Phosphoproteomic analysis of liver in a mouse reference population

Master Thesis

University of Natural Resources and Life Sciences, Vienna – BOKU
Department of Biotechnology

Fabian Frommelt
0946440

Prepared at the Institute of Molecular System Biology,
laboratory of Prof. Dr. Ruedi Aebersold
(ETH Zürich)
May 6, 2016

Supervisor: Peter Sykacek, Priv.-Doz. Dr.
Co-Supervisor: Peter Blattmann, Dr.

Phosphoproteomic analysis of liver in a mouse reference population

Fabian Frommelt
Department of Biotechnology, BOKU
Institute of Molecular System Biology, ETH Zürich
0946440
fabian.frommelt@boku.ac.at
fabianf@student.ethz.ch

Master thesis
July 2015 – May 2016

Referee: Peter Sykacek, Priv-Doz. Dr.
Supervisor: Peter Blattmann, Dr.

BOKU Wien,
Department of Biotechnology
Muthgasse 18
1190 Vienna, Austria

ETH Zürich
Institute of Molecular System Biology
Wolfgang-Pauli-Strasse 16
8093 Zürich, Switzerland

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Zurich,…… **……..**
(Date) (Signature)

Abstract

Both genetic and environmental risk factors affect complex phenotypic traits. Genetic loci that affect quantitative traits are termed quantitative trait loci (QTL) and can be identified by analyzing a phenotypic trait and the genetic markers of different individuals from a related family. In this study, we have examined the BXD mouse genetic reference panel – a panel of mice generated by crossing two different mouse strains – and have identified many such QTLs for gene and protein expression. To discover phosphoprotein-QTLs in mouse liver tissue, we measured the abundance of hepatic phosphopeptides in 40 BXD strains fed with two diverse diets using quantitative mass spectrometry. First, different phosphopeptide enrichment strategies were tested and further optimized. For the quantification of the SWATH-MS data, we generated a sample specific SWATH assay library which consisted of 2859 phosphopeptides of 1253 phosphoproteins. Secondly, the optimized single step phosphopeptide enrichment protocol was combined with the recently developed DIA method SWATH-MS to achieve high-quality quantitative data of the phosphoproteome over the cohort of the BXD strains. With our validated phospho-SWATH-MS approach we were able to quantify roughly 2000 phosphopeptides of 970 phosphoproteins and discovered 21 cis-phosho-pQTLs and 42 trans-phospho-pQTLs. The dataset is the first determination of the phosphoproteome across any genetically diverse reference population. Further, for the first time this dataset allows the identification of a large number of phosphoprotein-QTLs in a genetic reference population. In summary, the data showed a promising way to use quantitative phosphoproteomics data to elucidate the gene-environment interaction in complex diseases.

Zusammenfassung (German abstract)

Das Verständnis, wie genetische und umweltbedingte Risikofaktoren komplexe phänotypische Ausprägungen beeinflussen hilft, bei der Charakterisierung von komplexen Krankheiten. Quantitative trait Loci (QTL) Analyse ermöglicht die Korrelation von quantitativen Daten zu Abschnitten auf dem Genom, die solche genetischen Risikofaktoren beherbergen. In dieser Studie, die in die laufende Forschung und Charakterisierung der genetischen BXD-Maus Referenzpopulation eingebettet ist, haben wir quantitative Daten des Leber-Phosphoproteoms von 40 BXD Stämmen, welche mit zwei unterschiedlichen Diäten ernährt wurden, mit Hilfe der Massenspektrometrie (MS) gemessen. Diese Daten wurden verwendet um Phosphoprotein-QTLs zu identifizieren. Zunächst wurden verschiedene Phosphopeptid Anreicherungsstrategien getestet, um spezifisch die Phosphopeptide von Maus Leber anzureichern. Die Daten wurden mit der kürzlich publizierten MS-Methode SWATH gemessen. Für die Extraktion der SWATH-MS-Daten verwendeten wir eine Maus Leber Phosphopeptid SWATH-Proben-Bibliothek, welche für 2859 Phosphopeptide von 1253 Phosphoproteinen SWATH-Proben enthielt. Durch die Entfernung aller Phosphopeptide mit mehrdeutigen Lokalisierungen der Phosphorylierungsstelle, erhielten wir eine hochwertige Proben-Bibliothek. Mit unserem optimierten Phospho-SWATH-MS Ansatz konnten wir rund 2000 Phosphopeptide von 970 Phosphoproteinen quantifizieren. Mit den Daten war es möglich 21 cis-Phospho-pQTLs und 42 trans-Phospho-pQTLs zu entdecken. Der Datensatz ist daher die erste Bestimmung des Phosphoproteoms in einer BXD-Maus Population. Ferner ermöglicht dieser Datensatz zum ersten Mal die Identifizierung einer großen Anzahl von Phosphoprotein-QTLs. Zusammenfassend zeigen die Daten, eine vielversprechende Möglichkeit, quantitative Phosphoproteomics für die Aufklärung der Gen-Umwelt-Interaktion in komplexen Krankheiten zu verwenden.

Acknowledgements

I am very grateful to my co-advisor Peter Blattmann, who has been a great mentor during my time at the IMSB at ETH Zurich. Without his advice, support and guidance this truly interdisciplinary project would not have been possible. Also thanks a lot for the fruitful discussions, the helpful hand when R did not work as I wanted it to, the patience with me, the motivation and the regular meetings. The comments and feedback were very important for the success of the project. I also appreciated the invaluable suggestions and comments for my thesis and my future plans. In my opinion, you are a great advisor and I wish you all the best for your future. I am looking forward to finish this project and continue to collaborate in future ones.

I would like to thank my advisor Peter Sykacek for his time and interest for my master thesis project. Also thanks a lot for your comments and suggestions about my writing, my talks, and my scientific future. I am looking forward to keep in contact and continue collaborating in future projects.

I want to thank Prof. Ruedi Aebersold for the opportunity to carry out my master thesis within the very stimulating environment of his working group at the ETH Zurich. I would also like to thank the Aebersold group members for a very friendly and nice working atmosphere. All of them are unique and their positive attitude created a great social environment at work. Especially, I want to acknowledge Audrey van Drogen and Noel Imboden, who supported me at my lab-job.

I want to further thank the mass spectrometry operators Tatjana Sajic, Alexander Leitner, Peter Blattmann and Marco Faini for helping me with the MS measurements.

Sincere thanks to Yibo Wu for showing me the mouse liver lysis protocol and providing me with the SWATH assay library for total mouse liver tissue lysate. I also want to thank Evan G. Williams for helping me with the QTL analysis and helpful discussions. Further I want to acknowledge Tiannan Guo for teaching me the PCT lysis method and for the suggestions and discussions for optimizing the phosphopeptide enrichment. I would also like to thank Ariel Bensimon for helpful comments and suggestions for the phosphopeptide enrichment as well as for his support with food.

I want to thank our collaborator Evan G. Williams from the IPFL Genova form the Auwerx laboratory for preparing the mouse liver tissue samples.

Furthermore, I would like to thank Peter Blattmann, Peter Sykacek, Evan G. Williams, Orlando Eleonora, and Torsten Müller for proofreading parts of my thesis and providing valuable comments.

A special acknowledgment goes to Anna, who supported me during my studies and provided me with guidance and helpful discussions for considering scientific and non-scientific related questions over the past years.

Finally, I owe my deepest gratitude to my parents Barbara and Hubert, to my siblings Laura and Felix, and my friends, who always supported me during the final stages of my studies. Special thanks goes to my colleagues Katharina Leeb, Philipp Mundsperger, Andreas Weber and Karin Ortmayr for supporting me during my studies. A special acknowledgment goes to my mentor and good advisor Florian Rüker for all the discussions and advices during my studies at BOKU.

Table of Contents

1. Introduction

1.1. Gene-environment interactions in complex diseases

The identification of genes that contribute to the risk of complex diseases is one of the goals of medical genetics [1]. The understanding how genetic and environmental risk factors affect complex phenotypic traits can be used to elucidate the development of complex diseases. It is often not well understood how genetic factors, environmental factors, and their interactions contribute to the cause and development of complex diseases. Different methods, including linkage analysis, genome-wide association studies (GWAS) and quantitative trait loci (QTL) analysis, have been developed to narrow down the location of genes, contributing to genetic risk factors [2].

Until the 1980s the lack of polymorphic markers limited the genetic linkage analysis to a few model organisms [3]. The discovery of abundant molecular markers led to rapid advances in genotyping and by further improvements in the statistical methods for QTL analysis. Mapping studies such as the landmark study of Lander and Botstein became feasible [4]. Nowadays, an increasing number of large-scale genome-wide maps of QTLs for several model organisms, as well as in humans, reveal the improvements made in the field [1],[5].

For mapping acquired quantitative data to complex traits, it must be taken into consideration that the effect of any single factor may be shielded or mixed up by other contributing factors. In fact, separating these contributions separate to different effects is an enormous task, and thus only a few corresponding genetic risk factors have precisely been identified for complex diseases. Despite the limited clinical successes to date, much basic knowledge has been obtained on how gene–environment interactions affect complex disease.

Large amounts of private and national research effort and funding is still going into studying these complex diseases due to the severity of their impact on the population. Complex diseases comprises many common diseases from which large parts of the population are affected. Examples for complex diseases include Alzheimer's disease, scleroderma, asthma, cardio-vascular diseases, Parkinson's disease, multiple sclerosis, osteoporosis, connective tissue diseases, kidney diseases, autoimmune diseases, and many more [6].

Over the last decades, several approaches were developed, to study the interaction of gene products and environmental factors to the molecular level. These approaches has been promising, and we have seen that knowledge on the cellular level of

transcripts, proteins, and metabolites can further help the scientific community to understand the effects of gene-environment interactions on the cause of complex diseases. The aim of such studies is, to identify genetic risk factors, which are modified by environmental-specific manner and therefore can cause complex diseases [7].

One of these methods, linkage analysis, was for many years the predominant statistical tool for genetic mapping. Linkage studies have shown that they were successful in mapping genome regions that likely contain a rare allele variant, which show a large effect for the phenotype. With this technique, it was possible to study Mendelian diseases or also called monogenetic diseases, and identify loci for some complex traits, including Alzheimer's diseases and hypertension, within family datasets [8], [9]. It was possible to study the alleles that make large contributions to the disease or the variation of a quantitative trait [2].

Another approach is association analysis of complex traits in common variants, which focus on the analysis of modest effects in genetically unrelated populations. For such variants, association analysis like GWAS have proved to have more statistical power compared to linkage analysis. GWAS of single-nucleotide polymorphisms (SNPs) marker in large cohorts are used to associate complex traits to genomic loci [10]. However, one disadvantage of GWAS is, that rare variants cannot be well identified. This is very unfortunate, as nowadays, an emerging view is that rare variations with modest effect size seem to constitute much of variation in many complex diseases [11].

For the current study, QTL analysis was used. Thereby a region of the genome containing several genes which lead to variation in a quantitative trait is identified. The challenging analysis of complex phenotypic traits, became feasible by the establishment of large collections of genetic markers. These genetic markers were used to generate genetic maps, which were further used to correlate quantitative data to the markers and to identify the gene loci which are involved in the shaping of the quantitative trait (e.g. such as height, body weight, liver size) [12]. Linkage mapping of QTLs are conducted in families or the segregating progeny of crosses between genetically divergent strains such as the BXD mice. The difference to linkage analysis is that the cause of a quantitative trait, instead of dichotomous phenotype, is mapped to the genome. An advantage of QTL analysis in an inbred population is that it is possible to use a systems biology approach to integrate genotype-phenotype relationships across multiple layers of cellular organization. By using this approach, we can integrate the quantitative information of the transcriptome, the proteome, and

the metabolome to reveal affected molecular pathways which lead to the distinct phenotypes [3].

1.2. The BXD genetic mouse reference population

QTL analysis has to be performed in a family of related individuals that show a variation in the measured phenotype. For this reason the BXD panel of recombinant inbred (RI) strains was constructed from the two parental strains C57BL/6J and DBA/2J [13]. To create such RI strains, the two parental strains were crossed successively between siblings after the F2 generation. This repeated mating of the siblings was conducted for 20 generations. After 20 generations of inbreeding, with 99.5 % of the genome fixed, strains are considered fully inbred [14]. These inbred strains are almost homozygous at almost every location along the genome. Further they consisted of chromosomes with a fixed and permanent set of recombinations per chromosome. Thus, stable RI lines were obtained, which show vastly varying phenotypes [15]. This BXD mouse genetic reference population is used to investigate by QTL mapping the associations of complex traits to multiple genome regions (e.g. loci) on the mouse genome. Another advantage is, that both progenitor strains are known to exhibit widely different phenotypes and both strains have been sequenced, and show approximately 1.8 million SNPs [13]. Further, the BXD RI strains proved to be well suited as a translational/mechanistic bridge between reductionist and integrative approaches. In addition, the BXD reference panel is suited to combine the strengths of the two distinct scientific approaches in genotype-phenotype relations of complex traits: the reductionist and the system biology approach [16].

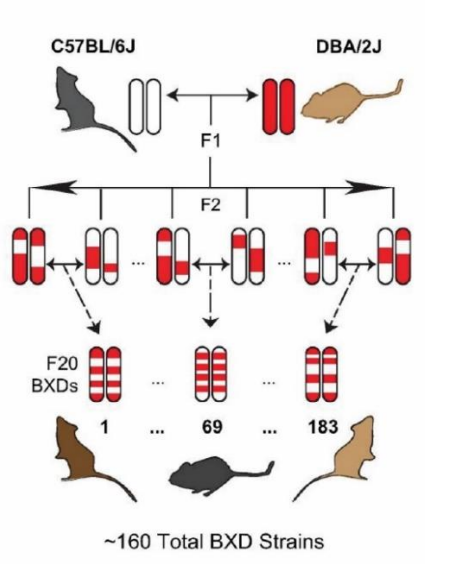


Figure 1: The BXD lines were created by crossing C57BL/6J and DBA/2J parents. *The resulting heterozygous F1 mice were again crossed to generate genetically diverse but nonreproducible F2 animals. These F2 progeny were iteratively inbred until generation F20+, at which point the genome was 99.5+% isogenic and the strains are considered fully inbred and together constitute a genetic reference population. The ~160 BXD strains are numbered 1–183. (The illustration was taken from [17])*

The BXD family is the largest and best characterized genetic reference population, consisting of approximately 160 RI strains [17]. The 40 BXD strains analyzed in the current study were fed with two diverse diets: a healthy “chow” diet, and an unhealthy high fat diet. These, environmental differences, on top of the variation in genetics across strains, lead to variable metabolic phenotypes across the BXD population [17]. For this set of BXD mouse strains, the trans-omics integration of several datasets with the genome, including transcriptome, metabolome, proteome, and phenome data, were already conducted [18]. For the systems proteomics datasets, a set of pre-defined proteins was measured with selected reaction monitoring (SRM), and integrated with a set of the metabolome and the transcriptome to discover QTLs [18]. In another study, the proteome was measured by using the data independent acquisition (DIA) SWATH-MS to systematically quantify the proteome, to further discover and validate new protein-QTLs [19]. In the current study, we aimed to add the phosphoproteome as another omics-layer, to the 40 BXD strains. The gained quantitative phosphoproteomic SWATH-MS data were used to discover phosphoprotein-QTLs. In order to measure the phosphoproteome prior to the data acquisition a selective enrichment of the phosphoproteome is needed. To test the optimized enrichment procedure for its performance and variability, we used a small set of young and old mouse liver tissue samples.

1.3. Biological process of aging

Aging is an intricate part of life generally characterized by a progressive decline in physiological function and an increase propensity to degenerative diseases and death [20]. Aging also comes with increasing physical and mental dysfunction and illnesses, including common metabolic, inflammatory, cardiovascular, and neurodegenerative diseases. The environmental and genetic risk factors for age related complex diseases are of high interest. Further, the understanding of the biological networks behind the aging process – the perturbation of metabolic pathways, and the alteration and decline in cellular function – are of high interest for the better understanding of aging in healthy organisms [21]. For several molecular processes evidence was found, that they are involved or at least contribute to the aging process. This includes DNA damage, mitochondrial dysfunction, accumulation of reactive oxygen species (ROS), and metabolic dysfunction. The major underlying molecular mechanisms of aging remain still largely elusive. Thus, many aging theories were developed, including the free-radical and the mitochondrial theory. Free radicals and ROS are toxic for the cell, which causes direct damage of sensitive biologically targets, and leading to oxidative stress. ROS are highly reactive molecules including hydrogen peroxide (H_2O_2), superoxide anion (O_2^-) and hydroxyl radical (OH). These molecules

have a high potential to cause oxidative deterioration of DNA, protein, and lipid. The ROS species are mainly produced as by-products by the mitochondrial respiration. The accumulation of ROS related damages leads to an upregulation of several enzymatic and nonenzymatic biological systems to defend against this toxicity [22].

In the current study, we used a small dataset of two young and two old mouse liver tissue samples. The same liver tissue samples were used in a multi-omics study, were the metabolic footprint of aging was unrevealed [21]. We used the mouse liver tissue samples of the "aging" experiment, to i) estimate the reproducibility of our phosphopeptide enrichment procedure by doing the experiment in triplicates, ii) estimate the reproducibility in the total tissue lysate, and iii) try to identify due to aging, differently regulated proteins and phosphoproteins.

1.4. Mass spectrometry-based proteomics

The method of choice for cellular, molecular and systems biology, to characterize in a reproducible manner a large quantity of proteins or phosphoproteins is mass spectrometry (MS). As predicted a few years ago by Aebersold and Mann, the abilities of MS technologies to identify and quantify thousands of proteins in a single shot already have a broad impact on biology and personalized medicine [23]. In a typical "bottom-up" proteomic experiment, proteins of a lysate are cleaved by the endopeptidase trypsin, separated over a liquid chromatography (LC) and further analyzed in the MS. The peptides are separated by their hydrophobicity, whereas hydrophobic peptides are longer retained by the reversed phase silica material of the LC column. The mass to charge ratio (m/z) of the ionized precursors is measured on the MS1, and after fragmentation via collision-induced dissociation, the m/z of the fragment-ions is determined on the second mass to charge analyzer.

