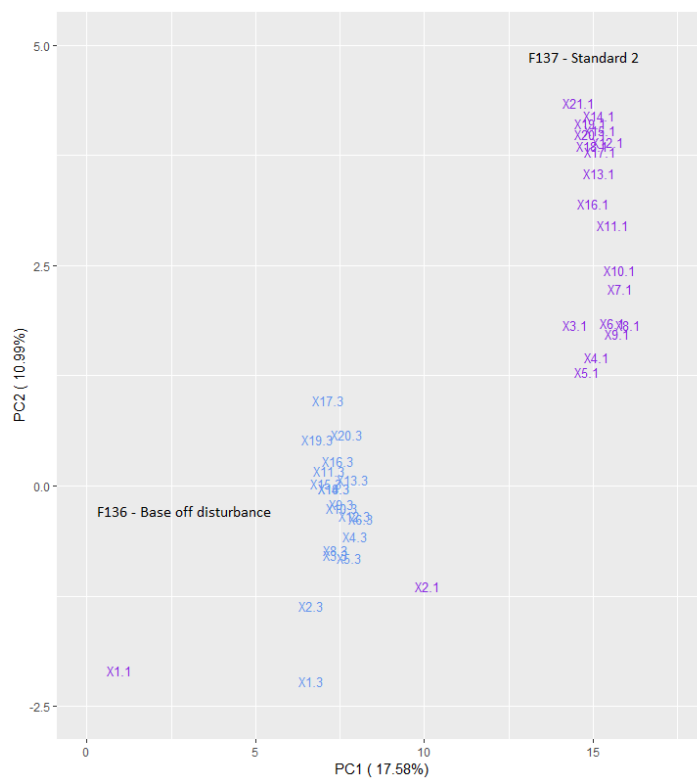
(b) Zoom on the data of F137
(The data points of F136 can be neglected)

Figure 7.1 – Time dependent behaviour of the historic standard fermentations

Chapter 8 - Outlook

The new fermentations showed a good approach to record the metabolic states of yeast in a bioreactor using spectral data. The individual runs could be differentiated from each other and the off-line measurements showed results that supported the on-line records in relation to the metabolism states. The next big step has to be a further implementation of fermentations of the same structure so that more data is available, which can be compared to the already known data and results. With more data, not only can previous assumptions be confirmed with greater accuracy, but also a better basis for modelling is formed.

The creation of a model on the basis of the data obtained could lead to early detection of errors that are discovered through on-line monitoring. This would improve the response time to unexpected disturbances. The model could thus be expanded by additional statistical methods such as PLS or Hotelling T^2, which enables the data to be examined more closely. Furthermore, the off-line data (sampling) was taken at large intervals during the processes (about every 2 hours). The time intervals between sampling could be reduced to check whether the time trend could be improved. If this is the case, one would have an even finer tool for finding deviations in a minimal period of time. In addition to larger amounts of data and increased methods of statistics to investigate them, operational improvements could also be made.

During handling with the light guide, which was installed via the lid, deviations in the spectra could be determined, which were due to the winding path of the light guide to the reactor, which could not be reproduced exactly from fermentation to fermentation. The use of a reactor that would allow the light guide to be installed horizontally from the side would be one possibility of change. It would be questionable whether the same measurement results would be obtained at minimal biomass concentrations through the glass wall or whether the absorption of / transmission through the glass is a problem.

In addition to collecting data regarding batch fermentations, records of fed-batch processes would also have to be planned in order to detect any changes due to the addition of substrate/s. Fed-batch processes are also used more frequently in industry than batch processes, which means that modelling could be applied to a wider range of processes.

Chapter 9 - Appendix

9.1 List of Equipment

9.1.1 Chemicals and Yeast

Chemicals
All chemicals used for media and stock solutions are found in Table 5.1 and 5.2.
The antifoam agent used was *Struktol J673*.
$NaOH$ was used as base to control the pH and in the *base-on* fermentation to disturb the optimum. All chemicals were purchased via *LACTAN* in highest quality.

Yeast Strain
The microorganism used in this thesis is a yeast strain of *Saccharomyces cerevisiae* that is used (and cultivated) by *"HAGOLD Hefe GmbH"*, part of *"LALLEMAND Inc."*. Since the information on the yeast strain is subject to industrial secrecy, no statement can be made for the exact determination of the strain. For further calculation a biomass of $26.9\,\text{g/C-mol}$ (with a composition of $CH_{1.769}N_{0.146}O_{0.631}$) has been used according to *Lange et al.* [47].

9.1.2   Bioreactor and Instruments

The instruments used during the processes are listed in Table 9.1 and the placement can be seen in Figure 9.1.

(a) Cover of the reactor (frontal view)

(b) Cover of the reactor (top view)
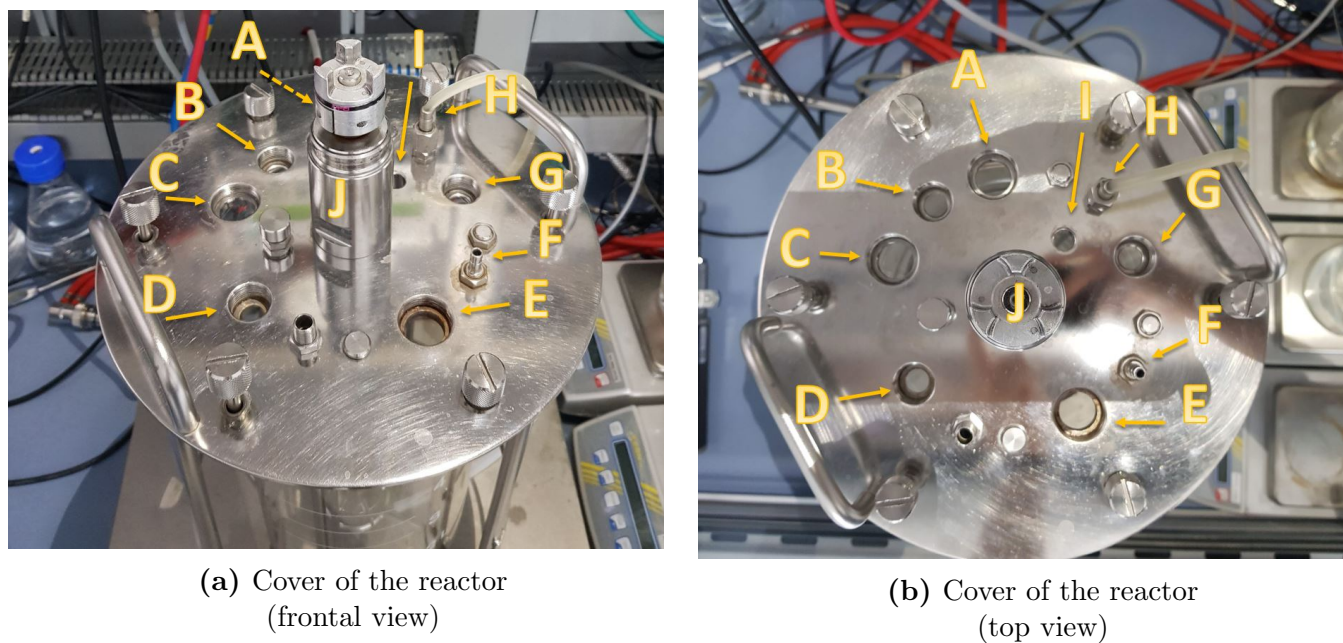
Figure 9.1 – Mounting positions of the devices on the bioreactor cover

Table 9.1 – Type designations and description of the connected devices

Position	Description	Type Designation
A	cooling finger	/
B	pO2 probe to monitor the oxygen partial pressure in the medium	Hamilton VisiFerm DO Arc 325
C	off-gas cooler with outlet for excess gas of the fermentation broth	/
D	Hamilton-Incyte probe for in-line determination of biomass	Hamilton Pre-Amp 243720
E	connection for 4-way access (addition of acid, base and/or feed)	/
F	ventilation hose for introducing air into the medium	/
G	pH probe that measures both pH and temperature	Hamilton EasyFerm HB Arc 225
H	tube for taking samples	/
I	inlet for the NIR probe	StellarNet BLACK-Comet CXR-SR-25 spectrometer
J	motor of the stirrer	/