MS is an essential tool for protein analysis owing to its speed, sensitivity, and versatility [24]. For the first time MS allows us to systematically measure the proteome of complex biological systems. The identification and characterization of the highly variable and complex environment of protein expression and its regulation within a biological system, was only feasible, to the ongoing developments in MS techniques and instruments over the past decades. With proteomics it is not only possible to study the protein content of cells, it further allows the characterization of post translational modification (PTMs), and the relative quantification of proteins. MS instruments for the analysis of proteins in the "bottom-up" proteomics approach, are typically interfaced to a liquid chromatography (LC) system for peptide separation to reduce complexity prior to the measurement. High-performance LC (HPLC) has become a standard LC system and can be coupled to many different MS settings. Several

different types of HPLC chromatographic materials are used in proteomics, ion exchange (IEX), reversed-phase, hydrophilic interaction chromatography (HILIC), affinity materials and size exclusion chromatography (SEC) [25]. After elution from the column, the analytes are transferred through a capillary to the ion source. For proteomics soft ionization techniques are used, which can be achieved with matrix-assisted desorption ionization (MALDI) and electrospray ionization (ESI). Nowadays, the standard technique for LC-MS/MS is ESI. Electrospray ionization is driven by high voltage (2 – 6 kV) applied between the emitter at the end of the chromatographic separation pipeline, and the inlet of the MS. This strong electric field causes the dispersion of the sample solution into aerosol of highly charged droplets. By applying a flow dry gas such as N_2 or Ar around the capillary the dispersion is increased and smaller droplets are formed. The liquid droplets evaporate further on their way to the MS inlet, until they release free ions, which are negatively or positively charged, due to the applied tip charge. The charged ions are attracted by the opposite charged inlet of the MS. This leads to single and multiple ionized peptides which are entering the MS device, where they are further transported to the mass analyzer, which is kept under vacuum [26]. Several different types of mass analyzers were constructed. The mass analyzers which were used for this study, included ion traps and Orbitraps, which separate ions based on differences in their mass to charge ratio (m/z) resonance frequency, and time-of-flight mass analyzers (ToF), which measure the flight time of ions over a defined distance to a detector. It is important to understand, that each of the mass analyzers have unique properties, such as mass range, analysis speed, resolution, sensitivity, ion transmission, and dynamic range. The final signal is detected on the detector and given out as m/z [25]. However, nowadays Hybrid mass spectrometers, which contain more than one mass analyzers, are used for specific analysis tasks. For the current thesis, we mainly used the Thermo Fisher OrbitrapElite and the SCIEX 5600+ TripleToF, to acquire high-end proteomics data. We used these instruments, as they enabled us to operate in different modes, including untargeted and targeted proteomics.

Untargeted proteomics

If the MS analysis is used for exploratory proteomics, also known as “shotgun proteomics”, discovery proteomics, or untargeted proteomics, it is operated in the data depended acquisition (DDA) mode. OrbitrapElite was operated in DDA mode and was used to analyze phosphopeptide enriched mouse liver tissue samples of the testing datasets. The most commonly employed operation mode for the instrument became acquisition of full scans in the Orbitrap analyzer and data-dependent MS/MS scans in the ion trap analyzer. In this operation mode, the ionized peptides are guided through the “S-lens”, which is an efficient ion transfer system, which is also called stacked ring

ion guide. For high resolution MS measurements precursor ions are accumulated in the high pressure cell of the dual-pressure ion trap assembly, which comprises of two identical linear quadrupoles. The first trap efficiently captures ions at relatively high pressure. From the high pressure cell an accumulated ion packet is passed through the C-trap on the Orbitrap analyzer. In the C-trap the ions are stored and lose energy in gentle collision with the bath gas. The ions are ejected orthogonally to the curved axis of the C-trap to the Orbitrap analyzer, were they enter as packets of different m/z. The ion packages forming a thin rotating ring, whereas the rotational frequencies highly depend on the ion energies, angles, and intimal positions. After stabilization, the amplifiers detect the current induced by these rings and by Fourier Transformation the signal is transformed into m/z signal [27]. Simultaneously with acquisition of the MS1 signals in the Orbitrap analyzer, the high abundant precursors are isolated and fragmented in the high pressure cell (another batch of ions, but the same precursors as analyzed on the Orbitrap analyzer) by CID. Ions are fragmented by collision with neutral molecules, such as helium or nitrogen. The MS2 spectra of the fragmented ions of the most abundant precursors are acquired in the low pressure linear ion trap. The second linear ion trap realizes this with extremely fast scan speed. Alternatively, the higher-energy collisional dissociation (HCD) cell can be used for fragmentation of the precursors [28], [27].

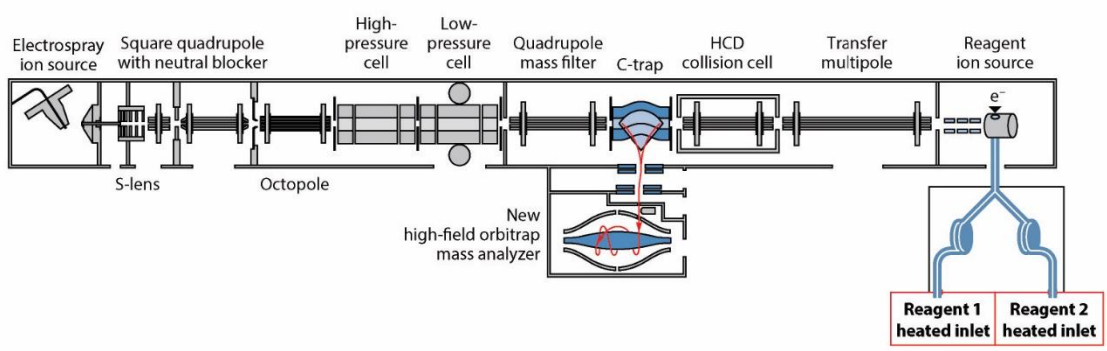


Figure 2: Schematic of the OrbitrapElite mass spectrometer. *The analyte coming from the interfaced LC is transferred to a gas phase in the electrospray ion source (left side) and thereby ionized. The ionized analyte is then guided through the S-lens, and the quadrupole and actople, before precursor selection happens in the high-pressure cell. A packet of precursors is selected and transported to the C-trap and further on the high-field Orbitrap mass analyzer (MS1 spectra). Another packet of the same precursors is at the same time again selected in the high-pressure cell and fragmented, and analyzed in the low pressure cell (MS2 spectra) [27].*

Peptides are identified by performing a peptide identification search against an *in silico* tryptic digest of a protein database, containing the proteome in which we are interested in. With the OrbitrapElite we were capable of detecting multiple thousands of phosphopeptides within a single measurement. By using label free quantification (LFQ), it was further possible to relatively quantify the intensities over the samples. However, in DDA mode only the most abundant precursors are selected for further

MS2 analysis, which leads to a negative bias towards low abundant peptides, which are less likely to be sampled. Often in DDA mode, measured precursors are blocked for a distinct time, which avoids, that always the same precursors trigger an MS2 event and further increases the number of sample analytes. This leads to a semi-stochastic sampling algorithm of the DDA methods. Due to this sampling, DDA methods lead inevitable to data sets with missing values. This becomes a problem if complex samples over a large cohort are measured [29]. In addition the quantification on MS2-level as opposed to MS1, has been shown to lead to more reproducible results [30].

Targeted proteomics

In contrast to untargeted proteomics, in targeted proteomics peptides with predefined mass windows are preselected for an MS analysis in a triple quadrupole (QQQ) MS instrument. Nowadays a commonly used targeted approach is SRM (or also called multiple reaction monitoring – MRM), which is suitable for the consistent detection and accurate quantification of specific, prior to the analysis determined sets of proteins across multiple samples [31]. It is often named the analytical “gold standard” for quantitative data analysis in proteomics. In SRM, predefined pairs of precursor and product ion masses, so called transitions, are monitored over the LC retention time, yielding a set of chromatographic traces with the retention time and signal intensity for a distinct transition. The first quadrupole (Q1) acts as filter to specifically selected predefined precursor m/z values. The second quadrupole (q2) severs as collision cell to fragment the precursor ions. A distinct predefined fragment ion of the precursor ion is filtered to its m/z and analyzed on the Q3. In SRM no full mass spectra are recorded which translates into an increased sensitivity [32].

The targeted proteomics approach of SRM enables the specific, quantitative measurement for a maximum of a few thousand transitions. To overcome these limitation, while achieving comparable specificity, another targeted approach was developed: SWATH-MS [33].

Data Independent Acquisition (DIA) via SWATH-MS

DIA is based on the sequential isolation and fragmentation of precursor windows on a first mass analyzer by further fragmentation and subsequent analysis on a second mass analyzer. In order to apply this MS measuring method, mass spectrometers with high sensitivity, rapid profiling, and high-resolution are needed [34].Thus, we made use of the recent advantages in DIA mass spectrometry, by using SWATH-MS for the precise analysis of protein abundances among a large cohort. We used the unbiased DIA method, sequential window acquisition of all theoretical MS2 spectra (SWATH-MS), which allows quantitative measurements of the proteome or phosphoproteome

on the SCIEX 5600+ TripleToF [33], [35]. In SWATH mode, the whole m/z range from 400 – 1200 m/z is separated in 64 (or 32) precursor isolation windows. For each of the consecutive MS1 precursor scans on the Q1, all contained precursors are fragmented on the q2 collision cell. This results in complex fragment ion maps for all detectable analytes in a time-resolved manner. After one cycle through all SWATH windows of the m/z range, the cycles are continuously iterated throughout the LC gradient.

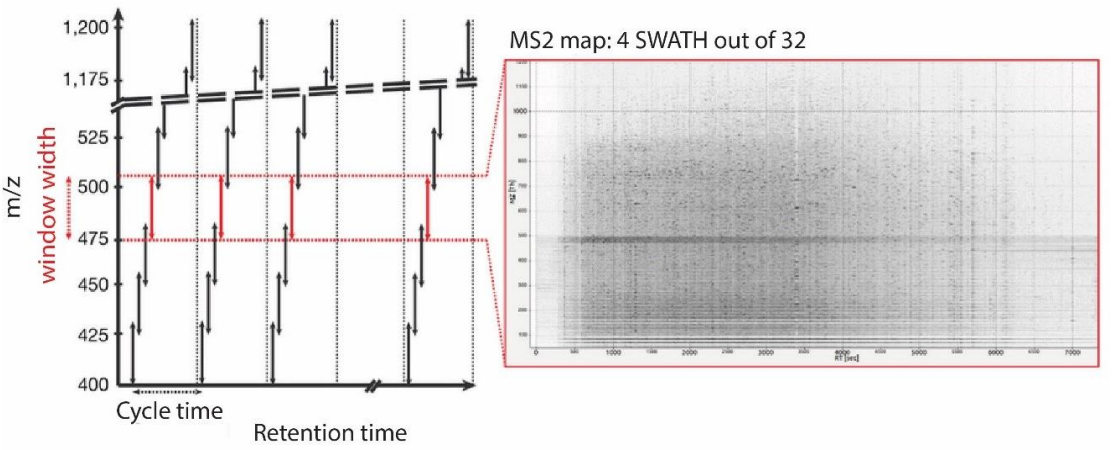


Figure 3: SWATH consists of sequential acquisition of fragment-ion spectra with precursor isolation windows (32 or 64 windows). *On the right side of the illustration, a SWATH window width is 25 m/z, which means with 32 windows the whole range from 400 – 1,200 m/z is covered. The left side, shows all fragment-ion spectra of the same isolation window aligned, which represents an MS2 map (also called SWATH). The illustration was taken from [33].*

In order to conduct in a targeted manner a quantitative analysis of SWATH-MS data, a spectral assay library, containing all peptides respectively phosphopeptides that are acquired in a sample, is required [33]. These libraries are limited by peptides that have already been detected by database searches with MS2 spectra generated usually from DDA mode analysis of the same samples on the same mass spectrometer, as the SWATH-MS measurements are conducted. The spectral library contains all transitions and the quantitative data are extracted by automatic tools like OpenSWATH [36]. The major advantage of the DIA approach is the reproducibility, as in theory, for every precursor, fragmentation spectra are recorded. This is in contrast to the DDA approaches, for which a semi-stochastic precursor selection limits the amount of precursors selected for analysis.

In the current thesis we used SWATH-MS for the reproducible analysis of the 76 liver tissue samples of the BXD mouse reference population. Prior to the analysis we had to conduct a phosphopeptide enrichment, as the measurement of low abundant proteins, like phosphoproteins, is not possible in a complex total tissue lysate sample. Thus, we generated a sample specific assay library for the quantification of the acquired SWATH-data.

Phosphoproteomics

Protein phosphorylation is one of the major PTMs which regulates many of the dynamic changes in cells. The cell uses phosphorylation and dephosphorylation events to induce fast and precise changes in protein properties. This regulation mechanism is used for example to regulate important signaling pathways, modulate the activity of metabolically active enzymes, or turning whole metabolic pathways on and off [37]. Through protein kinases the cell controls every basic process, including metabolism, growth, cell division and differentiation, organelle trafficking, and immune responses [38].

MS is the method of choice to accurately identify and quantify phosphorylated peptides. In order to measure the low abundant phosphoproteins, or tryptic digested phosphopeptides, several enrichment methods were developed [39]. In principle all enrichment strategies aim to enrich in a specifically and sensitively manner phosphopeptides, which harbor in their peptide sequences phosphoserines, phosphotyrosines, and phosphothreonines. The most predominantly used enrichment methods are immobilized metal affinity chromatography (IMAC) and metal oxide affinity chromatography (MOAC). IMAC uses metal cations as affinity agents for the negatively charged phosphate groups. The metal ions are immobilized via chelation on materials like magnetic or silica-based beads. At the moment the most prevalent used material is Ti^{4+}-IMAC [40], [41]. The MOAC uses the affinity of metals in metal oxide matrixes to the phosphate group. The most popular among this group of enrichment materials is TiO_2 [42]. A wide variety of other methods, including peptide immunoprecipitation with phospho-specific antibodies, were proposed. These other enrichment methods play a minor role in the field of phosphoproteomics [43]. Further, one should be aware, that enrichment introduces the most variation of any step in a standard phosphoproteomics workflow [42].

One of the advantages of MS-based phosphoproteomics is the ability to offer sitespecific resolution for systems level phosphorylation sites. One issue is that in large-scale phosphoproteomics experiments, most of the current peptide identification search tools do not assign the phosphorylation sites with a confidence value. Therefore, dozens of tools exist to score the phosphosite localizations in the peptide sequence, which were assigned via a database search. These tools often refer to the MS2 spectra to gain confidence for the correct site localization. In fact, around 20 – 40 % of identified phosphopeptides in standard phosphoproteomic data are lost due to the reason that the phosphorylation site is not confidently assigned [42]. In the current study LuciPHOr2 was used for the validation of correctly assigned phosphopeptides [44].

The phosphoproteome of mouse liver tissue was investigated in several previous studies. In a large-scale phosphorylation analysis of mouse liver around 5635 non redundant phosphorylation sites of 2328 proteins were identified. For this study a two-step phosphopeptide enrichment was performed, including first SCX followed by iron IMAC enrichment of the fractions. In addition immunoprecipitation of phosphotyrosine peptides was performed. The data were acquired via an Orbitrap LTQ with LC-MS/MS measurements in DDA mode [43]. In another study of mouse liver tissue, 15'000 phosphosites (MaxQuant average localization probability over 0.99) were analyzed. For the experiment, two sets of mouse liver tissue samples, one perturbed with insulin, the other an unperturbed as, were measured with an LC-MS/MS on an Orbitrap LTQ velo. Standards labelled with stable-isotope labeling with amino acids in cell culture (SILAC) were spiked to the samples [45]. A study using reductive demethylation labelling to investigate the mouse liver tissue proteome we were able to identify 7400 phosphorylation sites of 2300 phosphoproteins [46].

Due to their properties, phosphopeptides pose special challenges in their MS-based analysis [39]. As phosphoproteins are of low abundance, a specific chemical enrichment of the phosphopeptides must be conducted. Various enrichment strategies were developed, also by combining multiple enrichment strategies with fractionation to increase the coverage of the phosphoproteome. As the phosphopeptide enrichment highly increases the variability, the decision which technique is applied, can highly influence the results of a large-scale study. Further, the measurement method, the used mass spectrometer, and the length of the gradient also have a huge impact on the amount of acquired phosphopeptides. Other important issues deal mainly with the data analysis, for which several peptide identification scores, computational tools and pipelines, the filtering of site localizations, and the statistical analysis, can be used. Overall, each of these steps needs to be considered for the MS-based analysis of the phosphoproteome as each step have the probability to lead to wrong biological statements or weak phosphoproteomics data.

1.5. Aim of the study

The overall aim was to quantify reproducibly phosphopeptides in liver tissue across 40 strains of the BXD population, which were treated with two different diets. In order to be able to do this, the first specific aim was to establish a well working single-step based phosphopeptide enrichment strategy. Therefore several beads and buffer combinations were tested and the best performing was further optimized for the enrichment of mouse liver tissue samples. The second aim was to combine the improved approach with SWATH-MS and acquired a test dataset, to evaluate the variability for this combined data generation workflow. Further, the data set should be

used to compare the performance of three different assay libraries. The third aim was to use the final enrichment strategy and the best performing SWATH-assay library, to acquire the phosphoproteome of the BXD strains. Finally, the obtained quantitative data should be used for the discovery of phosphoprotein-QTLs.

2. Methods

2.1. Experimental part

2.1.1. Peptide preparation

Lysis of mouse liver tissue

The frozen mouse liver tissue samples were shortly thawed and approximately cut into 50 mg pieces, for further processing and stored at -80°C until they were lysed. For the pressure cycling technology tissue lysis, the 50 mg portions were further cut into approximately 6.1 mg pieces and mashed up with a sterile scalpel and a sterile syringe-needle for better lysis efficiency.

Conventional lysis with a glass dounce homogenizer

The 50 mg of frozen mouse liver tissue were placed into the 15 mL glass dounce homogenizer with a tight glass pestle (Tight Pestle A) and 4 mL of RIPA-M buffer containing the phosphatase and protease inhibitors were added. The inhibitor cocktail consisted of a final concentration of 10 mM NaF, 10 mM Sodium Pyrophosphate, 5 mM Glycerol 2-phosphate and Roche Protease inhibitor pill (1 pill is considered a 50x Stock solution, used to a final concentration of 1x in the lysis buffer). After 10 strokes the lysis buffer, containing the lysed protein and cell compartments, were separated to four 2 mL Eppendorf tubes and centrifuged at 20’000 g at 4 °C for 15 minutes. The supernatant that contained the lysed proteins was collected in a new tube and the pellet, which consisted of cell debris and cell compartments which were insoluble but maybe still contain some proteins was further processed. This pellet was resuspended in 400 µL Urea-T buffer which contained phosphatase and protease inhibitors. The resuspension was done by 10 minutes shaking at 1400 rpm at room temperature and 10 minutes sonication in an ice cooled sonication bath. Subsequent the samples were centrifuged at 20’000 g at 4 °C for 10 minutes. The RIPA-M and Urea-T supernatants containing the proteins were combined. The pellets containing insoluble proteins or tissue parts were discarded. The protein content in the supernatant was measured by BCA assay to estimate the lysis efficiency. The protocol was adapted from existing protocols as described [47] and [43].

Pressure Cycling Technology (PCT) lysis

The PCT lysis was conducted as described [48]. Due to high amounts of tissue per Microtube, the cycling times and enzyme concentrations of the original protocol were adapted. Per Microtube 6.1 mg of tissue was lysed and the protein content was inferred from previously done BCA measurements of the conventional lysis. The PCT lysis was conducted at 33 °C with 60 cycles, each consisting of 50 s at 45’000 psi and

10 s of relaxing pressure time at standard pressure. The lysis was accomplished in an 8 M urea in 0.1 ammonium bicarbonate (ABC) lysis buffer containing a mixture of phosphatase inhibitors. The phosphatase inhibitor cocktail was altered compared to the original protocol and the same mixture as used for the conventional lysis was used. The inhibitor cocktail consisted of 10 mM NaF, 10 mM Sodium Pyrophosphate and 10 mM 2-Glycerophosphate. The protein reduction with tris(2-carboxyethyl)phosphine (TCEP) and alkylation agent iodoacetamide (IAA) were added as a mixture to the lysed protein and incubated at 1000 rpm at 37 °C for 30 minutes. The final concentrations were 10 mM for IAA and 40 mM for TCEP. Subsequently, the enzymatic treatment was conducted again in the same Microtubes by adding Lys-C and trypsin. The cycling time for Lys-C was extended to 60 minutes corresponding to 60 cycles again at 33 °C. For the tryptic digestion the cycling time was set to 120 minutes or 120 cycles. The enzyme-to-substrate ratio for Lys-C was 1:200 and for trypsin 1:100. The Urea concentration was diluted with 0.1 M ABC for the Lys-C treatment to 6 M and for the trypsin digestion to 1.6 M.

BCA assays

To calculate the protein content of the various lysis, BCA assays were conducted. Protein concentration was measured on a 96-well plate reader against a Bovine serum albumin (BSA) standard curve. The assays were conducted in 96 well plates and 1 to 2 µL of sample per each lysis were used to determine the protein concentration. Before measuring on a microplate reader at 562 nm wavelength, the BCA assays were incubated for 45 minutes at 37 °C. For each sample the average of four replicates was calculated to infer lysis efficiency and calculate the lysis volume for the overnight acetone precipitation of the desired protein amount of the experiment.

Acetone precipitation of the total lysed protein

If the lysis was done by the conventional method the protein was purified by an overnight precipitation in ice cooled acetone. Typically for the testing experiments 0.5 mg of protein was precipitated whereas for the “aging” and the main “BXD-mouse reference population” experiments 1.5 mg of protein was precipitated. The volume alters due to resulting protein concentrations which depends on the lysis efficiency of each lysis. Six times the volume of acetone was added to the protein solution and the precipitation was conducted overnight at -20 °C. This step was conducted in 2 mL Eppendorf tubes which led to multiple tubes per sample. After overnight incubation, the precipitated total cell lysate proteome was centrifuged at 20’000 g at 4 °C for 15 minutes to gain the pellet, which should contain proteins. The Acetone was removed by pipetting and the samples were shortly dried on ice at room temperature and air.

Reduction and Alkylation

The resulting pellets from the acetone precipitation were dissolved in freshly prepared 300 µL of 8 M Urea in 0.1 M ABC. For fully resuspension of the pellet the samples were incubated on a shaker at 1400 rpm at room temperature for 10 minutes, followed by 10 minutes sonication in an ice cooled sonication bath. Afterwards the samples were incubated again on a shaker at 1000 rpm at room temperature for 5 to 10 minutes before the reduction reagent was added. For reduction of the disulfide bridges TCEP from a 50 mM stock was added 1:10 to gain a final concentration of 5 mM TCEP. For a sufficient reduction the samples were incubated at 1000 rpm at 37 °C for 30 minutes. For alkylation, freshly prepared 400 mM IAA in 8 M Urea in 0.1 M ABC was added to a final concentration of 10 mM IAA. Subsequently, the samples were incubated at 1000 rpm at 25 °C for 45 minutes in darkness.

Enzymatic treatment

Before the enzyme Lys-C was added, the samples were diluted with 0.1 M ABC from 8 M Urea to 5 M Urea. For the conventional lysis an enzyme-to-substrate ratio of 1:150 for Lys-C and 1:75 for trypsin was used. The Lys-C incubation time was 4 – 6 hours at 37 °C on the thermoshaker with gentle agitation at 600 rpm. Following the Lys-C treatment, the Urea concentration was further lowered to 1.5 M Urea by adding 0.1 M ABC. To the diluted samples, trypsin was added and the samples were incubated at 600 rpm at 37 °C overnight.

C18-purification of peptide

Before the first C18 purification, each sample was acidified with 5 % TFA (Trifluoroacetic acid) in H_2O to a pH range of 2 – 3, which corresponds to the pH of the equilibration buffer. Each sample was purified over a column with a total peptide binding capacity of 1 – 5 mg. C18 columns were first washed two times with 2 mL methanol and in a following step again two times with 0.1 % TFA in 80 % acetonitrile in H_2O. Three times 2 mL of the equilibration buffer 0.1 % TFA in 2 % acetonitrile in H_2O was added. On the equilibrated columns the samples were added, which were beforehand centrifuged at 20’000 rpm at 4 °C for 10 minutes to remove any precipitated substances. The flow-through of the samples were reloaded and because the precipitation and enzymatic treatment was conducted in several tubes for the same sample, in this step the samples were combined. The bound peptides, including the phosphopeptides, were washed five times with 2 mL of equilibration buffer to reduce the amount of salt and other contaminants and to clean up the peptides. Elution of the bound peptides was achieved by increasing the amount of organic solvent, by adding three times 1 mL of a 0.1 % TFA in 50 % acetonitrile elution buffer on the column. The peptides were dried under vacuum at 45 °C, and stored in −20 °C

before the phosphopeptide enrichment step. For the whole cell lysate, the samples were ready to be dissolved in MS buffer.

2.1.2. Phosphopeptide enrichment strategies

As there was no generally accepted phosphoenrichment protocol for murine liver samples, a set of experiments was performed to test the best performing beads and buffer combination for mouse liver tissue. Because of the high number of samples that need to be analyzed, only protocols with a single enrichment step without fractionation were taken into consideration. Hence three phosphopeptide enrichment beads were tested in combination with four varying loading buffers (Table 1). For beads, the commercially available Titanium dioxide (TiO_2) affinity chromatography, and magnetic MagReSyn® Ti-IMAC micro particle beads, were compared to monodisperse microsphere-based Titanium (VI) immobilized metal ion affinity chromatography (Ti^{4+}-IMAC) beads. For the TiO_2 beads the lactic acid, phthalic acid and 6 % TFA in 80 % acetonitrile buffers were considered for the comparison. The instruction of the magnetic MagReSyn® beads suggests to use glycolic acid buffer as loading buffer. In addition the 6 % TFA in 80 % acetonitrile buffer was used. For the "beads and buffer" combinations experiment, the amount of starting material, digested protein, was per replicate 500 µg. The dried and purified peptides were dissolved in 400 µL of the various loading buffers.

Table 1: Beads and buffer combinations used for the comparison experiment. *The aim of the experiment was to identify the best performing combination regarding the unique identified phosphopeptides per sample.*

Beads	Buffers
TiO_2 beads to protein ratio 2:1	6 % TFA in 80 % acetonitrile Lactic acid buffer Phthalic acid buffer
Ti^{4+}-IMAC beads to protein ratio 3:1	6 % TFA in 80 % acetonitrile
MagReSyn® (Ti-IMAC) Beads to protein ratio 2.5:1	6% TFA in 80 % acetonitrile Glycolic acid

TiO_2 enrichment procedure

The protocol for the enrichment of phosphopeptides with TiO_2 beads is based on the "Phosphopeptide enrichment using titanium dioxide (TiO_2) affinity chromatography" protocol of the Heck group (Prime-XS, Protocol – Access Site: IMP, UCHP and UU). The protocol is suitable for 1.5 – 3 mg of starting material, which was defined as protein measured via BCA before digestion to peptides. A stock solution, for the beads of 125 mg TiO_2 was resuspended in 20 ml HPLC grade H_2O was sufficient for 100 phosphopeptide enrichment procedures. The 500 µg digested, cleaned up, and dried peptides were resuspended in 400 µL loading buffer and were dissolved on a

thermoshaker for 10 minutes at 1400 rpm at room temperature. The peptides were sonicated for 10 minutes in an ice cooled sonication bath and centrifuged for another 10 minutes at 20'000 g at room temperature. To prepare the beads, 200 µL of the TiO_2 resin, which corresponds to 1.25 mg of beads, were transferred to 2 mL Eppendorf tubes. The TiO_2 resin was centrifuged for 1 minute at 200 g and the supernatant was discarded. All centrifugation steps from here on were done like this. The resin was washed twice with 360 µL methanol and once with 360 µL loading buffer. After centrifugation and removal of the supernatant 360 µL of loading buffer were added to the beads, which were afterwards incubated on the head-over-end rotator at 40 rpm for 15 minutes at room temperature. Meanwhile the in loading buffer dissolved samples were centrifuged at full speed for 10 minutes at room temperature to avoid adding undigested proteins on the beads. At the following step the sample was added to the beads. The sample with beads were incubated for 1 hour on the head-over-end rotator with the same conditions as the equilibration step. This time the flow-through was collected. The beads with the bound phosphopeptides were washed twice with 280 µL loading buffer, twice with 280 µL 0.1 % TFA in 80 % acetonitrile, and twice with 280 µL 0.1 % TFA in 50 % acetonitrile. Finally the beads were washed twice with 280 µL 0.1 % TFA and due to the higher solubility in the more polar buffer, the centrifugation speed was increased to 600 g. The phosphopeptides were eluted by adding two times 150 µL of 0.3 M ammonium hydroxide solution (pH 10.5 – 11) to the beads and treated as in the previous steps to separate beads and supernatant. Two previously prepared tubes per sample, each containing 15 µL of 15 % TFA in HPLC grade water were used to instantly neutralize the supernatant of the elution step. This was necessary to avoid dephosphorylation due to harsh basic conditions. The resulting elution mixtures were pipetted together and the pH was controlled to be in the pH range of 2 – 3, which corresponds to the pH of the equilibration buffer of the following C18 purification step. The pH was set either with 0.3 M ammonium hydroxide solution or 15 % TFA into the range of the loading buffer of the C18 desalting step. The samples were transferred for 10 – 15 minutes on the speed-vac at 45 °C to get rid of remaining ammonium and acetonitrile before they were centrifuged at full speed for 1 minute to separate any remaining beads from the supernatant. Finally the samples could be loaded on the equilibrated C18 for purification.

Magnetic MagReSyn® Ti-IMAC enrichment procedure

The phosphopeptide enrichment with the MagReSyn® Ti-IMAC was accomplished as described in the protocol which came along with the product description [49]. Some few adaptions were made to the protocol to provide similar conditions as for the other enrichment procedures. The dried peptides were resuspended in 400 µL loading

buffer, either glycolic acid as in the original protocol, or with 6 % TFA in 80 % ACN. The resuspension was achieved by 10 minutes shaking on the thermoshaker at 1400 rpm at room temperature, another 10 minutes sonication in an ice cooled sonication bath and 10 minutes centrifugation at 20'000 rpm at 4 °C. If in the protocol "gentile agitation" was required, the samples were resuspended at 750 rpm at room temperature on the thermoshaker.

Ti^{4+}-IMAC enrichment procedure

The monodisperse microsphere-based immobilized metal ion affinity chromatography (IMAC) beads were activated with $TiCl_4$ solution. The Ti^{4+} ions bound to the IMAC via immobilization on a linker which contains phosphonate groups at the end. The beads were activated as described [41]. The beads activation protocol had been further improved by our collaboration partners Houjiang Zhou and Minglang Ye, Cambridge University, Systems Biology Centre and in our laboratory from Tiannan Guo and Yi Zhu. The amount of Ti^{4+}-IMAC beads which had to be activated were 200 mg for the "beads and buffer combinations" experiment, 400 mg (2 x 200 mg) for the "amount of starting material and beads ratio" and the "aging" experiment. Another 1200 mg (2 x 300 mg + 2 x 400 mg) of beads were activated and pooled together after activation for the "BXD reference population" experiment. The amount of beads in the protocol was given by 100 mg and was changed according to the amounts needed for each of the experiments. The required amount of beads incubated over night at room temperature with $TiCl_4$ solution in the hood. The solution was stirred to avoid precipitation. The solution which contained the dispersed overnight activated beads was separated for washing into 2 mL Eppendorf tubes. The beads were centrifuged at 20'000 g for 5 minutes to form a pellet. The supernatant was discarded and the beads were washed three times with 2 mL of 0.1 % TFA in 30 % acetonitrile. Between each centrifugation step, the samples were shook at 1000 rpm for 1 minute at room temperature on a thermoshaker. The washed beads were dissolved in 0.1 % TFA in 30 % acetonitrile to a final concentration of 10 mg mL^{-1}. If the activation could only be accomplished in batches, the batches were mixed together. Activated and washed beads could be stored in the fridge for 3 months.

The enrichment protocol with Ti^{4+}-IMCA beads differs only slightly from the enrichment protocol of the TiO_2 beads mentioned above and therefore only the adaptions are mentioned. All centrifugation steps were made at higher speed due to the increased solubility of the Ti^{4+}-IMAC beads compared to TiO_2. For the washing steps centrifugation with up to 1400 g was performed. The final single washing step with 0.1 % TFA in HPLC grade water was done at 3800 g. Due to the higher beads to starting amount ratio and the with difficulty dispersible beads, between each of the

centrifugation steps of the protocol, the samples were incubated for 1 minute at 1000 rpm at room temperature on the thermoshaker. Furthermore, the methanol washing step in the beginning of the protocol was obsolete because the beads were already in the 0.1 % TFA in 30 % acetonitrile which was similar to the loading buffer. The different experiments were performed with changing amounts of starting material, different volumes of loading buffer and alteration of the beads to starting material ratio. Thus for each experiment the exact parameters are mentioned in the experimental design in the results chapter.

C18 purification of phosphopeptides

To ensure MS compatible sample quality, all phosphopeptide enriched samples were purified over a reversed phase C18 column for a second time after performing the phosphoenrichment. The amount of phosphopeptides after enrichment was in the range of 5 – 15 µg. Therefore the Ultra Micro Spin Column with 2-100 µl loading volume with a capacity of 5 – 60 µg were used. For all steps the centrifugation speed was 500 g and the volumes of the washing and equilibration buffers were 200 µl. The cartridges were first washed two times with methanol, followed by two times with a 0.1 % TFA in 80 % acetonitrile buffer, and three times with the equilibration buffer of 0.1 % TFA in 2 % acetonitrile. The flow-through of the samples were loaded a second time on the columns and were than collected. After the sample bound the C18-resin, it was washed five times with the equilibration buffer. As elution buffer 0.1 % TFA in 50 % acetonitrile was added three times with varying volumes. For the first two times 50 µl were added followed by the final elution, which was done with 100 µl. The eluted phosphopeptides were dried under vacuum at 45 °C on the speed-vac and were stored at -20 °C until they were measured.

Resuspension in MS buffer

Phosphopeptides were dissolved in MS buffer, which consists of 0.1 formic acid (FA) in 2 % acetonitrile in HPLC grade H_2O and iRT-peptides 1:20 (v/v). To keep the amount of iRT-peptides for all samples constant the iRT-peptides were typically added to the MS buffer. The exact amount of MS-buffer the samples were dissolved can be seen in the experimental design graphs.

2.1.3. Mass Spectrometry data acquisition

MS-samples with suboptimal purity or quality can lead to various problems during the measurements, including column blockage of the LC, interruption and spitting of the ion spray, and accumulation of contaminations within the mass spectrometry and detectors. The sample quality was therefore controlled by measuring first on an LC-LTQ system, which is exclusively used for this purpose, in order to avoid downtimes of high-end devices. Especially selective enrichment techniques, such as

phosphopeptide enrichment, tend to result in more problematic samples due to possible enrichment of phospholipids, incomplete separation of enrichment beads or leakages of bead particles, such as metal particles, due to harsh treatment of the sphere beads. Besides the previously mentioned quality assessment, the LTQ allowed an initial assessment on the success of the phosphoenrichment.

The samples were analyzed on an LTQ Orbitrap XL mass spectrometry, which was coupled to a Thermo EASY-nLC II system. A 60 minute gradient from 5 – 35 % buffer B (98 % acetonitrile in 0.1 % formic acid in HPLC grade H_2O) was used with an additional hold for 5 minutes at 35 % buffer B and 5 minutes at 100 % of buffer B to elute strong bound peptides from the column. The gradient was against buffer A (2 % acetonitrile in 0.1 % formic acid in HPLC grade H_2O) with a flow rate of 300 nL min^{-1}. The MS1 scans were generated in a range from m/z 150 to 2000. The MS1 Automatic Gain Control (AGC) was $3x10^4$ with a maximum of 50 ms accumulation time. The fragmentation of the precursor ions, was performed in CID mode with 35 % normalized collision energy for an maximal activation time of 30 ms. For the MS^n spectra the AGC was $1x10^4$ with a maximal filling time of 100 ms. It was sufficient to inject 1 µL of samples dissolved in MS buffer. In between the injections of samples, GluFib-standards were measured to monitor the stability of the LC-MS system. The reversed phase C18 analytical column was home-packed with ProntoSIL C18 AQ resin with 3 µm particle size and 200 Å pore size (column dimensions: 11 cm x 75 µm).

DDA measurements on the OrbitrapElite system

Pre-tested phosphopeptide enriched samples of the “beads and buffer combinations”, “amount of starting material and beads-ratio”, and the “aging” experiments were measured with the OrbitrapElite Hybrid Ion Trap-OrbitrapMass Spectrometer with a 180 minutes gradient from 5 – 15 % buffer B (98 % acetonitrile in 0.1 % formic acid in HPLC grade H_2O). The gradient was against buffer A (2 % acetonitrile in 0.1 % formic acid in HPLC grade H_2O) with a flow rate of 300 nL min-1. The phosphopeptide enriched samples were first separated on an ultra-high pressure LC system (Thermo EASY-nLC 1000), which was coupled to the OrbitrapElite. The separation was conducted on a Thermo PepMap analytical column with 150 mm length, 75 µm inner diameter, and 3 µm particle size. The MS1 survey scans ranged from m/z 350.00 to 1600 with a survey scan resolution of 120’000. The MS1 AGC was $1x10^6$ with a maximum of 200 ms accumulation time. The measuring method was top 15 high abundance peptides signal for MS2 with a dynamic exclusion for 30 secs. Single charged precursor ions and those of unknown charge states were excluded for MS2 triggering. Fragmentation was performed in CID mode with 35 % normalized collision

energy with a 2.5 isolation with. For the MS spectra the AGC target was $1x10^4$ with 100 ms accumulation rate.

All samples measured on the OrbitrapElite were diluted in 20 µL of MS-buffer, including iRT-peptides 1:20 (v/v). For the "beads and buffer combinations" the starting material for all samples was 0.5 mg and thus 4 µL of each sample were injected. Due to fiddling of experimental parameters, such as the amount of starting material, the injection volume changed for the conditions of the "amount of starting material and beads-ratio" experiment. For an overview over the injection volume see Table 2.

Table 2: The starting material scaled indirectly to the injection volume. *To avoid overloading of the column, the injection volumes were altered. Between 1.0 mg and 0.5 mg starting material the scaling was not changed, because the maximal amount of injection volume for the sample loading loop of the OrbitrapElite is 5 µL.*

Starting material [mg]	Injection volume [uL]	Experiment
0.5	4	Beads and buffer combination, amount of starting material and beads ratio, HeLa control
1	4	amount of starting material and beads ratio
2	2	amount of starting material and beads ratio
4	1	amount of starting material and beads ratio
1.5	3	Aging, BXD-mouse reference population

For the phosphopeptide enriched samples of the "aging" experiment 1.5 mg of starting material were used for digestion and phosphopeptide enrichment and therefore 3 µL were injected on the OrbitrapElite. After two consecutive sample runs, a GluFib-standard was measured, which helped to avoid carry over between samples, column blocking, and assessment of reproducible data acquisition conditions.

TripleToF MS analysis in DDA mode and SWATH mode

The data independent acquisition method SWATH-MS enabled us to reproducibly measure the proteome or phosphoproteome in a quantitative manner within large sample series. In order to use this approach, an SWATH assay library is necessary which was constructed out of DDA measurements of the same samples on the same MS instrument as the subsequent SWATH analysis.