9.2 R Code

Please note that the colour settings for graphic illustrations in code functions may differ from the actual used ones within this thesis as they have been changed in the illustrations afterwards due to better visibility in printed versions.

```r
#--- Directory und Libraries------
#setwd("C:/mydirectory") # change for the actual directory path
library(ggplot2)
library(plot3D)
library(Hmisc)
library(corrplot)
library(gplots)
library(tidyr)
library(ggfortify)
library(GGEBiplots)
library("FactoMineR", lib.loc="~/R/win-library/3.5")
library(factoextra)
library(lattice)
library(qpcR)
library(plotly)
library(rgl)
library(dplyr)
library(magick)
library(corrplot)
library(gridExtra)
library(prospectr)
library(spectacles)
library(plyr)
library(zoo)
library(grid)
library(gridExtra)
library(RColorBrewer)
#display.brewer.all()
library(gghighlight)
library(reshape)
library(gganimate)

#--- --- --- --- -----
#--- Functions -------

# MeanInt takes every x-th samle from data plus an interval of +i and -i
# around that sample and calculates the arithmetic mean
MeanInt = function(data, x, i){
                              data = na.omit(data)

                              dataframe = data.frame(matrix(nrow=0, ncol=
    ncol(data)))
                              colnames(dataframe) = colnames(data)
```

```r
                                    for (j in x){
                                           ctr = 0

                                           offsetmin = j - i
                                           offsetmax = j + i

                                           if (offsetmin < 0) {
                                             offsetmin = 0
                                           }
                                           if (offsetmax >= nrow(data)) {
                                             offsetmax = nrow(data)-1
                                           }

                                           y = data[(offsetmin):(offsetmax)
   ,]
                                           z = colMeans(y, na.rm = TRUE)

                                           dataframe[nrow(dataframe)+1,] =
   z

                                           ctr = ctr + 1
                                    }
                             return(dataframe)
                        }

#--- --- --- --- -----
#--- Reading raw data of the new fermentations and selecting spectral data
     from process monitoring -----

setwd("C:/Users/Markus/Desktop/Diplomarbeit_Programme/R/Hefe_Batch_
   historisch/Standard")
# historical fermentations (2016)
## Standards
NIR_134 = read.csv("Standard_SIPAT_F134_20160920_Batch.csv", header=TRUE,
    dec = ",", sep =";")
 NIR_134 = select(NIR_134,"NIR.Spectrum_1350.0000000":"NIR.Spectrum_
   1650.0000000")
 colnames(NIR_134) <- c(paste0(1350:1650))

NIR_137 = read.csv("Standard_SIPAT_F137_20160927_Batch.csv", header=TRUE,
    dec = ",", sep =";")
 NIR_137 = select(NIR_137,"NIR.Spectrum_1350.0000000":"NIR.Spectrum_
   1650.0000000")
 colnames(NIR_137) <- c(paste0(1350:1650))

setwd("C:/Users/Markus/Desktop/Diplomarbeit_Programme/R/Hefe_Batch_
   historisch/Stoerungen")
## Disturbances
### Air off Disturbance
```

```r
84  NIR_145 = read.csv("Stoerung_SIPAT_F145_20161013_Batch.csv", header=TRUE,
        dec = ",", sep =";")
85    NIR_145 = select(NIR_145,"NIR.Spectrum_1350.0000000":"NIR.Spectrum_
        1650.0000000")
86    colnames(NIR_145) <- c(paste0(1350:1650))
87  ### Base off Disturbance
88  NIR_136 = read.csv("Stoerung_SIPAT_F136_20160926_Batch.csv", header=TRUE,
        dec = ",", sep =";")
89    NIR_136 = select(NIR_136,"NIR.Spectrum_1350.0000000":"NIR.Spectrum_
        1650.0000000")
90    colnames(NIR_136) <- c(paste0(1350:1650))
91
92
93
94    # new fermentations (2019)
95    setwd("C:/Users/Markus/Desktop/Diplomarbeit_Programme/R/Hefe_Batch_
        reprod")
96    #Standards (Golden Batch 1 and 2)
97    NIR_295 = read.csv("F295_goldenB_1.csv", header=TRUE, dec = ",", sep =";
        ")
98    NIR_295 = select(NIR_295,"NIR.Spectrum_1350.0000000":"NIR.Spectrum_
        1650.0000000")
99    colnames(NIR_295) <- c(paste0(1350:1650))
100
101   NIR_296 = read.csv("F296_goldenB_2.csv", header=TRUE, dec = ",", sep =";
        ")
102   NIR_296 = select(NIR_296,"NIR.Spectrum_1350.0000000":"NIR.Spectrum_
        1650.0000000")
103   colnames(NIR_296) <- c(paste0(1350:1650))
104   # Air off Disturbance
105   NIR_297 = read.csv("F297_Luft-aus.csv", header=TRUE, dec = ",", sep =";"
        )
106   NIR_297 = select(NIR_297,"NIR.Spectrum_1350.0000000":"NIR.Spectrum_
        1650.0000000")
107   colnames(NIR_297) <- c(paste0(1350:1650))
108   # Base off Disturbance
109   NIR_298 = read.csv("F298_Base-aus.csv", header=TRUE, dec = ",", sep =";"
        )
110   NIR_298 = select(NIR_298,"NIR.Spectrum_1350.0000000":"NIR.Spectrum_
        1650.0000000")
111   colnames(NIR_298) <- c(paste0(1350:1650))
112   # Base on Disturbance
113   NIR_299 = read.csv("F299_Base-zuviel.csv", header=TRUE, dec = ",", sep =
        ";")
114   NIR_299 = select(NIR_299,"NIR.Spectrum_1350.0000000":"NIR.Spectrum_
        1650.0000000")
115   colnames(NIR_299) <- c(paste0(1350:1650))
116
117 #--- --- --- --- -----
118 #--- Remove rows including one or more NA objects -----
119  # historical fermentations
```

```r
120  NIR_134 = NIR_134[rowSums(is.na(NIR_134)) == 0,]
121  NIR_137 = NIR_137[rowSums(is.na(NIR_137)) == 0,]
122  NIR_145 = NIR_145[rowSums(is.na(NIR_145)) == 0,]
123  NIR_136 = NIR_136[rowSums(is.na(NIR_136)) == 0,]
124  # new fermentations
125  NIR_295 = NIR_295[rowSums(is.na(NIR_295)) == 0,]
126  NIR_296 = NIR_296[rowSums(is.na(NIR_296)) == 0,]
127  NIR_297 = NIR_297[rowSums(is.na(NIR_297)) == 0,]
128  NIR_298 = NIR_298[rowSums(is.na(NIR_298)) == 0,]
129  NIR_299 = NIR_299[rowSums(is.na(NIR_299)) == 0,]
130
131  #--- --- --- --- -----
132  #--- Changing the dataframes to matrices and cutting of noise and first 45
          Minutes -----
133  # historical
134  NIR_134 = as.matrix(NIR_134)
135    NIR_134 = NIR_134[45:nrow(NIR_134),10:295]
136    t_134 = seq(1,length = nrow(NIR_134))      # time
137      t_134 = as.matrix(t_134)
138      colnames(t_134) = "Time"
139    w_134 = as.numeric(colnames(NIR_134))      # wavelength
140
141  NIR_137 = as.matrix(NIR_137)
142    NIR_137 = NIR_137[45:nrow(NIR_137),10:295]
143    t_137 = seq(1,length = nrow(NIR_137))      # time
144      t_137 = as.matrix(t_137)
145      colnames(t_137) = "Time"
146    w_137 = as.numeric(colnames(NIR_137))      # wavelength
147
148  NIR_145 = as.matrix(NIR_145)
149    NIR_145 = NIR_145[45:nrow(NIR_145),10:295
150                      ]
151    t_145 = seq(1,length = nrow(NIR_145))      # time
152      t_145 = as.matrix(t_145)
153      colnames(t_145) = "Time"
154    w_145 = seq(1350,length = ncol(NIR_145))   # wavelength
155
156  NIR_136 = as.matrix(NIR_136)
157    NIR_136 = NIR_136[45:nrow(NIR_136),10:295
158                      ]
159    t_136 = seq(1,length = nrow(NIR_136))      # time
160      t_136 = as.matrix(t_136)
161      colnames(t_136) = "Time"
162    w_136 = seq(1350,length = ncol(NIR_136))   # wavelength
163
164
165    # new
166  NIR_295 = as.matrix(NIR_295)
167     NIR_295 = NIR_295[45:nrow(NIR_295),10:295]
168     t_295 = seq(1,length = nrow(NIR_295))      # time
169       t_295 = as.matrix(t_295)
```

```
170        colnames(t_295) = "Time"
171      w_295 = as.numeric(colnames(NIR_295))     # wavelength
172
173   NIR_296 = as.matrix(NIR_296)
174     NIR_296 = NIR_296[45:nrow(NIR_296),10:295]
175     t_296 = seq(1,length = nrow(NIR_296))     # time
176        t_296 = as.matrix(t_296)
177        colnames(t_296) = "Time"
178      w_296 = as.numeric(colnames(NIR_296))     # wavelength
179
180   NIR_297 = as.matrix(NIR_297)
181     NIR_297 = NIR_297[45:nrow(NIR_297),10:295]
182     t_297 = seq(1,length = nrow(NIR_297))     # time
183        t_297 = as.matrix(t_297)
184        colnames(t_297) = "Time"
185      w_297 = as.numeric(colnames(NIR_297))     # wavelength
186
187   NIR_298 = as.matrix(NIR_298)
188     NIR_298 = NIR_298[45:nrow(NIR_298),10:295]
189     t_298 = seq(1,length = nrow(NIR_298))     # time
190        t_298 = as.matrix(t_298)
191        colnames(t_298) = "Time"
192      w_298 = as.numeric(colnames(NIR_298))     # wavelength
193
194   NIR_299 = as.matrix(NIR_299)
195     NIR_299 = NIR_299[45:nrow(NIR_299),10:295]
196     t_299 = seq(1,length = nrow(NIR_299))     # time
197        t_299 = as.matrix(t_299)
198        colnames(t_299) = "Time"
199      w_299 = as.numeric(colnames(NIR_299))     # wavelength
200
201
202 #--- --- --- -----
203 #
      ###########################################################################
204 ###### 1) Working with whole data set without creating the averaged values
        (without "MeanInt") ####
205 #
      ###########################################################################
206 #-------------------------- PREPROCESSING
      ------------------------------------------------------
207 #--- Baseline Correction via StandardNormalVariate (SNV) -----
208   # historical
209   SNV_134 = apply(X = NIR_134,MARGIN = 1,FUN = snv) # MARGIN = 1 for rows,
      = 2 for columns
210   SNV_134 = t(SNV_134)                             # transpose to create
      the axes t and w
211   t_snv_134 = seq(1,length = nrow(SNV_134))
212   t_snv_134 = as.matrix(t_snv_134)
```