The DDA measurements were acquired on the TripleToF 5600+ from Sciex interfaced with a NanoLC-Ultra 2Dplus from Eksigent. Precursor selection on the MS1 was performed with the top 20 method. This method selected the 20 most intense peptide precursor ions, with an accumulation time set to 250 ms. MS1 scans covered a precursor range from m/z 360 to 1460. The fragmentation and ionization of the

precursors mimicked the fragmentation of the later conducted SWATH-MS measurements. Fragmentation was performed in CID mode with collision energy for each SWATH window. Single charged precursor ions and those of unknown charge states were excluded for MS2 triggering. Selected precursor ions were measured on the MS2 in high-sensitivity mode, for which the accumulation time was set to 150 ms per scan. The gradient for each DDA measurement was 120 minutes long, starting at 98 % buffer A (2% acetonitrile in 0.1 % formic acid in HPLC grade H_2O) and 2 % buffer B (98 % acetonitrile in 0.1 % formic acid in HPLC grade H_2O) to 30 % buffer B with a flow rate of 0.3 µL min^{-1}. For the phosphopeptide enriched samples 2 µL were determined as the optimal injection volume.

For the DIA acquisition the measurement set up the TripleToF 5600+ was altered. The precursor peptide ions were selected due to 64 variable precursor ion isolation windows. The variable windows were optimized for human samples and the windows were shortened in m/z regions with high amounts of precursors. The total cycling time for one cycle of precursor selection and MS/MS scans was 3.5 seconds. The total precursor selection range was from m/z 400 to 1200 and was separated in 64 variable precursor windows. At each step a complete, multiplexed fragment ion spectrum of all precursors present in one window was acquired. The length of the windows varied from m/z 7 to 90, with smaller windows in the lower m/z region and larger windows at the higher region of the total m/z ratio. The accumulation time of the ToF-MS instrument was 250 ms for the MS1 and 50 ms for the MS2 scans and for each of the 64 variable isolation windows, respectively, causing a total cycle time of 3.5 s. The gradient was shortened to 90 minutes from 2 % buffer B to 30 % buffer B. β-Gal-standard was measured after two consecutive samples, in order to continuously calibrate the masses and monitor the performance of the mass spectrometer.

2.2. Bioinformatics and biostatistical part

2.2.1. Computational tools for proteomic data analysis

Data conversation and annotation

Acquired raw format mass spectrometry data were converted to the open standard format mzXML using an automated pipeline, which is implemented within the laboratory data storage facility. The automated pipeline used the ProteoWizard [50] converter (version 3.0.7494) and for the SWATH-MS data, which had been measured later, an updated version of the ProteoWizard converter (version 3.0.5533). By a simple unique identifier, the raw and converted files are stored in openBIS (open Biology Information System) [51] where they were annotated e.g. with sample origin, lysis treatment, enrichment method, loading buffer and various other information.

Peptide identification searches

Within our laboratory an intuitive graphical user interface which is named iPortal, combines the trans-proteomic pipeline and various search engines for peptide and protein identifications of MS/MS data sets. The trans-proteomic pipeline is a collection of tools which were combined to create a reproducible and easy to handle software package for proteomic research. The TPP was developed at the Seattle Proteome Center (SPC) [52] and is partly integrated in the iPortal platform [53]. Within the iPortal platform commonly used proteomics tools were integrated into a user friendly interface. The workflows integrated contain various programs and algorithms developed in our laboratory and allow users to use iPortal for various complete workflows, including peptide identification searches with subsequent scoring of the search results (e.g. mProphet). Further workflows deal with label free quantification (LFQ) and the integration of the OpenSWATH pipeline allows SWATH library construction and OpenSWATH searches and quantification. The several workflows within iPortal are under ongoing development and new tools and workflows of our laboratory are permanently integrated and actualized.

The spectra acquired with DDA MS were analyzed via the iPortal TPP search and identification workflow, which combines the trans-proteomic pipeline with several search engines and scoring algorithms. Varying combinations of the three search engines Omssa, X! Tandem and Comet were used for the identification searches in iPortal. Later, the results were compared to MaxQuant output and the best performing engine combination were used for further analysis of the “aging” and “BXD-mouse reference population” experiments. All named search engines have in common, that they identify peptides by searching MS/MS spectra against hypothetical spectra generated from sequences present in the protein sequence databases. The search engines Omssa (Open Mass Spectrometry Search Algorithm) [54] and X! Tandem [55]were enabled in iPortal as default. Another search was conducted using the open source tandem mass spectrometry (MS/MS) sequence database search engine Comet. [56]. The third identification search was a combination of Omssa, X! Tandem and Comet. For all analysis in iPortal the static modification carbamidomethylation at cysteine was chosen. As variable modifications were chosen phosphorylation at serine, threonine, and tyrosine, and the variable modification oxidation at methionine. The mass tolerances were set to 50 ppm for precursor ions and 0.04 Da for fragment ions. For the digestion type, trypsin was chosen and 1 missed cleavage was allowed. The peptides were searched against an enriched murine UniProtKB/SwissProt protein database (monthly updated) using a target-decoy approach. The decoy proteins are amino acid sequences which are not found in nature and are later on used to calculate the false discovery rate (FDR). The decoy sequences were generated by inverting the

natural protein sequences. The identified peptides were scored using PeptideProphet, iProphet and ProteinProphet. PeptideProphet is a tool for statistical validation of MS/MS search engines spectra-to-peptide sequence assignment and was originally developed at Seattle Proteome Center (SPC) and is part of the TPP. The iProphet or InterProphet further refines the PeptideProphet results. ProteinProphet was developed at SPC and calculates probabilities for protein identifications based on MS/MS data by using the calculated values of the PeptideProphet. Spectral counts and peptides for ProteinProphet were filtered at FDR.

Label Free Quantification (LFQ) using OpenMS

The automated pipeline integrated in iPortal consists of parts of the OpenMS software framework. The computational workflow allows quantification of LC-MS/MS samples without prior labelling [57]. The phosphopeptide enriched “aging” samples were measured in DDA mode on the OrbitrapElite and the LFQ of the protein and peptide abundances was carried out in iPortal using the improved OpenMS tool. As an input for the LFQ the centroided profile mzXML data of each run and the pep.xml, the output of the iPortal TPP peptide identification search were used. For the peptide identification search the combination of three search engines were used as described earlier. The analysis output is used to compare the variation of the intensities of the identified phosphopeptides of the LFQ of the DDA measurements of the OrbitrapElite with the variation of the intensities of the same samples measured by SWATH-MS. For the LFQ the “LFQ Default – READONLY” parameters in iPortal with the improved OpenMS default settings and quantification on the MS1, was used [57].

MaxQuant – peptide identification search and LFQ

For an independent comparison the two testing experiment (“beads and buffer combinations” and “amount of starting material and beads radio”), and the “aging” datasets were analyzed too with the quantitative proteomics software package MaxQuant [58]. MaxQuant allows peptide identification as well as various quantification methods including labelled (e.g. SILAC) and LFQ. As peptide search algorithm the implemented Andromeda search engine was used [59]. The search was conducted against the murine UniProtKB/SwissProt protein database, which was downloaded as fasta file (03.09.2015). For all MaxQuant searches and quantifications the parameters were kept the same. Carbamidomethylation at cysteine was set as fixed modification. As variable modification were set oxidation at methionine and phosphorylation at serine, threonine and tyrosine. Commonly known laboratory contaminants were included within the search (marked as CON_ in the output and were filtered in the R-script based analysis). These for example may contain different forms of keratins or other abundant proteins [60]. The minimum peptide length

considered for the output was seven amino acids. As MaxQuant uses also a target-decoy based search approach, the decoy generation was achieved by reversing the target sequences of the protein targets within the database. The FDR was set to 0.01 to enable a fair comparison to the iPortal output. Razor proteins, re-quantification and "match between runs" was enabled. The alignment of the spectra allowed MaxQuant to search for peaks in other spectra in a two minutes retention time window if it could identify in any other spectra a peptide peak. The alignment time window as set to 20 minutes. The identified phosphopeptides within the first search were filtered again by an FDR of 0.01 before they were saved within a single Phospho(STY) file. The peptide level and the Phospho(STY) output were taken into consideration for further R-script based analysis. For the remaining parameters the default input was used.

2.2.2. Construction of phospho-PTM specific SWATH assay libraries

As SWATH-MS uses a targeted data extraction approach for peptide identification out of the multiplexed SWATH-MS/MS spectra, one needs to have or construct an assay transition library. If once such an SWATH assay library was constructed for a specific type of samples the library can be used for this type of samples again. Thus, for the whole tissue lysate the unpublished whole lysate mouse liver tissue SWATH assay library of Yibo Wu was used. For the phosphopeptide enriched mouse liver tissue samples a phospho-SWATH assay library was constructed. The single steps of the library construction followed the protocol as described [61]. The protocol was changed by using LuciPHOr2 to calculate a site localization probability and sort out phosphopeptides with a low site localization score. This approach of constructing a phospho-SWATH assay library was first used by Peter Blattmann for phosphopeptide enriched human cell lines (Peter Blattmann, unpublished).

DDA measurements and peptide identification for the phospho-SWATH assay library

The 12 samples of the aging experiment and 19 pooled samples of the BXD-mouse reference population were measured in DDA mode on the TripleToF MS. The samples were dissolved in MS-buffer containing iRT-peptides in the ratio 1:20 (v/v). Each of the 76 samples of the BXD-mouse reference population was dissolved in 13 uL MS-buffer and 2 uL of each sample were used for pooling. In total four samples were pooled together leading to 8 uL volume per pool sample. If the samples with both feeding strategies of one mouse were in the dataset they were pooled together. The four BXD-mouse samples with only one condition were pooled together. The pooling of samples was performed in order to reduce the number of injections but still have every sample measured. Peptide identification searches were done vie the iPortal platform using as search engines X! Tandem, Comet and Omssa. The searches were

done against an enriched murine UniProtKB/SwissProt protein database (January 1, 2016) using da target-decoy approach. The murine database was automatically enriched with contaminants (like kreatin, often found in human skin or hair) and decoy proteins. Carbamidomethylation at cysteine was set as static modification. Oxidation at methionine and phosphorylation at serine, threonine and tyrosine were set as variable modification. The mass tolerance of the precursor ion was set to 50 ppm. The fragment mass tolerance was set to 0.1 Da. Tryptic peptides with maximal 1 missed cleavages were allowed. The search results were processed via the in iPortal implemented TPP and scoring algorithms of PeptideProphet, iProphet and ProteinProphet were used. Peptides were filtered by a ProteinProphet FDR of 0.01.

LuciPHOr2 – phosphosite scoring

The current state of the art peptide search engines were designed to identify typically unmodified peptides and therefore have difficulties differentiating between phosphopeptides with different phosphosite localization. The iPortal output assigns the phosphate group to the most probable peptides but especially for multiphosphorylated peptides, or two possible phosphorylation sites next to each other, the search engines have difficulties to find the most probable of the different possible phosphorylation site. Luciphor2 was developed to perform PTM-site localization on MS/MS data by making use of the PeptideProphet search result and the original spectra in the mzXML format. LuciPHOr2 is capable of calculating a False Localization Rate (FLR) for several different kinds of PTMS [44]. It is an improved algorithm of LuciPHOr. In principal, it uses a modified target-decoy based approach as it is also used for the peptide identification to obtain a probability of false site localizations for a given modified peptide. Thus LuciPHOr2 utilizes the information which is present in the MS/MS spectra, including mass accuracy and peak intensities to calculate a model based probability and estimate the FLR for each peptide [62].

LuciPHOr2 was accessible through one of the clusters and could be executed via a bash line commands. LuciPHOr2 required as input the iprophet.pep.xml with the PeptideProphet scores and the according mzXML files of each run used for the prior executed peptide identification search. Further, Luciphor2 demanded a generated input text file. In the text file the path to the directory of the spectra and the pep.xml and their type was written. The MS2 tolerance was set to 0.04 Da and the fragmentation method was given as CID. The modeling threshold was set to 0.95 which was the minimum score for a peptide-spectrum match (PSM) needed to be considered for modeling in LuciPHOr2. The minimum number of PSMs for any charge state needed to be considered for modelling was set to 50. The maximum peptide length was set to 40 amino acids and the maximum charge state was restricted to +5.

As modifications phosphorylation on serine, threonine and tyrosine were allowed and the mass of the phosphorylation was given by +79.9666331 Da. LuciPHOr2 generated an output file for all peptides which fulfilled the restrictions and calculated a site localization score and estimated an FLR. With the help of a script the results of LuciPHOr2 were filtered by a global FLR threshold of 0.1 and the result was written back to the pep.xml (script by George A. Rosenberger). The global FLR was set to 0.1.

Generation of a phospho-SWATH assay library

To build a searchable spectral library for SWATH-MS data, we used an approach that was recently published [61]. We used SpectraST in the create mode to construct a spectral library from a peptide identification search and the according data. SpectraST is used for spectral library and searching, and was designed for DDA proteomics. It is part of the TPP and was developed at the Institute of System Biology in Seattle (ISB) [63]. All spectra with a peptide-probability higher than 0.9 were imported to the spectral library using the SpectraST tool. The proteins starting with "reverse_", were the decoy proteins/peptides. These were excluded as later decoy proteins/peptides were generated from the peptide sequences present in the spectral library. A file, which contained the retention times of the iRT-peptides (minus the LFLQFGAQGSPFLK iRT-peptide because it was not measured due to the short gradient), was used to recalibrate, accordingly to the theoretical values, the retention times of all measured peptides. The merged spectral library often contained redundant multiple spectra that were annotated to the same peptide precursor ion, as many different injections were performed. These redundant spectra from the same peptide precursor ion could differ in retention time and therefore the redundant spectral library was split according to a 2 minutes distance. Out of the split redundant spectral libraries a split consensus spectral library was generated for the different spectra annotated to the same peptide precursor ion. The consensus spectral library was generated from the split consensus spectral library using all available spectra for a certain peptide precursor.

From the consensus spectral library the best 6 transitions per peptide precursor were considered to generate an SWATH assay library. The mass to charge range for the precursor ions were constrained from 350 m/z to 2000 m/z. All precursor ions with less than 6 transitions were not considered for the final library. Duplicated masses for precursors were removed. The theoretical masses for all precursors were considered and an imported file provided the library with the information of the 64 SWATH-MS windows. The output file was saved as tab delimited file.

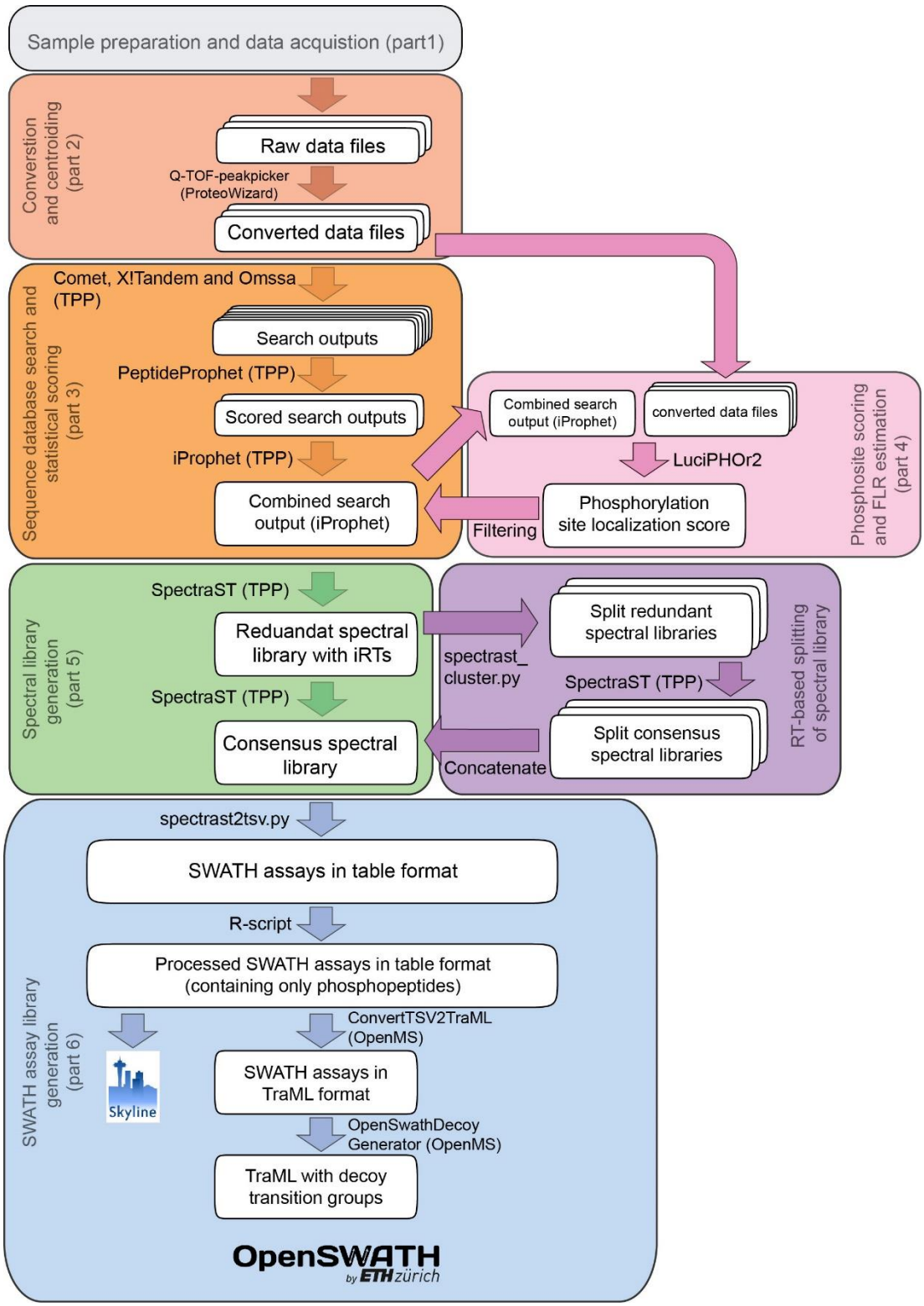


Figure 4: Workflow of the generation of a phospho-SWATH assay library with LuciPHOr2. *The flow chart graphic was adapted from the publication of Schubert et al. [61]. The workflow used had one additional step, the phosphorylation site localization scoring with LuciPHOr2 (part 4 in the illustration). After filtering the phosphopeptides with a low site localization score the workflow followed again the published one. Another variation was in part6 of the workflow, as non-phosphorylated peptides were filtered out of the SWATH assays in table format. Further the retention times of the iRT-peptides were changed to the theoretical at this step. This refined list was converted to a TraML format and out of the phosphopeptide sequences within the SWATH assay library decoys were generated.*

The SWATH assay library was further refined by loading the created tab delimited text file into R. The remained decoy peptides and all peptides without a phosphorylation site were removed from the list. The retention times of the iRT-peptides were changed to the theoretical values and the leading “subgroup” identifier which remained from the split library was removed so that there was only one assay for a distinct precursor.

By following again the protocol as described by Schubert et al. [61] the refinement tab delimited file was converted to a TraML library, the HUPO PSI mass spectrometry standard format for transition lists. To generate again decoy sequences the OpenSwathDecoyGenerator was used. The decoy generation was achieved by using the reversible sequence of the present phosphopeptide sequences in the library. The final library was named “Spec_Lib_cons_all_31DDA_docy” and uploaded to iPortal.

Construction of an IPF library for PTM detection and quantification using OpenSWATH

Inference of PeptidoForms (IPF) algorithm is an extension to OpenSWATH analysis software which is currently in development by George A. Rosenberger and available within iTestPortal platform. The newly developed algorithm aims to distinguish between closely related peptidoforms, with any further either automated, by tools like LuciPHOr2, or manual validation of the phosphorylation sites. The algorithm is especially developed for the detection and quantification of PTMs in DIA data such as SWATH-MS. In a few words, the algorithm generates and further tests hypothesis based on with peptide identification searches annotated spectral libraries. This unpublished workflow was used to construct two OpenSWATH/PTM libraries. For the filtered OpenSWATH/PTM library the filtered and refined tab delimited file of the above described phospho-SWATH assay library construction workflow, was used. This means, this library consisted only of already through LuciPHOr2 validated phosphorylation sites. Further all remaining peptides were removed from this spectral library. For the construction of the filtered OpenSWATH/PTM the filtered tab delimited file was converted to a MRM file. In the next step, this MRM file was converted to a TraML and the PTMs expected to be in the library were added. In our case this was phosphorylation on serine, threonine, and tyrosine and oxidation and the methionine. In the last step decoys were generated from the peptide sequences within the library, by using the shuffle method. For the second OpenSWATH/PTM assay library, the pep.xml of the unfiltered peptide identification search and the mzXMLs were used to generate a spectral library (without Luciphor2). Out of this spectral library the MRM file was generate. All further steps were conducted as for the filtered OpenSWATH/PTM library.

filtered OpenSWATH/PTM library

"Fabianf_DB_assays_ptms_LuciPHOr_cons_31DDA-tripleTOF_decoys" is the name of the openSWATH/PTM assay library with the filtered input.

unfiltered OpenSWATH/PTM library

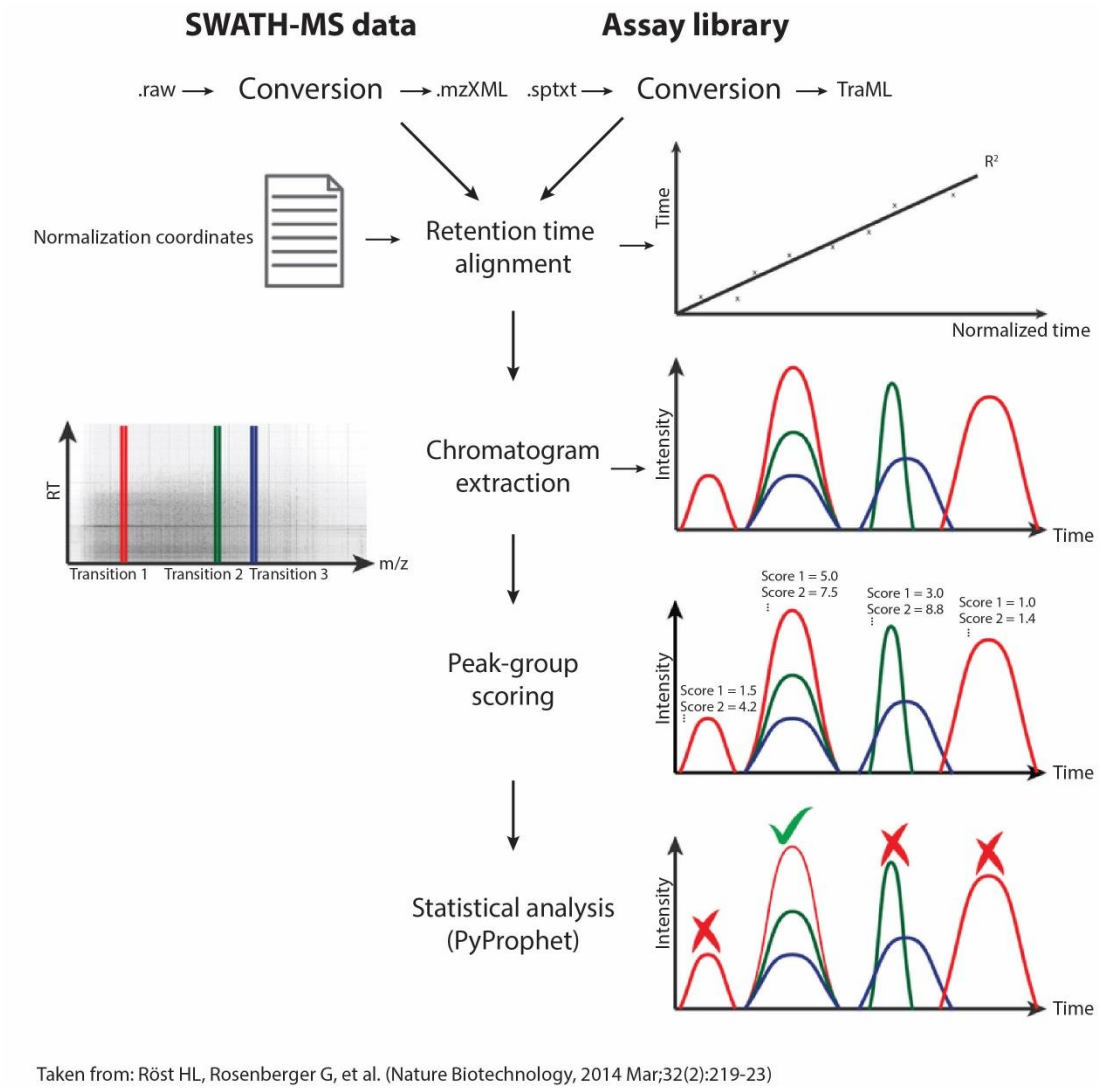
"20160222_fabianf_DB_assays_ptms_cons_31DDA-tripleTOF_decoys.TraML" is the name of the openSWATH/PTM assay without any filtering.

2.2.3. SWATH-MS analysis workflow

The analysis workflow was the same for all SWATH assay libraries. All phosphopeptide enriched samples and the whole tissue lysate of mouse liver was analyzed with the OpenSWATH pipeline implemented in iPortal. The phosphopeptide enriched samples were analyzed using the constructed phospho-SWATH assay library. The exact construction of the phospho-SWATH assay library is mentioned above. In addition to this library two OpenSWATH/PTM spectral libraries following the IPF-workflow (George A. Rosenberger, unpublished) were used to analysis the aging dataset. The three PTM libraries were used for benchmarking the result of the phosphopeptide enriched samples of the "aging" experiment. For the analysis of the whole cell lysate samples of the "aging" experiment, an mouse liver tissue library of Yibo Wu was used [19]. The acquired multiplexed spectra of the phosphopeptide enriched samples of the "BXD-mouse reference population" experiment were analyzed using the phospho-SWATH assay library.

OpenSWATH for SWATH-MS and phospho-SWATH-MS

The whole tissue lysate samples and the phosphopeptide enriched samples of the "aging" experiment, and the phosphopeptide enriched samples of the "BXD-mouse reference population" experiment were measured on the 5600+ TripleToF from Sciex in DIA mode and analyzed by OpenSWATH-workflow [36].



Taken from: Röst HL, Rosenberger G, et al. (Nature Biotechnology, 2014 Mar;32(2):219-23)

Figure 5: Steps performed by the OpenSWATH software during SWATH data analysis. *The graphic illustrates a peptide precursor with three fragment ion transitions, which are shown in red, green and blue. The graph on the left side shows the three transitions in a mass to charge ratio versus retention time graph. Over the whole retention time a specific mass is extracted. This is a so called transition. The four graphs on the right side illustrate the OpenSWATH analysis step. On the upper part the inputs and the formats of the input files are listed, which are necessary for the OpenSWATH pipeline. For the analysis, the raw data were converted to the mzXML data format. The OpenSWATH analysis required a SWATH assay library, which consisted of phosphopeptide assays. This library was generated in advance to the analysis and converted to a TraML. In the first step the acquired MS data were aligned according to the retention times of the iRT-peptides. In the second step, the transitions for the fragment ions were extracted by making use of the assay library. Afterwards the extracted peak-groups were scored by several algorithms. The scores considered among other attributes how well the transitions aligned in terms of retention time, the peak shapes and peak intensities. The PyProphet algorithm weighted the different scores and calculated a single score for each assay. As the library also consisted of decoy peptide sequences further filtering by an FDR was conducted by the OpenSWATH pipeline.*

The m/z extraction window was set to 0.05 Thomson in a retention time window length of 600 s. The retention time window was therefore +/- 300 seconds around the expected retention time of the spectral library. The minimum accepted R^2 for regression of the iRT-peptides was set to 0.95 Outlier detection was applied afterwards to remove wrongly assigned reference peptides. The minimum accepted relative number of iRT-peptides after outlier detection was set to 60%. The DIA score was enabled to score a single chromatographic feature using DIA / SWATH scores.

For the chromatogram extraction scoring PyProphet [64], an improved version of the mProphet [65] algorithm, was used. PyProphet considered extrinsic scores (e.g. Retention time deviation of the peakgroups compared to the expected retention time of the spiked-in iRT-peptides. Intensity correlation with the intensity of the peakgroup in the spectral library) and intrinsic scores (e.g. Co-elution score for precursor and transitions, peak shape score for precursor and fragments, intensity of the precursor and transitions). PyProphet used for building the model for m-score and d-score. The d-score cutoff prevented the writing of peakgroups with a d-score below the 1 cutoff. All peakgroups that scored below the cutoff were considered as bad peakgroups and were not considered in the subsequent feature alignment. The exact parameters for the default quantification were used as described in [64]

In detail PyProphet was run in 10 cross validation runs on the OpenSWATH workflow with an adjusted output, which contained as scores (xx_swath_prelim_score, bseries_score, intensity_score, isotope_correlation_score, isotope_overlap_score, library_corr, library_rmsd, log_sn_score, massdev_score, massdev_score_weighted, norm_rt_score, xcorr_coelution, xcorr_coelution_weighted, xcorr_shape, xcorr_shape_weighted, yseries_score).

The SWATH-MS data set from each sample were aligned using the TRIC algorithm (Röst et al, in revision). The feature alignment was enabled and the Spline interpolation method was used. Spline interpolation is a commonly used regression method using stepwise polynomial functions to build the model. The accuracy of the model was improved using generalized cross-validation. The peaks among the runs were grouped by using the global best overall peak detected clustering method. The method first started with a good peak in one spectra and searched for the same peak in other runs within a maximal three times median standard deviation in retention time seconds window. The peak with the best PyProphet score was selected as reference and all peaks from the other runs are retention time aligned according to this peak. The separation between true and false signal was achieved using a decoy-target based approach. The decoy assays were scored exactly the same was the target assays. The target FDR cutoff was set to 0.01.

Re-quantification was enabled for all LFQ analysis. The “aging” dataset was once analyzed with the same parameters, but with disabled re-quantification.

OpenSWATH/PTM analysis

The phosphopeptide enriched samples of the “aging” SWATH-MS acquired spectra were analyzed with the generate OpenSWATH/PTM libraries. Recommended parameter settings were used and are listed in the appendix. The quantification

considered MS1 and MS2 features. The iTestPortal settings were saved as a separate method named “PHOSPHO-SWATH_G” and are listed in the appendix.

SWATH2stats R-package for reshaping the OpenSWATH output

The feature alignment output text file of the OpenSWATH analysis, conducted via the iPortal or iTestPortal, were loaded to the recently published SWAH2stats R-package [66]. With the package the data could easily be annotated and filtered by distinct criteria depending on the analyzed data. The package allowed further visualization of the FDR thresholds and to transform the data to a format which was suitable for the subsequent mapDIA analysis.

The whole tissue lysate of the “aging” dataset was analyzed with OpenSWATH and subsequent filtered with the SWATH2stats R package. As filter criteria the peptides had to be detected in least at two replicates out of the three replicates per biological sample. The m-score cutoff was set to 0.01, leading to a Peptide FDR of 0.012. The phosphopeptide enriched samples of the “aging” dataset were analyzed with the phospho-SWATH assay library with enabled re-quantification within OpenSWATH. By filtering an overall Peptide FDR of 0.01 was achieved. As filter criteria a phosphopeptide had to be quantified in at least two out of three replicates and an m-score cutoff of 0.01 was used. The OpenSWATH analysis of the “aging” dataset without re-quantification and the two OpenSWATH/PTM library results were filtered in the same way. Phosphopeptides had to be detected in at least two replicates and the estimated m-score value was chosen in that way, that the resulting overall Peptide FDR was below 0.01.

The OpenSWATH results of the phosphopeptide enriched samples of the “BXD mouse reference population” experiment had slightly different filter criteria compared to the above mentioned “aging” datasets. All phosphopeptides which had not been detected in at least 60 % of the 76 samples were filtered out. The estimated m-score cutoff was set to 0.01 to achieve an overall Peptide FDR below 0.01.

MapDIA analysis

The software package mapDIA (Model-based Analysis of Quantitative Mass Spectrometry Data in Data Independent Acquisition Mode) was developed for processing and statistical analysis of quantitative proteomics data from DIA mass spectrometry [67]. The package is suited for protein level quantification and allows normalization and automated protein level based statistical analysis. The input for the mapDIA analysis was generated with the SWATH2stats R package.

For the “aging” dataset, regardless which SWATH assay library was used, the same input file parameters for mapDIA were used, including the total cell lysate. The

experimental design for the “aging” dataset was specified as ReplicateDesign which allowed the comparison of different conditions within each biological sample over at least two biological replicates. The ReplicateDesign corresponds to a paired statistical design. For normalization TIS was selected which stands for division by the total ion chromatogram. Further the standard deviation factor (SDF) was set to 2 for filtering out the outliers which were lying out of median +/- 2 standard deviations in each biological replicate. The median intra-protein correlation cutoff was set to 0.1 and the minimum numbers of observations per sample was set to 2. As minimum number of observed fragments per peptide 3 was set and as maximal 5. The minimum number of observed peptides per protein was set to 1. The “BXD-mouse reference population” dataset was analyzed in mapDIA using the IndependentDesign. In fact all the other values were used as for the “aging” dataset only that the size per sample was as 1. MapDIA created several output files. For the analysis and statistical data the “peptide_level.txt” or “protein_level.txt” was loaded into R.

2.2.4. STRING PPI-network construction

The Search Tool for the Retrieval of Interacting Genes (STRING) database was used for the functional interaction analysis of up- or downregulated phosphoproteins or proteins lists. STRING is an online search tool, which calculates a functional interaction network of proteins, globally integrates them, and finally scores the network. For the functional association between proteins, STRING uses information from various sources for combining and scoring of the interactions. The results are represented in a Protein-Protein-Interaction network (PPI-network). The interactions are derived from multiple databases: (i) known experimental interactions are imported from primary databases, (ii) pathway knowledge is parsed from manually curated databases, (iii) automated text-mining is applied to uncover statistical and/or semantic links between proteins, based on Medline abstracts and full-text articles, (iv) interactions are predicted de novo by a number of algorithms using genomic information as well as by co-expression analysis and (v) interactions that are observed in one organism are systematically transferred to other organisms, via pre-computed orthology relations. For the generation of the networks presented in the thesis, the actual version of STRING v10 was used, which includes an updated pipeline for inferring protein-protein associations from co-expression [68]. The sources of interaction from each of this level are than calibrated against the high-level functional groupings by the manually curated Kyoto Encyclopedia of Genes and Genomes (KEGG) pathway maps [69].

Further an R computing interface is implemented which allows the statistical analysis for enrichment of molecular, or functional pathways [68]. For this enrichment tests the Kyoto Encyclopedia of Genes and Genomes (KEGG) and Gene Ontology (GO) databases are used. We performed the enrichment tests via the STRING database against the union of all detected phosphoproteins and proteins of the "aging" experiment.

2.2.5. Mapping of phospho-pQTLs

To discover new phospho-pQTLs, we used SWATH-MS analyzed data of the BXD mouse genetic reference population. For a subset of phosphoproteins, for which the changes in abundances, were likely to origin from genetics, we performed a QTL mapping. This subset consisted of all phosphoproteins, which had a Spearman correlation coefficient of 0.5 or higher, between the quantified intensities of the two diets, high fat diet (HFD) and chow diet (CD), of the same BXD mouse strain, over all BXD strains. The QTL mapping was performed separately for each diet for all BXD mouse strains with an R script of Evan Williams, which uses the R package R/qtl, [70]. The genome of both parental strains has been sequenced, and severs as reference genome [71], [72]. The BXD genotype of each strain were genotyped in 2005 with 13'377 markers. The genotypic information gained from this makers were combined with previously used markers, which lead in total to 7636 informative makers. This makers differ between the two parental strains, and they turned out to be useful for mapping of QTLs in the BXD mouse reference population. The BXD genotype file used for QTL mapping in the current study, included a selected subset of approximately 3795 markers (of total 7636), which included all those markers with unique strain distribution patterns or marker genotype string. Further it contained the most proximal and most distal makers for each strain distribution pattern represented by two or more makers [13]. For mapping the QTLs, we used the Haley-Knott regression with the non-parametric model assumption [73]. For this model, an extension of the Kruskal-Wallis test was used, which is similar to the method described by Kruglyak and Lander [74]. We used the non-parametric method for the quantitative phosphopeptide data, as it can be that they are normal and non-normal distributed for the different phenotypes. For the significance threshold calculation 1000 permutations were applied to the data, and only the makers with a p value higher than 0.95 were accepted. The resulting data with the phenotype name were written to the final result.

The QTL analysis provided us with tow output lists, one for each diet inputs, which were further analyzed to identify cis- and trans-QTLs. All mapped traits, which were located within a 10 mega base pairs (Mb) range on the same chromosome as the

gene of the quantified phosphoprotein, and had a p-value below the 0.67 cutoff, were considered as cis-pQTLs. Whereas traits identified on the same chromosome farer than 10 Mb, or on a different chromosome, which were below a p-value of 0.05, were categorized as trans-pQTLs.

2.2.6. R-scripts for statistical data analysis

Beads and buffer combination

As an input for this script, we used peptide identification search results obtained via the iPortal platform or MaxQuant. The two datasets were reshaped, annotated, and analyzed in terms of identified unique phosphopeptides. The data were visualized with plots and some characteristics like the phosphopeptide enrichment factor were calculated. The results were visualized by using the ggplot2 R package.

Amount of starting material and beads ratio

The input for this script were peptide identification search results generated with MaxQuant or the iPortal platform. The generic functions form the beads and buffer combination analysis script were used to reshape and annotate the data. The script contain various visualizations with the R package ggplot2.

Refine phospho-SWATH consensus assay library

With the R-script decoys and all peptides which were not phosphopeptides were filtered out. The theoretical retention times of all iRT-peptides was added. If there was only one assay for a protein the leading “subgroup_” was removed from the identifier.

Transform and Filter SWATH Data for other Statistical Packages

The SWATH2stat package was used to filter and transform data of the OpenSWATH software into a format readable by mapDIA as described in the method chapter 2.2.3 SWATH-MS analysis workflow (Page 30).