48

```r
213   colnames(t_snv_134) = "Time"
214   w_snv_134 = as.numeric(colnames(SNV_134))
215
216   SNV_137 = apply(X = NIR_137,MARGIN = 1,FUN = snv) # MARGIN = 1 for rows,
        = 2 for columns
217   SNV_137 = t(SNV_137)                             # transpose to create
       the axes t and w
218   t_snv_137 = seq(1,length = nrow(SNV_137))
219   t_snv_137 = as.matrix(t_snv_137)
220   colnames(t_snv_137) = "Time"
221   w_snv_137 = as.numeric(colnames(SNV_137))
222
223   SNV_145 = apply(X = NIR_145,MARGIN = 1,FUN = snv) # MARGIN = 1 for rows,
        = 2 for columns
224   SNV_145 = t(SNV_145)                             # transpose to create
       the axes t and w
225   t_snv_145 = seq(1,length = nrow(SNV_145))
226   t_snv_145 = as.matrix(t_snv_145)
227   colnames(t_snv_145) = "Time"
228   w_snv_145 = as.numeric(colnames(SNV_145))
229
230   SNV_136 = apply(X = NIR_136,MARGIN = 1,FUN = snv) # MARGIN = 1 for rows,
        = 2 for columns
231   SNV_136 = t(SNV_136)                             # transpose to create
       the axes t and w
232   t_snv_136 = seq(1,length = nrow(SNV_136))
233   t_snv_136 = as.matrix(t_snv_136)
234   colnames(t_snv_136) = "Time"
235   w_snv_136 = as.numeric(colnames(SNV_136))
236
237
238
239   # new
240   SNV_295 = apply(X = NIR_295,MARGIN = 1,FUN = snv) # MARGIN = 1 for rows,
        = 2 for columns
241   SNV_295 = t(SNV_295)                             # transpose to create
       the axes t and w
242   t_snv_295 = seq(1,length = nrow(SNV_295))
243   t_snv_295 = as.matrix(t_snv_295)
244   colnames(t_snv_295) = "Time"
245   w_snv_295 = as.numeric(colnames(SNV_295))
246
247   SNV_296 = apply(X = NIR_296,MARGIN = 1,FUN = snv) # MARGIN = 1 for rows,
        = 2 for columns
248   SNV_296 = t(SNV_296)                             # transpose to create
       the axes t and w
249   t_snv_296 = seq(1,length = nrow(SNV_296))
250   t_snv_296 = as.matrix(t_snv_296)
251   colnames(t_snv_296) = "Time"
252   w_snv_296 = as.numeric(colnames(SNV_296))
253
```

```r
254  SNV_297 = apply(X = NIR_297,MARGIN = 1,FUN = snv) # MARGIN = 1 for rows,
       = 2 for columns
255  SNV_297 = t(SNV_297)                            # transpose to create
       the axes t and w
256  t_snv_297 = seq(1,length = nrow(SNV_297))
257  t_snv_297 = as.matrix(t_snv_297)
258  colnames(t_snv_297) = "Time"
259  w_snv_297 = as.numeric(colnames(SNV_297))
260
261  SNV_298 = apply(X = NIR_298,MARGIN = 1,FUN = snv) # MARGIN = 1 for rows,
       = 2 for columns
262  SNV_298 = t(SNV_298)                            # transpose to create
       the axes t and w
263  t_snv_298 = seq(1,length = nrow(SNV_298))
264  t_snv_298 = as.matrix(t_snv_298)
265  colnames(t_snv_298) = "Time"
266  w_snv_298 = as.numeric(colnames(SNV_298))
267
268  SNV_299 = apply(X = NIR_299,MARGIN = 1,FUN = snv) # MARGIN = 1 for rows,
       = 2 for columns
269  SNV_299 = t(SNV_299)                            # transpose to create
       the axes t and w
270  t_snv_299 = seq(1,length = nrow(SNV_299))
271  t_snv_299 = as.matrix(t_snv_299)
272  colnames(t_snv_299) = "Time"
273  w_snv_299 = as.numeric(colnames(SNV_299))
274
275
276  #--- --- --- -----
277  #--- Smoothing via savitzkyGolay (SG) -----
278   # historical
279   SG_134 = savitzkyGolay(X = SNV_134, m = 1, p = 1, w = 13)
280     ts_134 = seq(1,   length = nrow(SG_134))      # time
281       ts_134 = as.matrix(ts_134)
282       colnames(ts_134) = "Time"
283   ws_134 = as.numeric(colnames(SG_134))      # wavelength
284
285   SG_137 = savitzkyGolay(X = SNV_137, m = 1, p = 1, w = 13)
286     ts_137 = seq(1,   length = nrow(SG_137))      # time
287       ts_137 = as.matrix(ts_137)
288       colnames(ts_137) = "Time"
289   ws_137 = as.numeric(colnames(SG_137))      # wavelength
290
291   SG_145 = savitzkyGolay(X = SNV_145, m = 1, p = 1, w = 13)
292     ts_145 = seq(1,   length = nrow(SG_145))      # time
293       ts_145 = as.matrix(ts_145)
294       colnames(ts_145) = "Time"
295   ws_145 = as.numeric(colnames(SG_145))      # wavelength
296
297   SG_136 = savitzkyGolay(X = SNV_136, m = 1, p = 1, w = 13)
298     ts_136 = seq(1,   length = nrow(SG_136))      # time
```

```
299        ts_136 = as.matrix(ts_136)
300        colnames(ts_136) = "Time"
301   ws_136 = as.numeric(colnames(SG_136))       # wavelength
302
303
304   # new
305   SG_295 = savitzkyGolay(X = SNV_295, m = 1, p = 1, w = 13)
306      ts_295 = seq(1,    length = nrow(SG_295))       # time
307        ts_295 = as.matrix(ts_295)
308        colnames(ts_295) = "Time"
309     ws_295 = as.numeric(colnames(SG_295))       # wavelength
310
311   SG_296 = savitzkyGolay(X = SNV_296, m = 1, p = 1, w = 13)
312      ts_296 = seq(1,    length = nrow(SG_296))       # time
313        ts_296 = as.matrix(ts_296)
314        colnames(ts_296) = "Time"
315     ws_296 = as.numeric(colnames(SG_296))       # wavelength
316
317   SG_297 = savitzkyGolay(X = SNV_297, m = 1, p = 1, w = 13)
318      ts_297 = seq(1,    length = nrow(SG_297))       # time
319        ts_297 = as.matrix(ts_297)
320        colnames(ts_297) = "Time"
321     ws_297 = as.numeric(colnames(SG_297))       # wavelength
322
323   SG_298 = savitzkyGolay(X = SNV_298, m = 1, p = 1, w = 13)
324      ts_298 = seq(1,    length = nrow(SG_298))       # time
325        ts_298 = as.matrix(ts_298)
326        colnames(ts_298) = "Time"
327     ws_298 = as.numeric(colnames(SG_298))       # wavelength
328
329   SG_299 = savitzkyGolay(X = SNV_299, m = 1, p = 1, w = 13)
330      ts_299 = seq(1,    length = nrow(SG_299))       # time
331        ts_299 = as.matrix(ts_299)
332        colnames(ts_299) = "Time"
333     ws_299 = as.numeric(colnames(SG_299))       # wavelength
334
335
336 #--- --- --- -----
337 #-------------------------- PLOTTING OF DATA / OPTICAL ANALYSIS
        ------------------------------
338 #--- Raw data without preprocessing -----
339  t_134 = as.vector(t_134)
340  t_137 = as.vector(t_137)
341  t_145 = as.vector(t_145)
342  t_136 = as.vector(t_136)
343
344  t_295 = as.vector(t_295)
345  t_296 = as.vector(t_296)
346  t_297 = as.vector(t_297)
347  t_298 = as.vector(t_298)
348  t_299 = as.vector(t_299)
```

```
349
350  # historical
351  open3d()
352  mfrow3d(2, 2, sharedMouse = TRUE)
353  persp3d(t_134, w_134, NIR_134, col = RColorBrewer::brewer.pal(9,"Blues"),
354          #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
         +004:4.5E+004),
355          xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
         F134_GB1")
356  persp3d(t_137, w_137, NIR_137, col = RColorBrewer::brewer.pal(9,"Blues"),
357          #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
         +004:4.5E+004),
358          xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
         F137_GB2")
359  persp3d(t_145, w_145, NIR_145, col = RColorBrewer::brewer.pal(9,"Blues"),
360          #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
         +004:4.5E+004),
361          xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
         F145_Air off")
362  persp3d(t_136, w_136, NIR_136, col = RColorBrewer::brewer.pal(9,"Blues"),
363          #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
         +004:4.5E+004),
364          xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
         F136_Base off")
365
366
367  # new
368  open3d()
369  mfrow3d(3, 2, sharedMouse = TRUE)
370  persp3d(t_295, w_295, NIR_295, col = RColorBrewer::brewer.pal(9,"Blues"),
371          #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
         +004:4.5E+004),
372          xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
         F295_GB1")
373  persp3d(t_296, w_296, NIR_296, col = RColorBrewer::brewer.pal(9,"Blues"),
374          #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
         +004:4.5E+004),
375          xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
         F296_GB2")
376  persp3d(t_297, w_297, NIR_297, col = RColorBrewer::brewer.pal(9,"Blues"),
377          #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
         +004:4.5E+004),
378          xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
         F297_Air off")
379  persp3d(t_298, w_298, NIR_298, col = RColorBrewer::brewer.pal(9,"Blues"),
380          #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
         +004:4.5E+004),
381          xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
         F298_Base off")
382  persp3d(t_299, w_299, NIR_299, col = RColorBrewer::brewer.pal(9,"Blues"),
```

```r
383             #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
      +004:4.5E+004),
384             xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
      F299_Base on")
385
386
387  t_134 = as.matrix(t_134)
388  t_137 = as.matrix(t_137)
389  t_145 = as.matrix(t_145)
390  t_136 = as.matrix(t_136)
391
392  t_295 = as.matrix(t_295)
393  t_296 = as.matrix(t_296)
394  t_297 = as.matrix(t_297)
395  t_298 = as.matrix(t_298)
396  t_299 = as.matrix(t_299)
397
398  colnames(t_134) = "Time"
399  colnames(t_137) = "Time"
400  colnames(t_145) = "Time"
401  colnames(t_136) = "Time"
402
403  colnames(t_295) = "Time"
404  colnames(t_296) = "Time"
405  colnames(t_297) = "Time"
406  colnames(t_298) = "Time"
407  colnames(t_299) = "Time"
408
409
410  #--- Raw data preprocessed with SNV -----
411  t_snv_134 = as.vector(t_snv_134)
412  t_snv_137 = as.vector(t_snv_137)
413  t_snv_145 = as.vector(t_snv_145)
414  t_snv_136 = as.vector(t_snv_136)
415
416  t_snv_295 = as.vector(t_snv_295)
417  t_snv_296 = as.vector(t_snv_296)
418  t_snv_297 = as.vector(t_snv_297)
419  t_snv_298 = as.vector(t_snv_298)
420  t_snv_299 = as.vector(t_snv_299)
421
422  # historical
423  open3d()
424  mfrow3d(2, 2, sharedMouse = TRUE)
425  persp3d(t_snv_134, w_snv_134, SNV_134, col = RColorBrewer::brewer.pal(9,"
      Blues"),
426             #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
      +004:4.5E+004),
427             xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
      F134_SNV_GB1")
```

```r
persp3d(t_snv_137, w_snv_137, SNV_137, col = RColorBrewer::brewer.pal(9,"
    Blues"),
        #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
    +004:4.5E+004),
        xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
    F137_SNV_GB2")
persp3d(t_snv_145, w_snv_145, SNV_145, col = RColorBrewer::brewer.pal(9,"
    Blues"),
        #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
    +004:4.5E+004),
        xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
    F145_SNV_Air off")
persp3d(t_snv_136, w_snv_136, SNV_136, col = RColorBrewer::brewer.pal(9,"
    Blues"),
        #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
    +004:4.5E+004),
        xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
    F136_SNV_Base off")


# new
open3d()
mfrow3d(3, 2, sharedMouse = TRUE)
persp3d(t_snv_295, w_snv_295, SNV_295, col = RColorBrewer::brewer.pal(9,"
    Blues"),
        #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
    +004:4.5E+004),
        xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
    F295_SNV_GB1")
persp3d(t_snv_296, w_snv_296, SNV_296, col = RColorBrewer::brewer.pal(9,"
    Blues"),
        #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
    +004:4.5E+004),
        xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
    F296_SNV_GB2")
persp3d(t_snv_297, w_snv_297, SNV_297, col = RColorBrewer::brewer.pal(9,"
    Blues"),
        #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
    +004:4.5E+004),
        xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
    F297_SNV_Air off")
persp3d(t_snv_298, w_snv_298, SNV_298, col = RColorBrewer::brewer.pal(9,"
    Blues"),
        #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
    +004:4.5E+004),
        xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
    F298_SNV_Base off")
persp3d(t_snv_299, w_snv_299, SNV_299, col = RColorBrewer::brewer.pal(9,"
    Blues"),
        #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
    +004:4.5E+004),
```

```
456          xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
      F299_SNV_Base on")
457
458
459  t_snv_134 = as.matrix(t_snv_134)
460  t_snv_137 = as.matrix(t_snv_137)
461  t_snv_145 = as.matrix(t_snv_145)
462  t_snv_136 = as.matrix(t_snv_136)
463
464  t_snv_295 = as.matrix(t_snv_295)
465  t_snv_296 = as.matrix(t_snv_296)
466  t_snv_297 = as.matrix(t_snv_297)
467  t_snv_298 = as.matrix(t_snv_298)
468  t_snv_299 = as.matrix(t_snv_299)
469
470  colnames(t_snv_134) = "Time"
471  colnames(t_snv_137) = "Time"
472  colnames(t_snv_145) = "Time"
473  colnames(t_snv_136) = "Time"
474
475  colnames(t_snv_295) = "Time"
476  colnames(t_snv_296) = "Time"
477  colnames(t_snv_297) = "Time"
478  colnames(t_snv_298) = "Time"
479  colnames(t_snv_299) = "Time"
480
481
482  #--- Raw data preprocessed with SNV and afterwards SG -----
483  ts_134 = as.vector(ts_134)
484  ts_137 = as.vector(ts_137)
485  ts_145 = as.vector(ts_145)
486  ts_136 = as.vector(ts_136)
487
488  ts_295 = as.vector(ts_295)
489  ts_296 = as.vector(ts_296)
490  ts_297 = as.vector(ts_297)
491  ts_298 = as.vector(ts_298)
492  ts_299 = as.vector(ts_299)
493
494  # historical
495  open3d()
496  mfrow3d(2, 2, sharedMouse = TRUE)
497  persp3d(ts_134, ws_134, SG_134, col = RColorBrewer::brewer.pal(9,"Blues")
      ,
498          #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
      +004:4.5E+004),
499          xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
      F134_SG_GB1")
500  persp3d(ts_137, ws_137, SG_137, col = RColorBrewer::brewer.pal(9,"Blues")
      ,
```