LFQ, SWATH comparison of phosphopeptide enriched samples

The R script was used, to analyze the performance of two LFQ tools and the SWATH-MS on the phosphopeptide enriched samples of the “aging” experiment. DDA data were quantified via MaxQuant and the iPortal platform. The MaxQuant and the OpenMS output were used as input. As further input, DIA measurements analyzed with OpenSWATH and processed with SWATH2stats and mapDIA was used. The script contained code to annotate and reshape the data. Also generic functions previously written were used to calculate the statistics. For example the intensities of the phosphopeptides obtained by the various quantification tools were correlated and the CVs over all samples and within the replicates were calculated. Further the mapDIA output of the total proteome and the phosphopeptide enriched samples were used to identify due to age differently regulated proteins and phosphoproteins. As

main result, lists of regulated proteins and phosphoproteins were obtained. The script also provides code for various graphical representation of the data.

Comparison the phospho-SWATH assay library to the OpenSWATH/PTM libraries

The script compared the performance of the three SWATH assay libraries. Thus as test data, the phosphopeptide enriched samples of the “aging” experiment, were quantified with all three SWATH assay libraries. After data processing via OpenMS, SWATH2stats and mapDIA the data were loaded to the R-script and further analyzed. Generic code previously used for data analysis was reused. For example the script correlated the intensities of the phosphopeptides and calculates the CV within the replicates and among all samples.

Analysis of the BXD mouse genetic reference population

Statistical analysis of the mapDIA result of the SWATH-MS analysis of the BXD mouse reference population, was performed partly by reusing code from previously used scripts to analyze mapDIA output. In addition, with the script analysis of the due to diet or genotype regulated phosphoproteins were identified. The resulting lists were used as input for the phospho-pQTL mapping.

Mapping of phospho-pQTL

A script written from Evan Williams was used to discover phospho-pQTLs in a subset of proteins of the inbred BXD mouse genetic reference population. The script uses the quantitative data and the BXD mouse sample identifiers as an input. The BXD genotype file of the GeneNetwork, which includes a selected subset of approximately 3795 markers (out of 7636) and all those markers with unique strain distribution patterns were written to the identifier and quantitative data mapping of QTLs.

3. Materials

3.1. Experimental part

Common chemicals used for more than one experiment

Solvents:

HPLC grade H_2O (7732-15-5, Fisher Chemical)
Acetonitrile (A955-212, Fisher Chemical)
Trifluoroacetic acid (TFA) (85183, Thermo Fisher)

Chemicals:

Urea (GEPURE0067 Eurbio)
Ammonium bicarbonate (ABC) (09830 Fluka)
Tris (A3452, Tris hydrochloride Applichem)
Sodium chloride (1.06404 Merck)
tris(2-carboxyethyl)phosphine (TCEP) (20491, Pierce™)
Iodoacetamide (IAA) (I1149 Sigma)
cOmplete™, EDTA-free Protease Inhibitor Cocktail (000000011873580001, Roche)

Enzymes:

Lys-C or Lysyl Endopeptidase®, Mass Spectrometry Grade (125-0561 Wako)
Trypsin: Sequencing Grade Modified Trypsin (frozen) (V5113 Promega)

Thermoshaker: Thermomixer Compact (Eppendorf) and Thermomixer comfort (Eppendorf)

Mouse liver tissue samples

The mouse liver tissue samples were prepared in the Laboratory of Integrative Systems Physiology (LISP) at the École Polytechnique Fédérale de Lausanne (EPFL) from Evan Williams. In total 81 samples were sent to the ETH. The samples were always frozen and stored at -80 °C. The 76 samples of the BXD mouse reference population and the four aging mouse samples weighted around 50 – 200 mg. A single C57BL/6 sample approximately consisted of 1000 mg and was used for testing and optimization of lysis and enrichment protocols. For all experiments the mouse liver tissue was cut with a scalpel into 50 mg pieces. The sample amounts, concentrations and the lysis efficiency are listed in the appendix. The mouse liver tissue pieces were weighted on a Mettler Toledo Excellence XS205 DualRange.

Scalpel handle, no.3, without blade (Swann-Morton, Sheffield England)
Scalpel blade, stainless steel, no. 10 (Swann-Morton, Sheffield England)
Stainless steel sterile needle 22G x 1 ¼ - Nr. 12 (300900,BD Micorlance™ 3)

Conventional lysis

Glass dounce homogenizer (Kontes Glass Co., Vieland N.J., 7 mL) with a tight pestle A.

RIPA-M buffer:

1 % IGEPAL® CA-630 (I8896 SIGMA)
0.1 % Sodium deoxycholate (30970, Sigma)
150 mM NaCl
1 mM EDTA (EDTA disodium salt dihydrate, A2937, Applichem)
50 mM Tris pH 7.5 (pH was set with NaOH)
HPLC grade H_2O

Urea-T buffer:

50 mM Tris pH 8.1 (pH was set with NaOH)
75 mM NaCl
8 M Urea
HPLC grade H_2O

Protease and Phosphatase inhibitors for the RIPA-M buffer and Urea-T buffer:

cOmplete™, EDTA-free Protease Inhibitor Cocktail (stock of 50x)
10 mM NaF (S7920 Sigma-Aldrich) (stock of 500 mM)
10 mM Sodium pyrophosphate (71501, Fluka) (stock of 500 mM)
5 mM 2-Glycerophosphate (β-Glycerophosphate disodium salt hydrate, G9422, Sigma) (stock of 500 mM)
HPLC grade H_2O

PCT lysis

Barocyclers: Model: NEP232 and NEP2320 Enhanced.
Lysis buffer: 8 M Urea in 0.1 ABC plus same protease and phosphatase inhibitors with the same concentration as for the conventional lysis.
0.1 M ABC in HPLC grade H_2O
200 mM TCEP stock in lysis buffer; final concentration 10 mM
400 mM IAA stock in lysis buffer; final concentration 40 mM
IAA and TCEP are first mixed together and added to the lysis buffer containing the lysed tissue.
Sequencing Grade Modified Trypsin

Lys-C or Lysyl Endopeptidase®

10 % TFA in HPLC grade water

BCA assays

The BCA assays were conducted with Pierce™ BCA Protein Assay Kit from Thermo Fisher (23225, Thermo Fisher). For the standard curve Bovine Serum Albumin (BSA) was used in the final concentration range of 8 mg mL^{-1} to 0.125 mg mL^{-1} (23209, Albumin Standard, Thermo Scientific). Absorption at 562 nm on a Multi-Detection Microplate Reader (Synergy HT, BioTek®) was measured at 25 °C.

Acetone precipitation

-20 °C cold Acetone is added (32201 Sigma-Aldrich).

Reduction and Alkylation

0.1 M ABC stock solution

8 M Urea in 0.1 M ABC in HPLC grade H_2O

50 mM TCEP in HPLC grade H_2O stock solution

400 mM IAA in 0.1 M ABC stock solution

Enzymatic treatment

Sequencing Grade Modified Trypsin

Lys-C or Lysyl Endopeptidase®

0.1 M ABC in HPLC grade H_2O

C18 purification steps

Acidification with 5 % TFA (Trifluoroacetic acid, 85183, Thermo Fisher) in H_2O

Equilibration buffer, 0.1% TFA in 2% acetonitrile and H_2O

Washing buffers: Methanol (67-56-1, Fisher Chemical)

0.1 % TFA in 80 % acetonitrile and H_2O

Elution buffer: 0.1% TFA in 50% ACN and H_2O

Depending on the amount of peptides or respectively phosphopeptides, different C18 silica reverse-phase chromatography columns were used. For the cartridges the binding capacity varies and therefore also the maximal amount of peptides respectively phosphopeptides which can be efficiently purified vary.

Waters Sep-Pak® Sample Extraction Products, Sep-Pak® Vac C18 3cc (500 mg): 1 – 5 mg of peptides, > 2 mL elution volume

The Nest Group MicroSpin Columns: Ultra Micro Spin Column (2-100 µl loading, 5-60 µg capacity #SUM SS18V)

The Nest Group MicroSpin Columns: Ultra Micro Spin Column (2-100 µl loading, 3-30 µg capacity #SUM SS18V)

Phosphopeptide enrichment

Loading buffers

6 % TFA in 80 % acetonitrile
1 M glycolic acid (124737, Sigma-Aldrich) in 80 % acetonitrile and 5 % TFA
30 % Lactic acid solution (Lactic acid solution ≥ 85 %) in 0.9 % TFA and 70 % acetonitrile
saturated phthalic acid (600 mg of 402915, Sigma-Aldrich) in 2.5 % TFA and 83 % acetonitrile

Washing buffers and elution buffer for MagReSyn® Ti-IMAC beads

Washing buffer: 80% ACN, 1% TFA
Elution buffer: 1 % NH_4OH

Washing buffers and elution buffer for TiO_2 and Ti-IMAC beads

Methanol (only TiO_2)
Washing buffer A: 80 % acetonitrile, 0.1 % TFA
Washing buffer B: 50 % acetonitrile, 0.1 % TFA
Washing buffer C: 0.1 % TFA
Elution buffer: 0.3 M ammonium hydroxide buffer: 5 mL HPLC grade H_20 plus 100 µl from ammonium hydroxide stock solution (22/228, Sigma) (Ensure that the pH is 10.5-11. It may be required to add more ammonium hydroxide)
pH Adjustment: 15 % TFA

Titansphere TiO 5 µm, TiO2 (5020-75000, GL Sciences Inc. Japan)
Magnetic MagReSyn® Ti-IMAC immobilized metal affinity chromatography, as a 20 mg mL-1 suspension in 20 % ethanol, stored at 2 – 8 °C until usage. (MR-TIM002, MagReSyn®, Biosciences)
IMAC beads (China, Cooperation partner Key Lab of Separation Sciences for Analytical Chemistry, National Chromatographic R & A Center, Dalian Institute of Chemical Physics, Chinese Academy of Sciences, Dalian, China.)

IMAC beads activation

Titanium(IV) chloride solution, 0.09 M in 20 % HCl (404985 Sigma-Aldrich)
0.1 % TFA in 30 % acetonitrile

MS sample buffer

0.1 % Formic Acid (33014, Sigma Aldrich) in 2 % acetonitrile iRT-peptides 1:20 (v/v) to the sample (Ki-3002, Biognosys)

Mass Spectrometry

LTQ XL Linear Ion Trap Mass Spectrometer (Thermo Scientific™) interfaced with a Thermo EASY-nLC II system (Thermo Scientific™)

Orbitrap Elite Hybrid Ion Trap-Orbitrap Mass Spectrometer (Thermo Scientific™) interfaced with a ultrahigh pressure HPLC system Thermo Easy-nLC 1000 (Thermo Scientific™)

5600+ TripleToF Mass Spectrometer (AB Sciex) coupled to a an Eksigent 2Dplus Nano LC systems (Eksigent)

Analytical column for Oribtrap Elite™: PepMap with 75 µm inner diameter x 150 mm length, 3 µm particle size. Maximum loading approximately 5 µg, gradient lengths up to 4 hours (Thermo Scientific™)

Analytical column (75 µm inner diameter × 20 cm length) was home-packed directly in a fused silica PicoTip emitter (New Objective, USA) with ProntoSIL, 200 Å pore size, 3 µm particle size, C18 AQ resin (H184PS030, ProntoSIL, Bischoff).

Glufib, Glu1-Fibrinopeptide B (F-3261, Sigma)
β-galactosidase digest (4333606, AB beta-Galactosidase digested)
Buffer A: 2% acetonitrile and 0.1% formic acid in HPLC H2O
Buffer B: 98 % acetonitrile and 0.1 % formic acid in HPLC H2O

3.2. Software used for bioinformatics and biostatistics

iPortal platform

ProteoWizard converter (3.0.5533) for all measurements from December 2015 on (all SWATH-MS)
ProteoWizard converter (3.0.7494) for all measurements before December 2015
openBIS (open Biology Information System) (Version 13.04.x (r35657))
iPortal (version 3.5.7)
Within iPortal: Comet (version "2015.02 rev. 3"), Omssa (omssacl:

2.1.9), X!Tandem (X! TANDEM Jackhammer TPP (2013.06.15.1 - LabKey, Insilicos, ISB)),
MaxQuant (1.5.2.8)
murine UniProtKB/ SwissProt protein database enriched with contaminants and decoy peptides (reverse protein sequences of the database) are automated monthly updated. Were used from the (01.07.2015 until 01.03.2016)
murine UniProtKB/SwissProt protein database (03.09.2015 for MaxQuant analysis)

Label Free Quantification

OpenMS default settings via iPortal ("DEFAULT – READ ONLY") for more information of the ongoing OpenMS developments: http://open-ms.sourceforge.net/
The default settings of the analysis were as described in [57]

LuciPHOr2 PTM site scoring

The tool LuciPHOr2 was used to calculate a site localization score of the phosphopeptides identified via the peptide identification search in iPortal.
LuciPHOr2 JAVA-based version of LuciPHOr2 (Version: 1.2014Oct10)

IPF: PTM Detection and Quantification using OpenSWATH

Inference of PeptidoForms (IPF) is an extension to OpenSWATH currently in development and available within iTestPortal environment. The unpublished version is currently under development by George A. Rosenberger. The analysis was done with the version of the 16.03.2016.

mapDIA

Model-based Analysis of Quantitative Mass Spectrometry Data in Data Independent Acquisition Mode (mapDIA_ 2.2.1)

Mapping of phospho pQTL

R-scrip based mapping with the R/qtl-package

SRING v10: protein-protein interaction networks

Function protein association networks (Version: 10.0 [68]). Online toll which can be reached via string-db.org (used at the 20.04.2016)

R and R-studio

R (versions 3.2.3 and 3.2.4)
R studio (Version 0.99.892)

4. Results

4.1. Optimization of the phosphopeptide enrichment for mouse liver tissue

Acetone precipitation

The sample quality in MS analysis of phosphopeptides enriched samples can be influenced by incomplete removal of phospholipids. Thus, we performed the overnight acetone precipitation at – 20 °C, to purify the proteins prior to the digestion and enrichment. Phospholipids might also lead to issues and unspecific binding during the phosphopeptide enrichment. The acetone precipitation was assumed to be critical for the purity of the samples. Thus, it is highly recommend to further put efforts in investigating and verifying these assumptions as it would help to increase the quality of the phosphopeptide enriched samples.

4.1.1. Beads and buffer combinations

For the specific enrichment of phosphopeptides a variety of enrichment strategies were developed. Within our laboratory several enrichment protocols were available and it was not clear, which one performs best for mouse liver tissue. Thus, several enrichment protocols and two lysis methods were tested in respect to screen for the best performing combination of beads, buffers, and lysis method, for mouse liver tissue. The two lysis methods, conventional lysis (CON) with a glass dounce homogenizer and pressure cycling technology (PCT), which was facilitated with a Barocycler, were compared to each other. For the selection of the beads and the enrichment procedures constrains were set. One of the requirements was, that the enrichment should be feasible without any fractionation. These constrains were considered, as we planned to process a large number of samples, and the measurement time on the high-end MS devices was limited.

The enrichment protocols, and enrichment parameters used for the experiment are listed in Table 1 in the methods chapter 2.1.2 (Page 16). For the peptide identification via the iPortal platform three different search engine combinations were used. The peptide identification searches were conducted with i) Comet alone (Comet), ii) with Omssa and X!Tandem (OmXT) and iii) with all three search engines together (CoOmXT). Another peptide identification search was performed with MaxQuant.

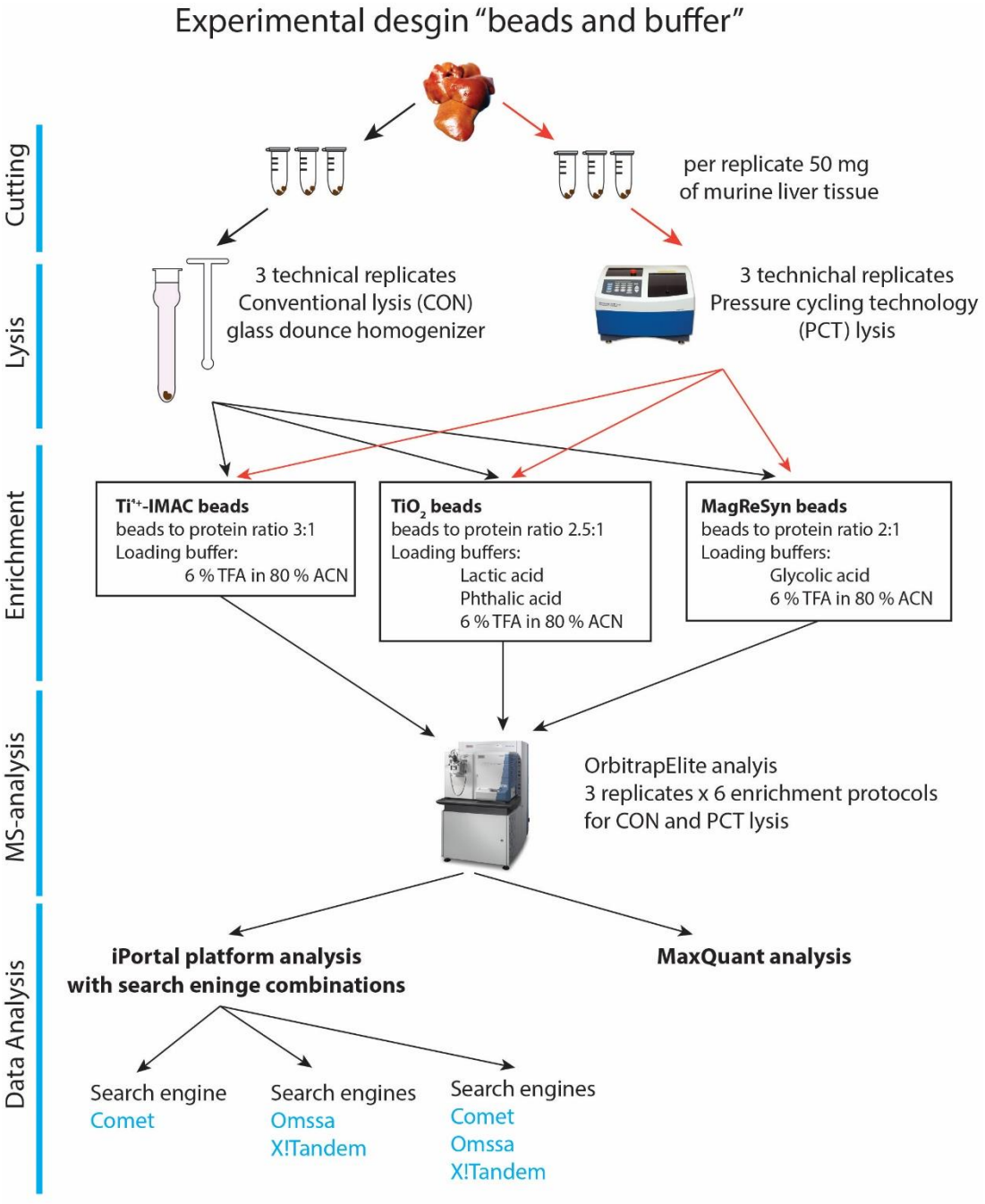


Figure 6: Experimental Design of the beads and buffer combinations experiment with a simplified workflow. *The workflow of the experimental design shows the two different lysis methods (CON and PCT), and the 6 beads and buffer combinations, tested. For the bioinformatics part, 3 combinations of search engines via the iPortal platform and one setting of the MaxQuant analysis were compared. The purification and enzymatic treatment steps are not shown in the graph.*

All samples of the experiment were measured on the Orbitrap-LTQ to test whether the phosphopeptide enrichment was successful or not. All samples that did not show any signal on the Orbitrap-LTQ were not measured on the high-end devices. The phosphoenrichment of MagReSyn® with 6 % TFA in 80 % acetonitrile caused blocking of the LC column on the OrbitrapElite and thus, only the first replicate was analyzed. It was assumed that the harsh conditions of the loading buffer lead to the leakage of the metal nanoparticles associated with the beads.