55

```r
501          #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
    +004:4.5E+004),
502          xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
    F137_SG_GB2")
503 persp3d(ts_145, ws_145, SG_145, col = RColorBrewer::brewer.pal(9,"Blues")
    ,
504          #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
    +004:4.5E+004),
505          xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
    F145_SG_Air off")
506 persp3d(ts_136, ws_136, SG_136, col = RColorBrewer::brewer.pal(9,"Blues")
    ,
507          #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
    +004:4.5E+004),
508          xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
    F136_SG_Base off")
509
510
511 # new
512 open3d()
513 mfrow3d(3, 2, sharedMouse = TRUE)
514 persp3d(ts_295, ws_295, SG_295, col = RColorBrewer::brewer.pal(9,"Blues")
    ,
515          #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
    +004:4.5E+004),
516          xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
    F295_SG_GB1")
517 persp3d(ts_296, ws_296, SG_296, col = RColorBrewer::brewer.pal(9,"Blues")
    ,
518          #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
    +004:4.5E+004),
519          xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
    F296_SG_GB2")
520 persp3d(ts_297, ws_297, SG_297, col = RColorBrewer::brewer.pal(9,"Blues")
    ,
521          #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
    +004:4.5E+004),
522          xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
    F297_SG_Air off")
523 persp3d(ts_298, ws_298, SG_298, col = RColorBrewer::brewer.pal(9,"Blues")
    ,
524          #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
    +004:4.5E+004),
525          xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
    F298_SG_Base off")
526 persp3d(ts_299, ws_299, SG_299, col = RColorBrewer::brewer.pal(9,"Blues")
    ,
527          #xlim = range(20:456), ylim = range(1350:1640), zlim = range(3.5E
    +004:4.5E+004),
528          xlab = "time", ylab = "wavelength", zlab = "absorbance", main = "
    F299_SG_Base on")
```

```
529
530
531  ts_134 = as.matrix(ts_134)
532  ts_137 = as.matrix(ts_137)
533  ts_145 = as.matrix(ts_145)
534  ts_136 = as.matrix(ts_136)
535
536  ts_295 = as.matrix(ts_295)
537  ts_296 = as.matrix(ts_296)
538  ts_297 = as.matrix(ts_297)
539  ts_298 = as.matrix(ts_298)
540  ts_299 = as.matrix(ts_299)
541
542  colnames(ts_134) = "Time"
543  colnames(ts_137) = "Time"
544  colnames(ts_145) = "Time"
545  colnames(ts_136) = "Time"
546
547  colnames(ts_295) = "Time"
548  colnames(ts_296) = "Time"
549  colnames(ts_297) = "Time"
550  colnames(ts_298) = "Time"
551  colnames(ts_299) = "Time"
552
553
554  #--- --- --- -----
555  #--------------------------- PERFORMING PrincipalComponentAnalysis (PCA)
          -------------------------
556  # Raw data without preprocessing
557  pca.134 = prcomp(x = NIR_134, center = TRUE, scale. = TRUE)
558  pca.137 = prcomp(x = NIR_137, center = TRUE, scale. = TRUE)
559  pca.145 = prcomp(x = NIR_145, center = TRUE, scale. = TRUE)
560  pca.136 = prcomp(x = NIR_136, center = TRUE, scale. = TRUE)
561
562  pca.295 = prcomp(x = NIR_295, center = TRUE, scale. = TRUE)
563  pca.296 = prcomp(x = NIR_296, center = TRUE, scale. = TRUE)
564  pca.297 = prcomp(x = NIR_297, center = TRUE, scale. = TRUE)
565  pca.298 = prcomp(x = NIR_298, center = TRUE, scale. = TRUE)
566  pca.299 = prcomp(x = NIR_299, center = TRUE, scale. = TRUE)
567
568
569  # Raw data preprocessed with SNV
570  pca.134snv = prcomp(x = SNV_134, center = TRUE, scale. = TRUE)
571  pca.137snv = prcomp(x = SNV_137, center = TRUE, scale. = TRUE)
572  pca.145snv = prcomp(x = SNV_145, center = TRUE, scale. = TRUE)
573  pca.136snv = prcomp(x = SNV_136, center = TRUE, scale. = TRUE)
574
575  pca.295snv = prcomp(x = SNV_295, center = TRUE, scale. = TRUE)
576  pca.296snv = prcomp(x = SNV_296, center = TRUE, scale. = TRUE)
577  pca.297snv = prcomp(x = SNV_297, center = TRUE, scale. = TRUE)
578  pca.298snv = prcomp(x = SNV_298, center = TRUE, scale. = TRUE)
```

```r
579  pca.299snv = prcomp(x = SNV_299, center = TRUE, scale. = TRUE)
580
581
582  # Raw data preprocessed with SNV and afterwards SG
583  pca.134sg = prcomp(x = SG_134, center = TRUE, scale. = TRUE)
584  pca.137sg = prcomp(x = SG_137, center = TRUE, scale. = TRUE)
585  pca.145sg = prcomp(x = SG_145, center = TRUE, scale. = TRUE)
586  pca.136sg = prcomp(x = SG_136, center = TRUE, scale. = TRUE)
587
588  pca.295sg = prcomp(x = SG_295, center = TRUE, scale. = TRUE)
589  pca.296sg = prcomp(x = SG_296, center = TRUE, scale. = TRUE)
590  pca.297sg = prcomp(x = SG_297, center = TRUE, scale. = TRUE)
591  pca.298sg = prcomp(x = SG_298, center = TRUE, scale. = TRUE)
592  pca.299sg = prcomp(x = SG_299, center = TRUE, scale. = TRUE)
593
594
595  #--- --- --- -----
596  #------------------------- PLOTTING OF PCA-DATA
         ----------------------------------------------
597  # Raw data without preprocessing
598  roha = autoplot(pca.134, data = t_134, colour = 'Time', main = "F134_GB1"
         )
599  rohb = autoplot(pca.137, data = t_137, colour = 'Time', main = "F137_GB2"
         )
600  rohc = autoplot(pca.145, data = t_145, colour = 'Time', main = "F145_Air
         off")
601  rohd = autoplot(pca.136, data = t_136, colour = 'Time', main = "F136_Base
          off")
602
603  roh1 = autoplot(pca.295, data = t_295, colour = 'Time', main = "F295_GB1"
         )
604  roh2 = autoplot(pca.296, data = t_296, colour = 'Time', main = "F296_GB2"
         )
605  roh3 = autoplot(pca.297, data = t_297, colour = 'Time', main = "F297_Air
         off")
606  roh4 = autoplot(pca.298, data = t_298, colour = 'Time', main = "F298_Base
          off")
607  roh5 = autoplot(pca.299, data = t_299, colour = 'Time', main = "F299_Base
          on")
608
609  grid.arrange(roha, rohb, rohc, rohd, ncol = 2)
610  grid.arrange(roh1, roh2, roh3, roh4, roh5, ncol = 2)
611
612  # Raw data preprocessed with SNV
613  snva = autoplot(pca.134snv, data = t_snv_134, colour = 'Time', main = "
         F134_SNV_GB1")
614  snvb = autoplot(pca.137snv, data = t_snv_137, colour = 'Time', main = "
         F137_SNV_GB2")
615  snvc = autoplot(pca.145snv, data = t_snv_145, colour = 'Time', main = "
         F145_SNV_Air off")
```

```
616  snvd = autoplot(pca.136snv, data = t_snv_136, colour = 'Time', main = "
        F136_SNV_Base off")
617
618  snv1 = autoplot(pca.295snv, data = t_snv_295, colour = 'Time', main = "
        F295_SNV_GB1")
619  snv2 = autoplot(pca.296snv, data = t_snv_296, colour = 'Time', main = "
        F296_SNV_GB2")
620  snv3 = autoplot(pca.297snv, data = t_snv_297, colour = 'Time', main = "
        F297_SNV_Air off")
621  snv4 = autoplot(pca.298snv, data = t_snv_298, colour = 'Time', main = "
        F298_SNV_Base off")
622  snv5 = autoplot(pca.299snv, data = t_snv_299, colour = 'Time', main = "
        F299_SNV_Base on")
623
624  grid.arrange(snva, snvb, snvc, snvd, ncol = 2)
625  grid.arrange(snv1, snv2, snv3, snv4, snv5, ncol = 2)
626
627  # Raw data preprocessed with SNV and afterwards SG
628  sga = autoplot(pca.134sg, data = ts_134, colour = 'Time', main = "F134_SG
        _GB1")
629  sgb = autoplot(pca.137sg, data = ts_137, colour = 'Time', main = "F137_SG
        _GB2")
630  sgc = autoplot(pca.145sg, data = ts_145, colour = 'Time', main = "F145_SG
        _Air off")
631  sgd = autoplot(pca.136sg, data = ts_136, colour = 'Time', main = "F136_SG
        _Base off")
632
633  sg1 = autoplot(pca.295sg, data = ts_295, colour = 'Time', main = "F295_SG
        _GB1")
634  sg2 = autoplot(pca.296sg, data = ts_296, colour = 'Time', main = "F296_SG
        _GB2")
635  sg3 = autoplot(pca.297sg, data = ts_297, colour = 'Time', main = "F297_SG
        _Air off")
636  sg4 = autoplot(pca.298sg, data = ts_298, colour = 'Time', main = "F298_SG
        _Base off")
637  sg5 = autoplot(pca.299sg, data = ts_299, colour = 'Time', main = "F299_SG
        _Base on")
638
639  grid.arrange(sga, sgb, sgc, sgd, ncol = 2)
640  grid.arrange(sg1, sg2, sg3, sg4, sg5, ncol = 2)
641
642  #--- --- --- -----
643  #
        ##############################################################################
644  ###### 2) Working with whole data set using the "MeanInt" function as
        first preprocess ###########
645  #
        ##############################################################################
```

```
646  #-------------------------------- SELECTION OF SPECIFIC AREAS (for averaging)
         -----------------------
647  # the parameters for the MeanInt funtion as follows: "data = raw spectral
         data"
648  #                                                    "x    = from row 5
         till the respective end, every 20th row"
649  #                                                    "i    = 7"
650   MW_134 = MeanInt(NIR_134, c(seq(5, nrow(NIR_134), 20)), 7)
651   MW_137 = MeanInt(NIR_137, c(seq(5, nrow(NIR_137), 20)), 7)
652   MW_145 = MeanInt(NIR_145, c(seq(5, nrow(NIR_145), 20)), 7)
653   MW_136 = MeanInt(NIR_136, c(seq(5, nrow(NIR_136), 20)), 7)
654
655   MW_295 = MeanInt(NIR_295, c(seq(5, nrow(NIR_295), 20)), 7)
656   MW_296 = MeanInt(NIR_296, c(seq(5, nrow(NIR_296), 20)), 7)
657   MW_297 = MeanInt(NIR_297, c(seq(5, nrow(NIR_297), 20)), 7)
658   MW_298 = MeanInt(NIR_298, c(seq(5, nrow(NIR_298), 20)), 7)
659   MW_299 = MeanInt(NIR_299, c(seq(5, nrow(NIR_299), 20)), 7)
660
661  #--- Optional cutting of wavelength -----
662   # # This was only done once to estimate the temporal impact of the first
         45 minutes.
663   # MW_134 = MW_134[,10:291]
664   # MW_137 = MW_137[,10:291]
665   # MW_145 = MW_145[,10:291]
666   # MW_136 = MW_136[,10:291]
667   #
668   # MW_295 = MW_295[,10:291]
669   # MW_296 = MW_296[,10:291]
670   # MW_297 = MW_297[,10:291]
671   # MW_298 = MW_298[,10:291]
672   # MW_299 = MW_299[,10:291]
673
674  #--- --- --- -----
675  #--- Creating time and wavelength as a vector for the averaged datasets
         ------------------------------
676   t_MW_134 = seq(1,length = nrow(MW_134))     # time
677   t_MW_134 = as.matrix(t_MW_134)
678   t_MW_137 = seq(1,length = nrow(MW_137))     # time
679   t_MW_137 = as.matrix(t_MW_137)
680   t_MW_145 = seq(1,length = nrow(MW_145))     # time
681   t_MW_145 = as.matrix(t_MW_145)
682   t_MW_136 = seq(1,length = nrow(MW_136))     # time
683   t_MW_136 = as.matrix(t_MW_136)
684
685   t_MW_295 = seq(1,length = nrow(MW_295))     # time
686   t_MW_295 = as.matrix(t_MW_295)
687   t_MW_296 = seq(1,length = nrow(MW_296))     # time
688   t_MW_296 = as.matrix(t_MW_296)
689   t_MW_297 = seq(1,length = nrow(MW_297))     # time
690   t_MW_297 = as.matrix(t_MW_297)
691   t_MW_298 = seq(1,length = nrow(MW_298))     # time
```

```
692   t_MW_298 = as.matrix(t_MW_298)
693   t_MW_299 = seq(1,length = nrow(MW_299))      # time
694   t_MW_299 = as.matrix(t_MW_299)
695
696
697   colnames(t_MW_134) = "Time"
698   colnames(t_MW_137) = "Time"
699   colnames(t_MW_145) = "Time"
700   colnames(t_MW_136) = "Time"
701
702   colnames(t_MW_295) = "Time"
703   colnames(t_MW_296) = "Time"
704   colnames(t_MW_297) = "Time"
705   colnames(t_MW_298) = "Time"
706   colnames(t_MW_299) = "Time"
707
708
709   w_MW_134 = as.numeric(colnames(MW_134))      # wavelength
710   w_MW_137 = as.numeric(colnames(MW_137))      # wavelength
711   w_MW_145 = as.numeric(colnames(MW_145))      # wavelength
712   w_MW_136 = as.numeric(colnames(MW_136))      # wavelength
713
714   w_MW_295 = as.numeric(colnames(MW_295))      # wavelength
715   w_MW_296 = as.numeric(colnames(MW_296))      # wavelength
716   w_MW_297 = as.numeric(colnames(MW_297))      # wavelength
717   w_MW_298 = as.numeric(colnames(MW_298))      # wavelength
718   w_MW_299 = as.numeric(colnames(MW_299))      # wavelength
719
720 #--- --- --- -----
721 #--- Baseline Correction via StandardNormalVariate (SNV)
        ----------------------------------------
722   # historical
723   SNV_MW_134 = apply(X = MW_134,MARGIN = 1,FUN = snv) # MARGIN = 1 for
        rows, = 2 for columns
724   SNV_MW_134 = t(SNV_MW_134)                          # transpose to
        create the axes t and w
725   t_snv_MW_134 = seq(1,length = nrow(SNV_MW_134))
726   t_snv_MW_134 = as.matrix(t_snv_MW_134)
727   colnames(t_snv_MW_134) = "Time"
728   w_snv_MW_134 = as.numeric(colnames(SNV_MW_134))
729
730   SNV_MW_137 = apply(X = MW_137,MARGIN = 1,FUN = snv) # MARGIN = 1 for
        rows, = 2 for columns
731   SNV_MW_137 = t(SNV_MW_137)                          # transpose to
        create the axes t and w
732   t_snv_MW_137 = seq(1,length = nrow(SNV_MW_137))
733   t_snv_MW_137 = as.matrix(t_snv_MW_137)
734   colnames(t_snv_MW_137) = "Time"
735   w_snv_MW_137 = as.numeric(colnames(SNV_MW_137))
736
```

```
737    SNV_MW_145 = apply(X = MW_145,MARGIN = 1,FUN = snv) # MARGIN = 1 for
         rows, = 2 for columns
738    SNV_MW_145 = t(SNV_MW_145)                          # transpose to
         create the axes t and w
739    t_snv_MW_145 = seq(1,length = nrow(SNV_MW_145))
740    t_snv_MW_145 = as.matrix(t_snv_MW_145)
741    colnames(t_snv_MW_145) = "Time"
742    w_snv_MW_145 = as.numeric(colnames(SNV_MW_145))
743
744    SNV_MW_136 = apply(X = MW_136,MARGIN = 1,FUN = snv) # MARGIN = 1 for
         rows, = 2 for columns
745    SNV_MW_136 = t(SNV_MW_136)                          # transpose to
         create the axes t and w
746    t_snv_MW_136 = seq(1,length = nrow(SNV_MW_136))
747    t_snv_MW_136 = as.matrix(t_snv_MW_136)
748    colnames(t_snv_MW_136) = "Time"
749    w_snv_MW_136 = as.numeric(colnames(SNV_MW_136))
750
751
752    # new
753    SNV_MW_295 = apply(X = MW_295,MARGIN = 1,FUN = snv) # MARGIN = 1 for
         rows, = 2 for columns
754    SNV_MW_295 = t(SNV_MW_295)                          # transpose to
         create the axes t and w
755    t_snv_MW_295 = seq(1,length = nrow(SNV_MW_295))
756    t_snv_MW_295 = as.matrix(t_snv_MW_295)
757    colnames(t_snv_MW_295) = "Time"
758    w_snv_MW_295 = as.numeric(colnames(SNV_MW_295))
759
760    SNV_MW_296 = apply(X = MW_296,MARGIN = 1,FUN = snv) # MARGIN = 1 for
         rows, = 2 for columns
761    SNV_MW_296 = t(SNV_MW_296)                          # transpose to
         create the axes t and w
762    t_snv_MW_296 = seq(1,length = nrow(SNV_MW_296))
763    t_snv_MW_296 = as.matrix(t_snv_MW_296)
764    colnames(t_snv_MW_296) = "Time"
765    w_snv_MW_296 = as.numeric(colnames(SNV_MW_296))
766
767    SNV_MW_297 = apply(X = MW_297,MARGIN = 1,FUN = snv) # MARGIN = 1 for
         rows, = 2 for columns
768    SNV_MW_297 = t(SNV_MW_297)                          # transpose to