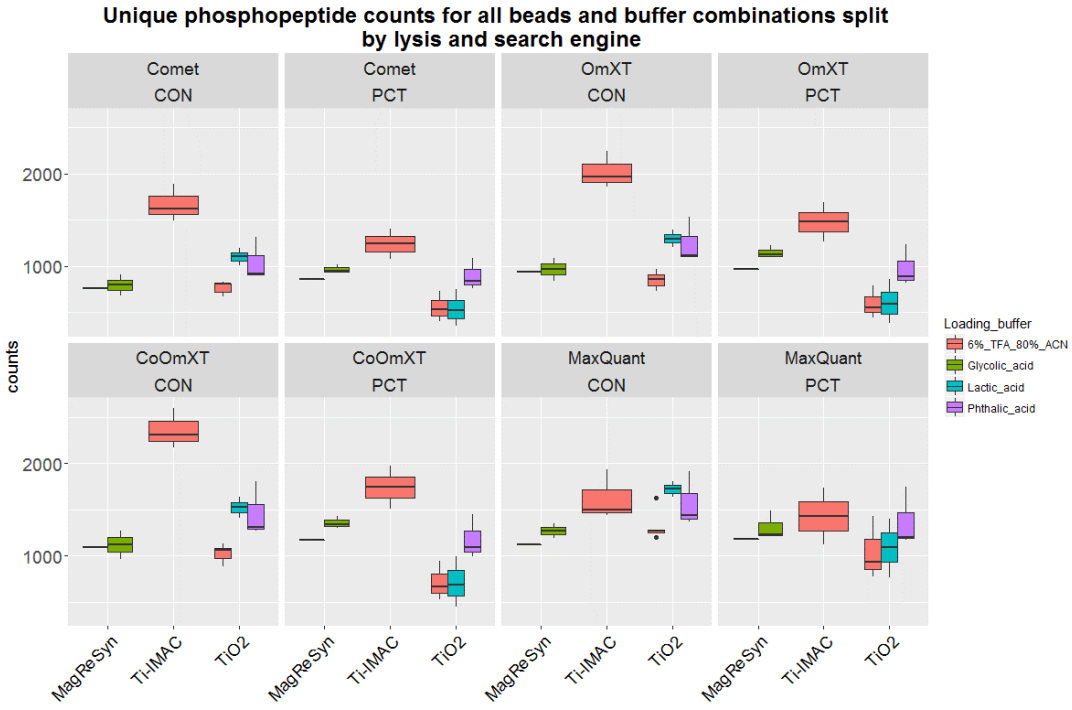


Figure 7: The unique identified phosphopeptides of the "beads and buffer" combinations experiment plotted as Box-Whisker plots. *Each section shows the result for one search engine split by the lysis method (CON for conventional lysis and PCT for pressure cycling technology lysis). For peptide identification the iPortal platform was used with different settings for the search engines. In the upper row, on the left-hand side, the results for the iPortal platform with Comet alone and on the right-hand side the results with the search engines Omssa and X!Tandem are shown. In the lower row, on the left-hand side, the combination of all three search engines is shown. In the lower row, on the right-hand side, the result for MaxQuant is shown. On the x-axis the beads used for the respective experiment are listed with the color of the boxplots corresponding to the loading buffer used for each experiment.*

To investigate, which phosphopeptide enrichment setup and downstream analysis setting performed best, we determined the highest number of unique identified phosphopeptides, for all combinations (Figure 7). We found, that the best beads and buffer combination was Ti^{4+}-IMAC beads with 6 % TFA in 80 % acetonitrile with the CON lysis, analyzed with the enabled search engine combination Comet, X!Tandem and Omssa. The average number of phosphopeptides for this combination was 2361 +/- 218. Further, the CON lysis showed for all combinations a higher number of unique identified phosphopeptides. An exception was if the data analysis was conducted with MaxQuant. For this setting, the differences between the various beads and buffer combinations were not informative, as they all performed rather comparable.

Next we investigated the specificity of the various phosphopeptide enrichment conditions. Therefore we calculated for all conditions the enrichment factor, which is defined as the ratio between all detected peptides and the detected phosphopeptides (Figure 8). The results of the enrichment factor substantiated the better performance of the Ti^{4+}-IMAC beads. The average enrichment factor for the Ti^{4+}-IMAC beads analyzed with iPortal and the enabled three search engines was 87 % +/- 2 %. For

the other beads and buffer conditions the specificity was below 50 %, except for a view conditions, for which it was around 50 %.

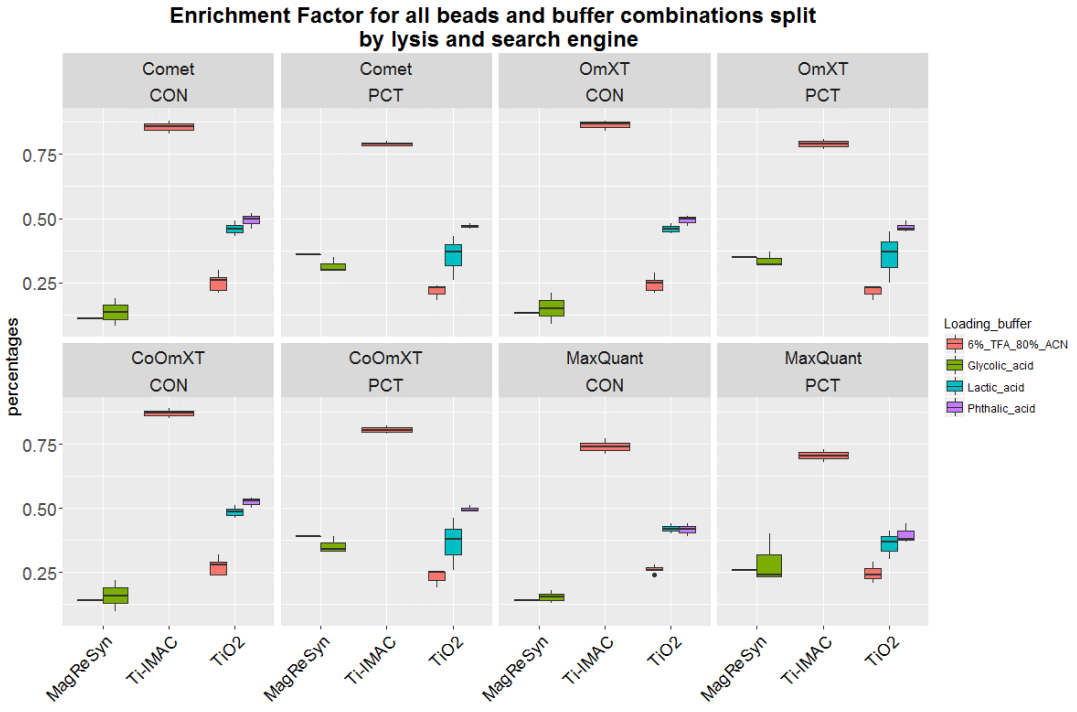


Figure 8: An overview of the enrichment factor of the phosphopeptide enrichment experiments. *A value for the specificity of a phosphopeptide enrichment experiment is the enrichment factor. The Ti^{4+}-IMAC beads with 6 % TFA in 80 % acetonitrile loading buffer showed highest enrichment factor of all beads and buffer combination and lysis methods. The enrichment factor for this combination was similar for all used search engines.*

In summary, the results of the beads and buffer combinations showed, that we were able to gain the highest number of phosphopeptides with the Ti^{4+}-IMAC beads with 6 % TFA in 80 % acetonitrile as loading buffer. Further we assumed, that an enrichment parameter setting optimization, increases the number of phosphopeptides per single injection. Thus, we performed an experiment for the phosphopeptide enrichment protocol optimization.

4.1.2. Parameter optimization of the Ti^{4+}-IMAC phosphopeptide enrichment protocol

Previously no one in our laboratory used the Ti^{4+}-IMAC beads with 6 % TFA in 80 % acetonitrile as loading buffer for phosphopeptide enrichment of mouse liver tissue. As the literature highly recommends pre-experiments to decide the optimum beads ratio when it comes to different samples, we decided to investigate the optimal conditions for the most critical enrichment parameters for mouse liver tissue: i) beads to starting material ratio, ii) the amount of starting material, and iii) the loading buffer volume [75]. The beads to peptide ratio was altered in the range from 3:1 up to 20:1. For testing of the amount of starting material the beads ratio was kept constant and the protein amount was increased from 0.5 mg up to 4 mg per sample. The loading buffer volume

was altered, by keeping the other two parameters constant, in order to control whether or not, the concentration of the phosphopeptides influenced the binding dynamics. As a control we used HeLa (HeLa H2B) total cell lysate. The experiment, including the lysis step, was carried out as a duplicate.

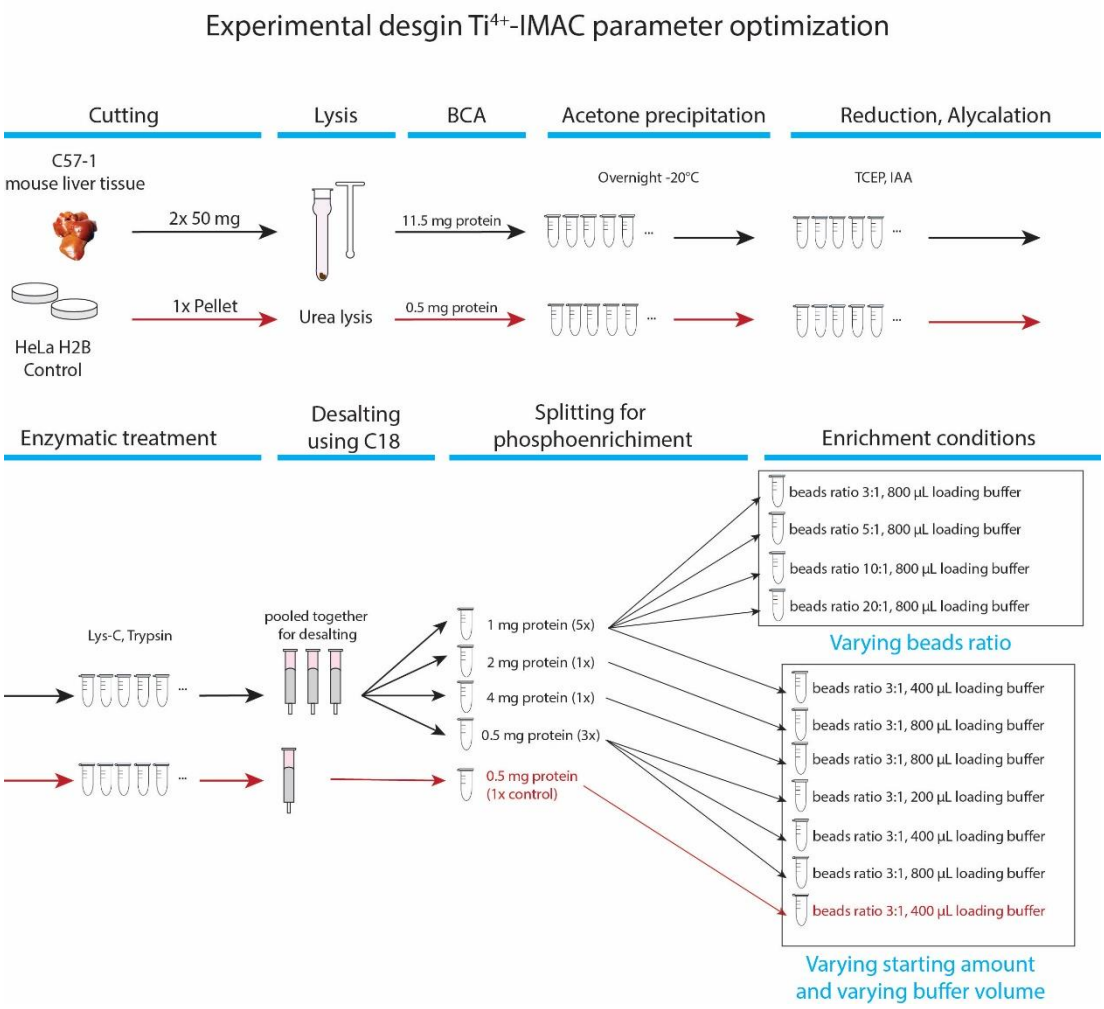


Figure 9: A brief overview of the experimental design of the Ti4+-IMAC optimization experiment. The experiment was conducted to test for the influence of the beads to starting material ratio, the increase of starting material with constant beads ratio, and the variation of the volume of the loading buffer. The variation in volume of the loading buffer let an alteration of the phosphopeptide concentration during the enrichment step which could influence the binding dynamics. The testing for the loading buffer was integrated into the testing of the other two parameter conditions, to decrease the number of samples. After the phosphopeptide enrichment step, the samples were measured on the OrbitrapElite. The acquired data were analyzed with iPortal, with the enabled three search engines, and with MaxQuant.

The protocol used for the phosphopeptide enrichment was described in the methods chapter 2.1 Experimental part (Page 16). A simplified workflow and the conditions for the samples, including the three altered parameters are shown in Figure 9. The samples were initially tested for contaminations on the LTQ Orbitrap XL. The spectra used for analysis were acquired on the OrbitrapElite by using the phosphopeptide enrichment settings described in the methods chapter 2.1.3 Mass Spectrometry data acquisition (Page 19). The injection volume was scaled and can also be found in chapter 2.1.3 Mass Spectrometry data acquisition Table 2 (Page 21).

Starting material

With increasing amounts of starting material an increased number of phosphopeptides could be identified (Figure 10). However, as the costs and effort increased significantly and the gain in peptides was attenuated above 2 mg, it was decided to use 1.5 mg starting material in the following experiments.

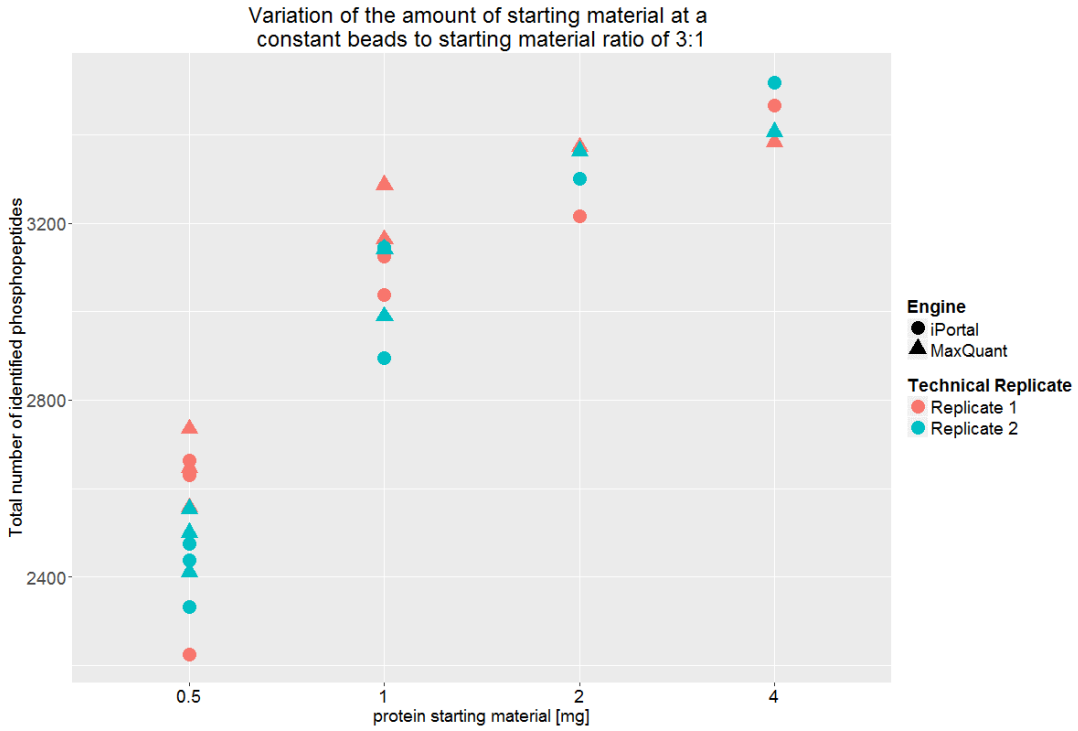


Figure 10: Variation of the amount of starting material at a constant beads to peptide ratio of 3:1. *Each of the filled circle shaped points represents a sample analyzed via the iPortal platform. The filled triangle shaped points represent the number of unique identified phosphopeptides found via the MaxQuant peptide identification search. The color code stands for the replicates, whereas red colored data points stand for the first replicate and blue colored for the second replicate. The protein amount, which was digested and subsequently used for one of the conditions, increased from 0.5 up to 4 mg per sample. As the beads ratio was kept constant the amount of beads increased with higher amount of starting material. The buffer volume for both, 0.5 mg and 1 mg, were not taken into consideration. All the conditions with the same starting amount and beads ratio were used for plotting the results.*

Beads to starting material ratio

To assess the best beads ratio, we identified the number of phosphopeptides and plotted the two replicates for each beads ratio (Figure 11). The results reflected that the previously used 3:1 beads ratio was in suboptimal region. If the acquired data were analyzed via the iPortal with Comet, X!Tandem and Omssa as enabled search engines, the best performance was identified at a beads ratio of 10:1. The average number of phosphopeptides for the two replicates was 3351. Further the results showed, that at a higher beads ratio of 20:1 the phosphopeptide enrichment performed worse compared to a ratio of 10:1. These results showed that the starting material to beads ratio did affect the enrichment selectivity to a large extend and an effective enrichment with Ti^{4+}-IMAC beads could be achieved by a specific beads ratio.