         create the axes t and w
769    t_snv_MW_297 = seq(1,length = nrow(SNV_MW_297))
770    t_snv_MW_297 = as.matrix(t_snv_MW_297)
771    colnames(t_snv_MW_297) = "Time"
772    w_snv_MW_297 = as.numeric(colnames(SNV_MW_297))
773
774    SNV_MW_298 = apply(X = MW_298,MARGIN = 1,FUN = snv) # MARGIN = 1 for
         rows, = 2 for columns
775    SNV_MW_298 = t(SNV_MW_298)                          # transpose to
         create the axes t and w
```

```
776   t_snv_MW_298 = seq(1,length = nrow(SNV_MW_298))
777   t_snv_MW_298 = as.matrix(t_snv_MW_298)
778   colnames(t_snv_MW_298) = "Time"
779   w_snv_MW_298 = as.numeric(colnames(SNV_MW_298))
780
781   SNV_MW_299 = apply(X = MW_299,MARGIN = 1,FUN = snv) # MARGIN = 1 for
        rows, = 2 for columns
782   SNV_MW_299 = t(SNV_MW_299)                              # transpose to
        create the axes t and w
783   t_snv_MW_299 = seq(1,length = nrow(SNV_MW_299))
784   t_snv_MW_299 = as.matrix(t_snv_MW_299)
785   colnames(t_snv_MW_299) = "Time"
786   w_snv_MW_299 = as.numeric(colnames(SNV_MW_299))
787
788
789 #--- --- --- -----
790 #--- Smoothing via savitzkyGolay (SG) ------
      ---------------------------------------------------
791 # historical
792 SG_MW_134 = savitzkyGolay(X = SNV_MW_134, m = 1, p = 1, w = 13)
793 ts_MW_134 = seq(1,   length = nrow(SG_MW_134))      # time
794 ts_MW_134 = as.matrix(ts_MW_134)
795 colnames(ts_MW_134) = "Time"
796 ws_MW_134 = as.numeric(colnames(SG_MW_134))     # wavelength
797
798 SG_MW_137 = savitzkyGolay(X = SNV_MW_137, m = 1, p = 1, w = 13)
799 ts_MW_137 = seq(1,   length = nrow(SG_MW_137))      # time
800 ts_MW_137 = as.matrix(ts_MW_137)
801 colnames(ts_MW_137) = "Time"
802 ws_MW_137 = as.numeric(colnames(SG_MW_137))     # wavelength
803
804 SG_MW_145 = savitzkyGolay(X = SNV_MW_145, m = 1, p = 1, w = 13)
805 ts_MW_145 = seq(1,   length = nrow(SG_MW_145))      # time
806 ts_MW_145 = as.matrix(ts_MW_145)
807 colnames(ts_MW_145) = "Time"
808 ws_MW_145 = as.numeric(colnames(SG_MW_145))     # wavelength
809
810 SG_MW_136 = savitzkyGolay(X = SNV_MW_136, m = 1, p = 1, w = 13)
811 ts_MW_136 = seq(1,   length = nrow(SG_MW_136))      # time
812 ts_MW_136 = as.matrix(ts_MW_136)
813 colnames(ts_MW_136) = "Time"
814 ws_MW_136 = as.numeric(colnames(SG_MW_136))     # wavelength
815
816 # new
817 SG_MW_295 = savitzkyGolay(X = SNV_MW_295, m = 1, p = 2, w = 13)
818 ts_MW_295 = seq(1,   length = nrow(SG_MW_295))      # time
819 ts_MW_295 = as.matrix(ts_MW_295)
820 colnames(ts_MW_295) = "Time"
821 ws_MW_295 = as.numeric(colnames(SG_MW_295))     # wavelength
822
823 SG_MW_296 = savitzkyGolay(X = SNV_MW_296, m = 1, p = 2, w = 13)
```

```
824 ts_MW_296 = seq(1,    length = nrow(SG_MW_296))      # time
825 ts_MW_296 = as.matrix(ts_MW_296)
826 colnames(ts_MW_296) = "Time"
827 ws_MW_296 = as.numeric(colnames(SG_MW_296))      # wavelength
828
829 SG_MW_297 = savitzkyGolay(X = SNV_MW_297, m = 1, p = 2, w = 13)
830 ts_MW_297 = seq(1,    length = nrow(SG_MW_297))      # time
831 ts_MW_297 = as.matrix(ts_MW_297)
832 colnames(ts_MW_297) = "Time"
833 ws_MW_297 = as.numeric(colnames(SG_MW_297))      # wavelength
834
835 SG_MW_298 = savitzkyGolay(X = SNV_MW_298, m = 1, p = 2, w = 13)
836 ts_MW_298 = seq(1,    length = nrow(SG_MW_298))      # time
837 ts_MW_298 = as.matrix(ts_MW_298)
838 colnames(ts_MW_298) = "Time"
839 ws_MW_298 = as.numeric(colnames(SG_MW_298))      # wavelength
840
841 SG_MW_299 = savitzkyGolay(X = SNV_MW_299, m = 1, p = 2, w = 13)
842 ts_MW_299 = seq(1,    length = nrow(SG_MW_299))      # time
843 ts_MW_299 = as.matrix(ts_MW_299)
844 colnames(ts_MW_299) = "Time"
845 ws_MW_299 = as.numeric(colnames(SG_MW_299))      # wavelength
846
847
848
849 #--- --- --- -----
850 #--------------------------- PERFORMING PrincipalComponentAnalysis (PCA)
        ------------------------
851 # Raw data preprocessed with MeanInt
852 MW_pca.134 = prcomp(x = MW_134, center = TRUE, scale. = TRUE)
853 MW_pca.137 = prcomp(x = MW_137, center = TRUE, scale. = TRUE)
854 MW_pca.145 = prcomp(x = MW_145, center = TRUE, scale. = TRUE)
855 MW_pca.136 = prcomp(x = MW_136, center = TRUE, scale. = TRUE)
856
857 MW_pca.295 = prcomp(x = MW_295, center = TRUE, scale. = TRUE)
858 MW_pca.296 = prcomp(x = MW_296, center = TRUE, scale. = TRUE)
859 MW_pca.297 = prcomp(x = MW_297, center = TRUE, scale. = TRUE)
860 MW_pca.298 = prcomp(x = MW_298, center = TRUE, scale. = TRUE)
861 MW_pca.299 = prcomp(x = MW_299, center = TRUE, scale. = TRUE)
862
863 # MeanInt data processed with SNV
864 MW_pca.134snv = prcomp(x = SNV_MW_134, center = TRUE, scale. = TRUE)
865 MW_pca.137snv = prcomp(x = SNV_MW_137, center = TRUE, scale. = TRUE)
866 MW_pca.145snv = prcomp(x = SNV_MW_145, center = TRUE, scale. = TRUE)
867 MW_pca.136snv = prcomp(x = SNV_MW_136, center = TRUE, scale. = TRUE)
868
869 MW_pca.295snv = prcomp(x = SNV_MW_295, center = TRUE, scale. = TRUE)
870 MW_pca.296snv = prcomp(x = SNV_MW_296, center = TRUE, scale. = TRUE)
871 MW_pca.297snv = prcomp(x = SNV_MW_297, center = TRUE, scale. = TRUE)
872 MW_pca.298snv = prcomp(x = SNV_MW_298, center = TRUE, scale. = TRUE)
873 MW_pca.299snv = prcomp(x = SNV_MW_299, center = TRUE, scale. = TRUE)
```

```r
# MeanInt data processed with SNV and afterwards SG
MW_pca.134sg = prcomp(x = SG_MW_134, center = TRUE, scale. = TRUE)
MW_pca.137sg = prcomp(x = SG_MW_137, center = TRUE, scale. = TRUE)
MW_pca.145sg = prcomp(x = SG_MW_145, center = TRUE, scale. = TRUE)
MW_pca.136sg = prcomp(x = SG_MW_136, center = TRUE, scale. = TRUE)

MW_pca.295sg = prcomp(x = SG_MW_295, center = TRUE, scale. = TRUE)
MW_pca.296sg = prcomp(x = SG_MW_296, center = TRUE, scale. = TRUE)
MW_pca.297sg = prcomp(x = SG_MW_297, center = TRUE, scale. = TRUE)
MW_pca.298sg = prcomp(x = SG_MW_298, center = TRUE, scale. = TRUE)
MW_pca.299sg = prcomp(x = SG_MW_299, center = TRUE, scale. = TRUE)

#--- --- --- -----
#-------------------------- PLOTTING OF PCA-DATA
        --------------------------------------------
# Raw data preprocessed with MeanInt
roha = autoplot(MW_pca.134, data = t_MW_134, colour = 'Time', main = "F134
    _GB1")
rohb = autoplot(MW_pca.137, data = t_MW_137, colour = 'Time', main = "F137
    _GB2")
rohc = autoplot(MW_pca.145, data = t_MW_145, colour = 'Time', main = "F145
    _Air off")
rohd = autoplot(MW_pca.136, data = t_MW_136, colour = 'Time', main = "F136
    _Base off")

roh1 = autoplot(MW_pca.295, data = t_MW_295, colour = 'Time', main = "F295
    _GB1")
roh2 = autoplot(MW_pca.296, data = t_MW_296, colour = 'Time', main = "F296
    _GB2")
roh3 = autoplot(MW_pca.297, data = t_MW_297, colour = 'Time', main = "F297
    _Air off")
roh4 = autoplot(MW_pca.298, data = t_MW_298, colour = 'Time', main = "F298
    _Base off")
roh5 = autoplot(MW_pca.299, data = t_MW_299, colour = 'Time', main = "F299
    _Base on")


grid.arrange(roha, rohb, rohc, rohd, ncol = 2)
grid.arrange(roh1, roh2, roh3, roh4, roh5, ncol = 2)

# MeanInt data processed with SNV
snva = autoplot(MW_pca.134snv, data = t_snv_MW_134, colour = 'Time', main
    = "F134_SNV_GB1")
snvb = autoplot(MW_pca.137snv, data = t_snv_MW_137, colour = 'Time', main
    = "F137_SNV_GB2")
snvc = autoplot(MW_pca.145snv, data = t_snv_MW_145, colour = 'Time', main
    = "F145_SNV_Air off")
snvd = autoplot(MW_pca.136snv, data = t_snv_MW_136, colour = 'Time', main
    = "F136_SNV_Base off")

```

```r
snv1 = autoplot(MW_pca.295snv, data = t_snv_MW_295, colour = 'Time', main
    = "F295_SNV_GB1")
snv2 = autoplot(MW_pca.296snv, data = t_snv_MW_296, colour = 'Time', main
    = "F296_SNV_GB2")
snv3 = autoplot(MW_pca.297snv, data = t_snv_MW_297, colour = 'Time', main
    = "F297_SNV_Air off")
snv4 = autoplot(MW_pca.298snv, data = t_snv_MW_298, colour = 'Time', main
    = "F298_SNV_Base off")
snv5 = autoplot(MW_pca.299snv, data = t_snv_MW_299, colour = 'Time', main
    = "F299_SNV_Base on")


grid.arrange(snva, snvb, snvc, snvd, ncol = 2)
grid.arrange(snv1, snv2, snv3, snv4, snv5, ncol = 2)

# MeanInt data processed with SNV and afterwards SG
sga = autoplot(MW_pca.134sg, data = ts_MW_134, colour = 'Time', main = "
    F134_SG_GB1")
sgb = autoplot(MW_pca.137sg, data = ts_MW_137, colour = 'Time', main = "
    F137_SG_GB2")
sgc = autoplot(MW_pca.145sg, data = ts_MW_145, colour = 'Time', main = "
    F145_SG_Air off")
sgd = autoplot(MW_pca.136sg, data = ts_MW_136, colour = 'Time', main = "
    F136_SG_Base off")

sg1 = autoplot(MW_pca.295sg, data = ts_MW_295, colour = 'Time', main = "
    F295_SG_GB1")
sg2 = autoplot(MW_pca.296sg, data = ts_MW_296, colour = 'Time', main = "
    F296_SG_GB2")
sg3 = autoplot(MW_pca.297sg, data = ts_MW_297, colour = 'Time', main = "
    F297_SG_Air off")
sg4 = autoplot(MW_pca.298sg, data = ts_MW_298, colour = 'Time', main = "
    F298_SG_Base off")
sg5 = autoplot(MW_pca.299sg, data = ts_MW_299, colour = 'Time', main = "
    F299_SG_Base on")


grid.arrange(sga, sgb, sgc, sgd, snvd, snvd, ncol = 2)
grid.arrange(sg1, sg2, sg3, sg4, sg5, ncol = 2)

#--- --- --- -----
#
    #############################################################################

###### 3) Combining the datasets to unify the axis of rotation (largest
    variance in PCA) #########
#
    #############################################################################

#--- Adding the number of fermentation as a label in first column -----
  new134 = cbind(134, SG_MW_134)
```

```
943   new137 = cbind(137, SG_MW_137)
944   new145 = cbind(145, SG_MW_145)
945   new136 = cbind(136, SG_MW_136)
946
947   new295 = cbind(295, SG_MW_295)
948   new296 = cbind(296, SG_MW_296)
949   new297 = cbind(297, SG_MW_297)
950   new298 = cbind(298, SG_MW_298)
951   new299 = cbind(299, SG_MW_299)
952
953
954   # ! Depending on the wanted combinations, individual fermentation can be
          excluded via #comment !
955   # This was done for example for comparing new and historical with each
         other without the respective other
956
957
958   all_fermentations = rbind(new134, new137, new145, new136,
959                             new295, new296, new297, new298, new299
960                             )
961
962   RR = all_fermentations
963   # optional saving of dataset prior analysis
964   # write.csv2(x = RR, file = "All_Fermentations_priorPCA.csv", sep = ";",
          col.names = TRUE, row.names = FALSE)
965 #--- --- --- -----
966 #-------------------------- PERFORMING PrincipalComponentAnalysis (PCA)
        -------------------------
967   RR.pca = prcomp(x = RR[,-1], center = TRUE, scale. = TRUE)
968
969   #summary(RR.pca)
970   #autoplot(RR.pca)
971 #--- --- --- -----
972 #--- Collecting the PC-values in a new dataframe -----
973   df_out <- as.data.frame(RR.pca$x)
974 #--- --- --- -----
975 #--- Adding fermentation number (FNr) and time (as sequence) to the new
        data frame -----
976   df_out$FNr  = c(rep(134, nrow(SG_MW_134)),
977                   rep(137, nrow(SG_MW_137)),
978                   rep(145, nrow(SG_MW_145)),
979                   rep(136, nrow(SG_MW_136)),
980                   rep(295, nrow(SG_MW_295)),
981                   rep(296, nrow(SG_MW_296)),
982                   rep(297, nrow(SG_MW_297)),
983                   rep(298, nrow(SG_MW_298)),
984                   rep(299, nrow(SG_MW_299))
985                   )
986
987   df_out$Time = c(seq(1,nrow(SG_MW_134)),
988                   seq(1,nrow(SG_MW_137)),
```

```
989                         seq(1,nrow(SG_MW_145)),
990                         seq(1,nrow(SG_MW_136)),
991                         seq(1,nrow(SG_MW_295)),
992                         seq(1,nrow(SG_MW_296)),
993                         seq(1,nrow(SG_MW_297)),
994                         seq(1,nrow(SG_MW_298)),
995                         seq(1,nrow(SG_MW_299))
996                         )
997  #--- --- --- -----
998  #--- Percentage display of the total variance of the PCs -----
999    df_out = df_out %>% select(FNr, Time, PC1:nrow(df_out))
1000   #View(head(df_out))
1001
1002   percentage = round(RR.pca$sdev / sum(RR.pca$sdev) * 100, 2)
1003   percentage = paste(colnames(df_out[,3:ncol(df_out)]), "(", paste( as.
       character(percentage), "%", ")", sep="") )
1004
1005  #--- --- --- -----
1006  #--- Changing the class types of the fermentation and time column to level
         -----
1007   df_out$FNr  = factor(df_out$FNr, levels = c("134",
1008                                               "137",
1009                                               "145",
1010                                               "136",
1011                                               "295",
1012                                               "296",
1013                                               "297",
1014                                               "298",
1015                                               "299"
1016                         ))
1017   df_out$Time = factor(df_out$Time, levels = c(seq(1,nrow(SG_MW_137)) #
       nrow argument representing the longest file in the bound data set
1018                         ))                                          #
       137 or 298 : new or historical
1019  #--- --- --- -----
1020  #--- Creating a general theme for the background -----
1021   theme = theme(panel.background = element_blank(),panel.border=element_
       rect(fill=NA),panel.grid.major = element_blank(),panel.grid.minor =
       element_blank())#,strip.background=element_blank(),axis.text.x=element_
       text(colour="black"),axis.text.y=element_text(colour="black"),axis.
       ticks=element_line(colour="black"),plot.margin=unit(c(1,1,1,1),"line"))
1022  #--- --- --- -----
1023  #--------------------------- PLOTTING OF PCA-DATA
       -----------------------------------------------
1024   #   Depending on the selected line, the data points are represented as
       points ("geom_point()")
1025   #   or, according to their time course, as ascending numbers ("geom_text
       ()").
1026   #   Changes in the axis ranges can be performed via x- and y-lim.
1027   #   The respective "Px6"-plot displays the chosen PCs for each
       fermentation next to each other.
```

```
1028
1029    # Plotting PC1 against PC 2 -----
1030
1031    P1.2 = ggplot(df_out, aes(x = PC1, y = PC2, color = FNr, label =
         rownames(df_out)))+
1032        geom_point()+
1033        #xlim(-10,20)+
1034        #ylim(-10,15)+
1035        #theme+
1036        #geom_text(size=3.5)+
1037        xlab(percentage[1]) + ylab(percentage[2])+
1038        scale_color_manual(values=c("coral3", "blueviolet", "aquamarine4", "
         #6495ED",        # colours for the historical fermentations
1039                                    "orange", "brown", "darkgreen", "blue",
         "red" # colours for the new fermentations
1040                                    ))
1041    P1.2
1042
1043    # PC1 against PC2 for each fermentation respectively
1044    Px6 = P1.2 + facet_wrap( ~ FNr, nrow = 3) +
1045                theme(legend.position = "none") #+ xlim(-3,0) + ylim(-1,1)
1046    Px6
1047
1048
1049    # Plotting PC1 against PC 3 -----
1050    P1.3 = ggplot(df_out, aes(x = PC1,y = PC3, color = FNr, label = rownames
         (df_out)))+
1051        geom_point()+
1052        #xlim(-8,20)+
1053        #ylim(-10,5)+
1054        #theme+
1055        #geom_text(size=3)+
1056        xlab(percentage[1]) + ylab(percentage[3])+
1057        scale_color_manual(values=c("coral3", "blueviolet", "aquamarine4", "
         #6495ED",        # colours for the historical fermentations
1058                                    "orange", "brown", "darkgreen", "blue", "
         red" # colours for the new fermentations
1059                                    ))
1060    P1.3
1061
1062    # PC1 against PC3 for each fermentation respectively
1063    Px6 = P1.3 + facet_wrap( ~ FNr, nrow = 3) +
1064                theme(legend.position = "none") #+ xlim(-3,0) + ylim(-4,5)
1065    Px6
1066
1067    # Plotting PC2 against PC 3 -----
1068    P2.3 = ggplot(df_out, aes(x = PC2,y = PC3, color = FNr, label = rownames
         (df_out)))+
1069        geom_point()+
1070        #xlim(-10,20)+
1071        #ylim(-10,5)+
```

```
1072    #theme+
1073    #geom_text(size=3)+
1074    xlab(percentage[2]) + ylab(percentage[3])+
1075    scale_color_manual(values=c("coral3", "blueviolet", "aquamarine4", "
        #6495ED",      # colours for the historical fermentations
1076                              "orange", "brown", "darkgreen", "blue", "
        red" # colours for the new fermentations
1077                              ))
1078    P2.3
1079
1080    # PC2 against PC3 for each fermentation respectively
1081    Px6 = P2.3 + facet_wrap( ~ FNr, nrow = 3) +
1082          theme(legend.position = "none") #+ xlim(-5,5) + ylim(-5,5)
1083    Px6
1084
1085    # Plotting PC1 against PC 4 -----
1086    P1.4 = ggplot(df_out, aes(x = PC1,y = PC4, color = FNr, label = rownames
        (df_out)))+
1087      geom_point()+
1088      #xlim(-15,20)+
1089      #ylim(-12.5,5)+
1090      #theme+
1091      #geom_text(size=3)+
1092      xlab(percentage[1]) + ylab(percentage[4])+
1093      scale_color_manual(values=c("coral3", "blueviolet", "aquamarine4", "
        #6495ED",      # colours for the historical fermentations
1094                              "orange", "brown", "darkgreen", "blue", "
        red" # colours for the new fermentations
1095                              ))
1096    P1.4
1097
1098    # PC1 against PC4 for each fermentation respectively
1099    Px6 = P1.4 + facet_wrap( ~ FNr, nrow = 3) +
1100          theme(legend.position = "none") #+ xlim(-5,0) + ylim(-4,5)
1101    Px6
1102
1103
1104    # Plotting PC1 against PC x (in this case x = 5 -> PC5) -----
1105    P1.x = ggplot(df_out, aes(x = PC1,y = PC5, color = FNr, label = rownames
        (df_out)))+
1106      geom_point()+
1107      #xlim(-7.5,20)+
1108      #ylim(-5,5)+
1109      #theme+
1110      #geom_text(size=3)+
1111      xlab(percentage[1]) + ylab(percentage[5])+
1112      scale_color_manual(values=c("coral3", "blueviolet", "aquamarine4", "
        #6495ED",      # colours for the historical fermentations
1113                              "orange", "brown", "darkgreen", "blue", "
        red" # colours for the new fermentations
1114                              ))
```

```
1115   P1.x
1116
1117   # PC1 against PC x for each fermentation respectively
1118   Px6 = P1.x + facet_wrap( ~ FNr, nrow = 3) +
1119               theme(legend.position = "none") #+ xlim(-3,0) + ylim(-2,2)
1120   Px6
```

Chapter 10 - Bibliography

[1] Anna G. Cavinato, David M. Mayes, Zhihong Ge, and James B. Callis. Noninvasive method for monitoring ethanol in fermentation processes using fiber-optic near-infrared spectroscopy. *Analytical Chemistry*, 62(18):1977–1982, 1990.

[2] Zhihong Ge, Anna G. Cavinato, and James B. Callis. Noninvasive spectroscopy for monitoring cell density in a fermentation process. *Analytical Chemistry*, 66(8):1354–1362, 1994.

[3] John Crowley, S. Alison Arnold, Nigel Wood, Linda M. Harvey, and Brian McNeil. Monitoring a high cell density recombinant pichia pastoris fed-batch bioprocess using transmission and reflectance near infrared spectroscopy. *Enzyme and Microbial Technology*, 36(5):621–628, 2005.

[4] Matthew Scarff, S. Alison Arnold, Linda M. Harvey, and Brian McNeil. Near infrared spectroscopy for bioprocess monitoring and control: Current status and future trends. *Critical Reviews in Biotechnology*, 26(1):17–39, 2006.

[5] Albert E. Cervera, Nanna Petersen, Anna E. Lantz, Anders Larsen, and Krist V. Gernaey. Application of near-infrared spectroscopy for monitoring and control of cell culture and fermentation. *Biotechnology Progress*, 25(6):1561–1581, 2009.

[6] P. Roychoudhury, R. O'Kennedy, B. McNeil, and L. M. Harvey. Multiplexing fibre optic near infrared (nir) spectroscopy as an emerging technology to monitor industrial bioprocesses. *Anal Chim Acta*, 590(1):110–7, 2007.

[7] H. Sahm, G. Antranikian, K.P. Stahmann, and R. Takors. *Industrielle Mikrobiologie*. Springer Berlin Heidelberg, 2014.

[8] L. Dagge, K. Harr, M. Paul, and G. Schnedl. Classification of process analysis: offline, atline, online, inline. *Cement International*, 7:72–81, 2009.

[9] R.W. Kessler. *Prozessanalytik: Strategien und Fallbeispiele aus der industriellen Praxis*. Wiley-VCH, 2006.

[10] J. Popp, V.V. Tuchin, A. Chiou, and S.H. Heinemann. *Handbook of Biophotonics, Volume 3: Photonics in Pharmaceutics, Bioanalysis and Environmental Research*. Wiley, 2012.

[11] FDA. Guidance for Industry: Q8 Pharmaceutical Development, 2004.

[12] Paul Kroll, Alexandra Hofer, Sophia Ulonska, Julian Kager, and Christoph Herwig. Model-based methods in the biopharmaceutical process lifecycle. *Pharmaceutical Research*, 34, 2017.

[13] L. Neutsch, Alexandra Hofer, and Paul Kroll. Probennahme im bioprozess, 2019.

[14] P.H. Raven, R.F. Evert, and S.E. Eichhorn. *Biologie der Pflanzen*. De Gruyter, 2006.

[15] Aaron M. Neiman. Ascospore formation in the yeast saccharomyces cerevisiae. *Microbiology and molecular biology reviews : MMBR*, 69(4):565–584, 2005.

[16] B. Sonnleitner and O. Kappeli. Growth of saccharomyces cerevisiae is controlled by its limited respiratory capacity: Formulation and verification of a hypothesis. *Biotechnol Bioeng*, 28(6):927–37, 1986.

[17] Francisca Randez-Gil, Pascual Sanz, and Jose A. Prieto. Engineering baker's yeast: room for improvement. *Trends in Biotechnology*, 17(6):237–244, 1999.

[18] Kaisa Poutanen. Biotechnology in the food chain: New tools and applications for future foods. In *Biotechnology in the Food Chain: New tools and applications for future foods*. VTT, 1997.

[19] Jens Nielsen. Production of biopharmaceutical proteins by yeast. *Bioengineered*, 4(4):207–211, 2013.

[20] Jin Hou, Keith EJ Tyo, Zihe Liu, Dina Petranovic, and Jens Nielsen. Metabolic engineering of recombinant protein secretion by saccharomyces cerevisiae. *FEMS yeast Research*, 12(5):491–510, 2012.

[21] H.W. Siesler, Y. Ozaki, S. Kawata, and H.M. Heise. *Near-Infrared Spectroscopy: Principles, Instruments, Applications*. Wiley, 2008.

[22] Michael J. McShane and Gerard L. Coté. Near-infrared spectroscopy for determination of glucose, lactate, and ammonia in cell culture media. *Applied Spectroscopy*, 52(8):1073–1078, 1998.

[23] M. R. Riley, M. Rhiel, X. Zhou, M. A. Arnold, and D. W. Murhammer. Simultaneous measurement of glucose and glutamine in insect cell culture media by near infrared spectroscopy. *Biotechnol Bioeng*, 55(1):11–5, 1997.

[24] Martin Rhiel, Michael B. Cohen, David W. Murhammer, and Mark A. Arnold. Nondestructive near-infrared spectroscopic measurement of multiple analytes in undiluted samples of serum-based cell culture media. *Biotechnology and bioengineering*, 77(1):73–82, 2002.

[25] S. Alison Arnold, John Crowley, Nigel Woods, Linda M. Harvey, and Brian McNeil. Insitu near infrared spectroscopy to monitor key analytes in mammalian cell cultivation. *Biotechnology and Bioengineering*, 84(1):13–19, 2003.

[26] A. Hagman and P. Sivertsson. The use of nir spectroscopy in monitoring and controlling bioprocesses. *Process Control and Quality*, 11(2):125–128, 1998.

[27] S. Alison Arnold, Rumpai Gaensakoo, Linda M. Harvey, and Brian McNeil. Use of at-line and in-situ near-infrared spectroscopy to monitor biomass in an industrial fed-batch escherichia coli process. *Biotechnology and Bioengineering*, 80(4):405–413, 2002.

[28] Jeffrey W. Hall, Brian McNeil, Malcolm J. Rollins, Indira Draper, Brad G. Thompson, and Graeme Macaloney. Near-infrared spectroscopic determination of acetate, ammonium, biomass, and glycerol in an industrial escherichia coli fermentation. *Applied Spectroscopy*, 50(1):102–108, 1996.

[29] Marko Sandor, Ferdinand Rüdinger, Dörte Solle, Roland Bienert, Christian Grimm, Sven Groß, and Thomas Scheper. Nir-spectroscopy for bioprocess monitoring & control. *BMC Proceedings*, 7(Suppl 6):P29–P29, 2013.

[30] Liang Zhao, Hsu-Yuan Fu, Weichang Zhou, and Wei-Shou Hu. Advances in process monitoring tools for cell culture bioprocesses. *Engineering in Life Sciences*, 15(5):459–468, 2015.

[31] G. Reich. Near-infrared spectroscopy and imaging: basic principles and pharmaceutical applications. *Adv Drug Deliv Rev*, 57(8):1109–43, 2005.

[32] K. Kiviharju, K. Salonen, U. Moilanen, E. Meskanen, M. Leisola, and T. Eerikäinen. On-line biomass measurements in bioreactor cultivations: comparison study of two on-line probes. *Journal of industrial microbiology & biotechnology*, 34(8):561–566, 2007.

[33] José Alves-Rausch, Roland Bienert, Christian Grimm, and Dirk Bergmaier. Real time in-line monitoring of large scale bacillus fermentations with near-infrared spectroscopy. *Journal of biotechnology*, 189:120–128, 2014.

[34] N. Misra, Carl Sullivan, and P. Cullen. Process analytical technology (pat) and multivariate methods for downstream processes. *Current Biochemical Engineering*, 2(1):4–16, 2015.

[35] Bence Kozma, András Salgó, and Szilveszter Gergely. On-line glucose monitoring by near infrared spectroscopy during the scale up steps of mammalian cell cultivation process development. *Bioprocess and biosystems engineering*, 42(6):921–932, 2019.

[36] T. De Beer, A. Burggraeve, M. Fonteyne, L. Saerens, J. P. Remon, and C. Vervaet. Near infrared and raman spectroscopy for the in-process monitoring of pharmaceutical production processes. *Int J Pharm*, 417(1-2):32–47, 2011.

[37] J. Classen, F. Aupert, K. F. Reardon, D. Solle, and T. Scheper. Spectroscopic sensors for in-line bioprocess monitoring in research and pharmaceutical industrial application. *Anal Bioanal Chem*, 409(3):651–666, 2017.

[38] W. Kessler. *Multivariate Datenanalyse: für die Pharma, Bio- und Prozessanalytik*. Wiley, 2011.

[39] Johann Lohninger. Hauptkomponentenanalyse (PCA). `http://www.statistics4u.info/fundstat_germ/cc_pca.html`, 2012. Online; accessed 19-August-2019.

[40] J. Hausjell, J. Weissensteiner, C. Molitor, H. Halbwirth, and O. Spadiut. E. coli hms174(de3) is a sustainable alternative to bl21(de3). *Microb Cell Fact*, 17(1):169, 2018.

[41] Rogelio Lopes Brandão, Julio Cesar Camara Rosa, Jacques Robert Nicoli, Marcos Vinicius Simi Almeida, Ana Paula do Carmo, Heloa Teixeira Queiros, and Ieso Miranda Castro. Investigating acid stress response in different saccharomyces strains. *Journal of Mycology*, 2014:9, 2014.

[42] X. Liu, B. Jia, X. Sun, J. Ai, L. Wang, C. Wang, F. Zhao, J. Zhan, and W. Huang. Effect of initial ph on growth characteristics and fermentation properties of saccharomyces cerevisiae. *J Food Sci*, 80(4):M800–8, 2015.

[43] Antonio Peña, Norma Silvia Sánchez, Helber Álvarez, Martha Calahorra, and Jorge Ramírez. Effects of high medium ph on growth, metabolism and transport in saccharomyces cerevisiae. *FEMS Yeast Research*, 15(2), 2015.

[44] A. D. King Jr and Charles W. Nagel. Influence of carbon dioxide upon the metabolism of pseudomonas aeruginosa. *Journal of Food Science*, 40(2):362–366, 1975.

[45] Rodney P. Jones and Paul F. Greenfield. Effect of carbon dioxide on yeast growth and fermentation. *Enzyme and Microbial Technology*, 4(4):210–223, 1982.

[46] S. L. Chen and F. Gutmanis. Carbon dioxide inhibition of yeast growth in biomass production. *Biotechnology and Bioengineering*, 18(10):1455–1462, 1976.

[47] H. Lange and S. Heijnen. Statistical reconciliation of the elemental and molecular biomass composition of saccharomyces cerevisiae. *Biotechnology and bioengineering*, 75:334–44, 2001.

List of Tables

List of Figures