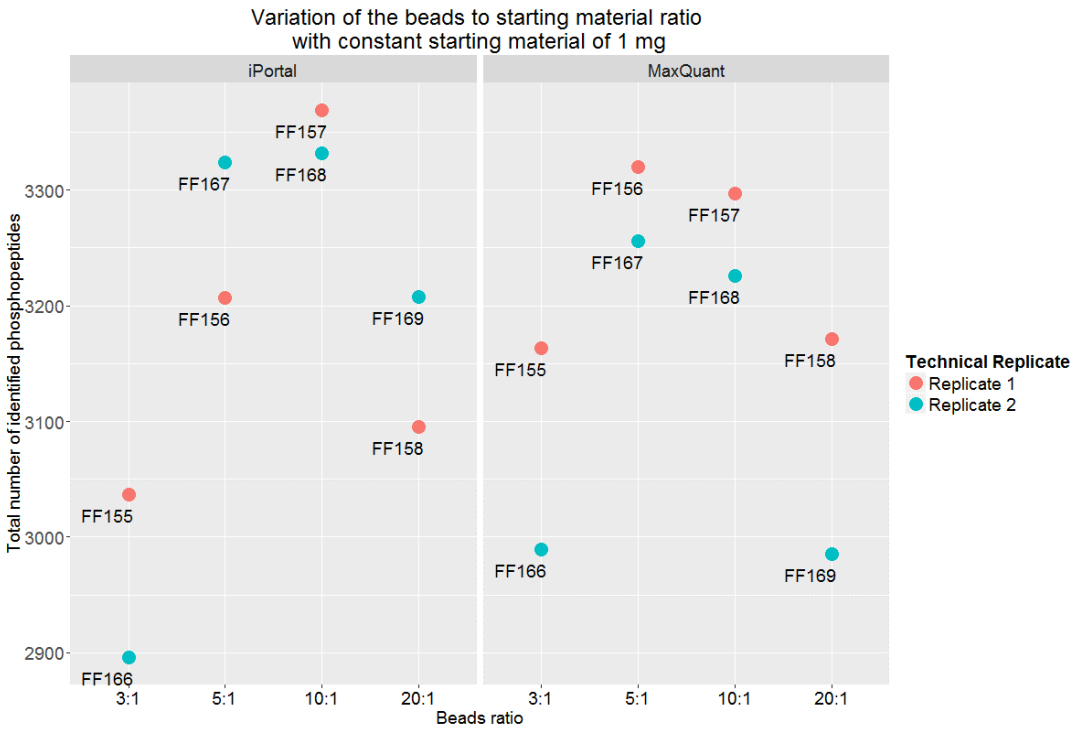


Figure 11: Result alteration of the beads to peptide ratio. *The red filled and blue filled points represent the first and second replicate, respectively. For this samples, the amount of starting material was kept at 1 mg and the beads to peptide ratio was changed to 3:1, 5:1, 10:1, and 20:1. For iPortal the highest number of unique phosphopeptides were identified with a beads to peptide ratio 10:1. Contrary to the iPortal identification workflow result, the MaxQuant result showed a higher number of unique identified phosphopeptides at beads to starting material ratio of 5:1.*

Loading buffer

As third optimization parameter the loading buffer volume was varied for several conditions of the 0.5 and 1 mg starting amount samples. The largest average differences for the 0.5 mg amount of starting material samples were identified between 200 µL and 800 µL of loading buffer volume, with 274 and 160 unique identified phosphopeptides for the peptide identification search result using the three search engines and for the MaxQuant search result, respectively. The figure is not shown, as the influence of the loading buffer volume was minor compared to the other results. In fact we needed 800 µL of loading buffer volume to ensure, that the increased amounts of beads and starting material used for the final conditions were efficiently dissolved during the phosphopeptide enrichment.

4.1.3. Final experimental workflow

The efforts made in the optimization experiment let to an improved parameter setup for the phosphopeptide enrichment of mouse liver tissue. We came up with a final workflow, which was suitable for the following large scale BXD mouse genetic reference population experiments. The experimental workflow is summarized in Figure 12. As starting material for the enrichment of one biological sample, 1.5 mg of reduced, alkylated, and digested protein was used. The beads to starting material ratio was increased to 10:1 as at this ratio better performance of the phosphopeptide

enrichment was obtained. By increasing these two parameters, it was necessary to increase the loading buffer volume up to 800 µL, to guarantee solubility during the phosphopeptide enrichment.

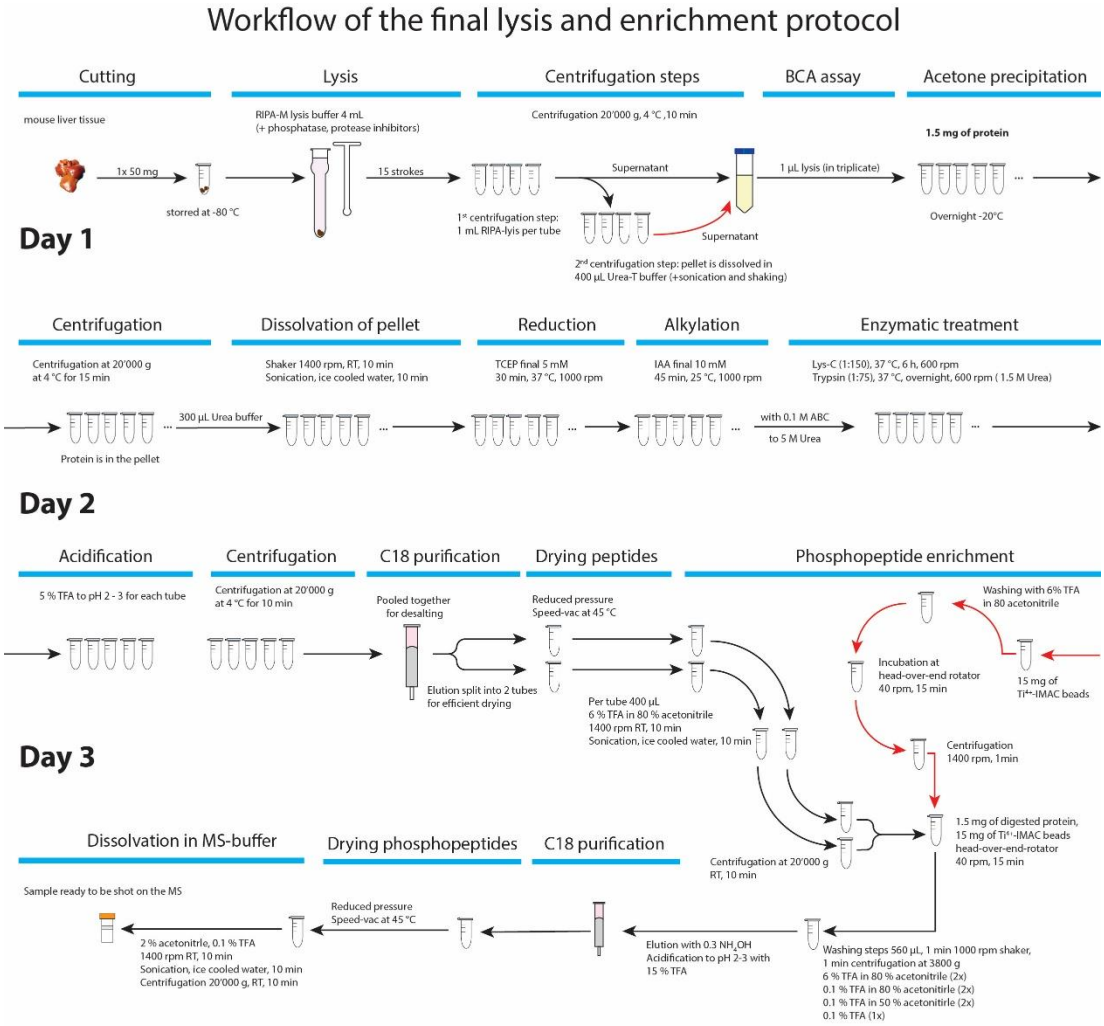


Figure 12: Final lysis and phosphopeptide enrichment protocol for mouse liver tissue. *The protocol last for three days. The figure illustrates this in the way that each row represents one day. The first row represents day one and the second day is shown in the second row. The third day of the protocol is represented by the last two rows, of working steps. At the first day the tissue was lysed and protein concentration was measured via BCA assay. For the phosphopeptide enrichment 1.5 mg of proteins were used as starting material. The 1.5 mg protein were separated into 2 mL tubes and 6 times the volume of -20 °C cold acetone was added. The protein was precipitated overnight at -20 °C. At the beginning of the second day the acetone was removed and reduction, alkylation and digestion of the proteins were conducted. The enzymatic treatment with trypsin was carried out overnight. On the third day of the protocol, the peptides were cleaned up via a C18 desalting step and the samples were dried. The dried samples were either stored at -20 °C, until a batch of samples were ready for phosphopeptide enrichment, or the enrichment was conducted on the same day. The already activated beads were washed and the samples were dissolved in 6 % TFA in 80 % acetonitrile. For the optimal beads ratio of 10:1, 15 mg of Ti4+-IMAC beads were washed and later on incubated with the samples for 1 hour on the head-over-end rotator. The beads with the bound phosphopeptides were extensively washed before the phosphopeptides were eluted. After the second desalting process the phosphopeptides were dried under reduced pressure. The dried phosphopeptides were stored at -80 °C until MS measurement or directly dissolved in MS-buffer. At this point the samples were ready for analysis on a high-end mass spectrometer.*

4.2. SWATH assay libraries

The phosphopeptide enriched mouse liver tissue samples were quantitatively measured in a highly reproducible manner with SWATH-MS. To extract the quantitative data from the acquired multiplex MS2 spectra, an assay library and the OpenSWATH software [36] is needed. Recently an improved version of the OpenSWATH software was developed that is intended to improve the performance for extracting data for peptides with PTMs. This improved version is called OpenSWATH/PTM (George A. Rosenberger, unpublished). To test the difference in performance, we used the two different software tools, to extract a test dataset. For the extraction, a SWATH assay library, which contains assays for sets of precursor ions, is needed. This information is retrieved from the performed searches of spectra acquired in DDA mode. Before the phosphopeptide identified from the peptide identification searches, could be used for the library construction, we performed a computational test to assign a confidence value to each assigned phosphosite. The correct localization of a phosphosite depends of observing one or few specific fragments in the MS2 of the DDA measurements. Few common search tools can estimate the false localization rate (FLR) of their search results. Therefore we used LuciPHOr2 in order to get an estimation of the FLR and filter only peptides with an FLR lower than 0.1 in the OpenSWATH workflow. As the OpenSWATH/PTM software contains some functionality to ensure correct localization, we compared both a phospho assay library that was filtered with LuciPHOr2 and an assay library that was not filtered. In summary, we performed three analyses of the testing dataset: i) LuciPHOr2 filtered library with OpenSWATH, ii) LuciPHOr2 filtered library with OpenSWATH/PTM, and iii) unfiltered library with OpenSWATH/PTM.

4.2.1. Building a phospho-SWATH assay library with LuciPHOr2

For the construction of the phospho-SWATH library the search results of 31 DDA injections were used. The annotated phosphopeptides identified in the search results were filtered with LuciPHOr2 by taking into account phosphopeptides with a FLR of lower than 0.1. By filtering with this threshold we lost about 30 % of the phosphopeptides, which was expected and is comparable according to the literature [42]. For the removed phosphopeptides the respective spectra did not provide the confidence to correctly assign the localization of the phosphorylation site (Table 3). However, this is a rather strict filter but we aimed to generate a high-quality assay library for which we have a good certainty for the localization of the modification. From the different observed spectra for the same peptide, a consensus spectra was generated and the 6 best transitions per peptide were selected for the phospho-SWATH assay library. The final phospho-SWATH assay library consisted of 2859 unique identified phosphopeptides from 1253 unique phosphoproteins.

Table 3: Number of unique detected phosphopeptides for each filtering step in the SWATH assay library building. *All phosphopeptides were excluded which did not full fill the selection criteria, to achieve a high quality assay library. As selection criteria were chosen the phosphorylation site localization score and, the quantity and quality of the transition ions for the phosphopeptide precursors.*

Assay library generation step	Unique phosphopeptides	Unique phosphoproteins
Combined 31 DDA TripleToF measurements	5245	1685
LuciPHOr2 (10% FLR)	3836	1585
phospho-SWATH assay library	2859	1253

4.2.2. Building the OpenSWATH/PTM libraries

To construct an assay library which can be used for the OpenSWATH/PTM extension, the same search results of the 31 DDA injections, as for the phospho-SWATH library, were used. The spectral library, for which all phosphopeptides with a FLR lower than 0.1 were removed, was used to build the filtered OpenSWATH/PTM library. The spectral library was converted to an OpenSWATH/PTM assay library, considering information for the residue specific adaptions. In our case the variable modifications were phosphorylation on serine, tyrosine, and threonine, plus oxidation on the methionine residue. Next, in the construction workflow for the OpenSWATH/PTM approach, the OpenSWATH Assay Generator was used to generate PTM assays. The charge of the fragment ions was limited from 1+ to maximum 4+ and the maximum alternative localizations was set to 20. In the final steps decoys were generated, by using the peptide sequences present in the library. The filtered OpenSWATH/PTM library consisted of assays for 2859 unique identified phosphopeptides from 1253 unique phosphoproteins.

For the unfiltered OpenSWATH/PTM library again the same search results of the 31 DDA injections was used as starting point. In this case the spectral library was constructed without applying site localization scoring with LuciPHOr2. This means no prior FLR threshold filtering was applied for this OpenSWATH/PTM library. From the consensus spectral library construction step on, the same construction workflow as above described for the filtered OpenSWATH/PTM library was used. The final unfiltered OpenSWATH/PTM library contained assays for 3399 phosphopeptides from 1170 phosphoproteins.

4.2.3. Comparison between the three SWATH assay libraries

To compare the performance of the libraries, the phosphopeptide enriched samples of the “aging” dataset, which were measured in DIA mode on the TripleToF 5600+, were extracted once with each library. For the phospho-SWATH assay library the OpenSWATH software was used, whereas for the two OpenSWATH/PTM libraries the OpenSWATH/PTM extension was used. Briefly mentioned, the test dataset consisted of twelve phosphopeptide enriched samples derived in triplicates from two

old and two young mouse liver tissue samples. Further information to the dataset can be found in chapter 4.3 Aging experiment (Page 58).

Unique phosphopeptides for each analysis step

The data, which were extracted with the three different SWATH libraries, were monitored during the various analysis steps. We used the number of unique phosphopeptides to compare the performance of the three libraries. The results for all libraries are listed in Table 4. The first two rows for each library provides information about the number of phosphopeptides extracted from each library. After extraction the data were process via SWATH2stats, were we applied tow filter criteria: i) phosphopeptides which were not detected at least in 2 of the 3 technical replicates were removed and ii) the resulting achieved peptide FDR had to be equal or lower than 0.01. The last row shows the union of all phosphopeptides in the final mapDIA output.

Table 4: The unique number of phosphopeptides and phosphoproteins for the three SWATH assay libraries for the OpenSWATH analysis and following analysis steps. *The number of unique phosphopeptides and phosphoproteins decreased during the analysis as the data were first filtered via SWATH2stats, before normalization and outlier removal was achieved via mapDIA.*

Analysis step	Unique phosphopeptides	Unique phosphoproteins
Phospho-SWATH assay library		
Union in all runs before filtering	2237	1041
On average in all runs before filtering in SWATH2stats (at least in 2 replicates, peptide FDR = 0.01)	$\bar{x}$ = 1485	$\bar{x}$ = 827
Union in all runs after filtering	1824	892
On average in all runs after filtering in SWATH2stats	$\bar{x}$ = 1410	$\bar{x}$ = 795
Union in all runs after mapDIA outlier filtering	1467	756
Filtered OpenSWATH/PTM		
Union in all runs before filtering	2351	1077
On average in all runs before filtering in SWATH2stats (at least in 2 replicates, peptide FDR = 0.01)	$\bar{x}$ = 1743	$\bar{x}$ = 946
Union in all runs after filtering	1792	879
On average in all runs after filtering in SWATH2stats	$\bar{x}$ = 1560	$\bar{x}$ = 869
Union in all runs after mapDIA outlier filtering	1657	823
Unfiltered OpenSWATH/PTM		
Union in all runs before filtering	3169	1024
On average in all runs before filtering in SWATH2stats (at least in 2 replicates, peptide FDR = 0.01)	$\bar{x}$ = 2251	$\bar{x}$ = 8173
Union in all runs after filtering	1696	657
On average in all runs after filtering in SWATH2stats	$\bar{x}$ = 1524	$\bar{x}$ = 914
Union in all runs after mapDIA outlier filtering	1650	657

To shortly summarize, the unfiltered and filtered OpenSWATH/PTM libraries performed equally and found around 1650 phosphopeptides. The OpenSWATH analysis conducted with the phospho-SWATH library, identified and quantitative 1467 unique phosphopeptides. Hence, the new OpenSWATH/PTM approach was more sensitive in detecting phosphopeptides, but the difference between the filtered and unfiltered assay library was negligible. Despite much more peptides were present in the assay library less phosphopeptides could be quantified with the unfiltered OpenSWATH/PTM library.

Amount of missing values

Another comparison criteria for extraction of quantitative data of the different SWATH assay libraries was the amount of missing values in the data. We had missing values, as for the OpenSWATH and the OpenSWATH/PTM analysis the re-quantification parameter was disabled. Re-quantification allows the algorithm to quantify peak-groups in one spectra by inferring the retention time of the peak-group from a spectra, where the peak-group was identified and quantified. The missing values were counted in the output file of the SWATH2stats package, after removing the phosphopeptides which were not quantified in at least two technical replicates. It was not possible to count them in the final result output of mapDIA, as the analysis tool inferred the quantitative value of the missing values, leading to a complete output matrix. The data quantified with the phospho-SWATH assay library had 24.20 % missing values, whereas the filtered and unfiltered OpenSWATH/PTM libraries with the OpenSWATH/PTM for data quantification contained 13.69 % and 7.33 % missing values, respectively. These results shows that the OpenSWATH/PTM algorithm was more sensitive and was able to quantify a more complete data matrix.

Regulated phosphoproteins

In addition, the log2 fold changes (log2FC) and adjusted p-values for each of the mapDIA datasets of the SWATH assay libraries were calculated. We considered all phosphopeptides with a log2FC +/- 0.5 and an adjusted p-value of equal or lower than 0.1 as potentially differently regulated in old mouse liver tissue due to aging. The log2FC was calculated by dividing all samples by the mean of the young mouse samples, followed by a log2 transformation. For the p-value calculation a pairwise t-test was conducted, by comparing the replicates of the young mouse liver tissue with the old mouse liver tissue. The p-values were corrected by using Benjamini-Hochberg.

With this method, we identified the up- and downregulated phosphopeptides in mouse liver tissue for the three results. We observed, that between the two OpenSWATH/PTM libraries, the number of regulated phosphopeptides altered only

by 3. The phospho-SWATH assay library showed 25 regulated phosphopeptides. The exact amount of up- and downregulated phosphopeptides for all libraries is shown in Table 5.

Table 5: Regulated phosphopeptides in mouse liver tissue in the three SWATH assay libraries. *A phosphopeptide was considered regulated if the adjusted p-value was below 0.1 and the log2 fold change of the average of the old mouse samples was +/- 0.5.*

Used library for the dataset	Upregulated	Downregulated
Phospho-SWATH assay library	18	7
Filtered OpenSWATH/PTM	18	8
Unfiltered OpenSWATH/PTM	19	4

Further it was analyzed, if the three libraries categorized the same phosphopeptides due to their quantification as regulated. The results indicated that out of the total 30 differently regulated phosphoproteins in all three libraries, 27 % or 8 were detected among all of them. The highest similarity with 53%, in terms of regulated phosphoproteins, was found among the two SWATH assay libraries constructed out of the filtered LuciPHOr2 list (Figure 13b).

Reproducibility

As mentioned before the dataset consisted of 3 technical replicate for each of the four biological samples. This enabled us to calculate the variability among all samples and within each replicates for the results of the three assay libraries. As Figure 13a shows, the variability was for all three libraries in the same range, regarding the CVs among all samples or the CVs for the replicates. What can also be seen is that the variability among the replicates was with an average CV of 26.68 % (+/- 0.79 %) lower than, compared to the average CV among all samples with 38.79 % (+/- 0.91 %). To summarize the violin plots, the result indicated that in terms of reproducibility all three libraries perform equally consistent.

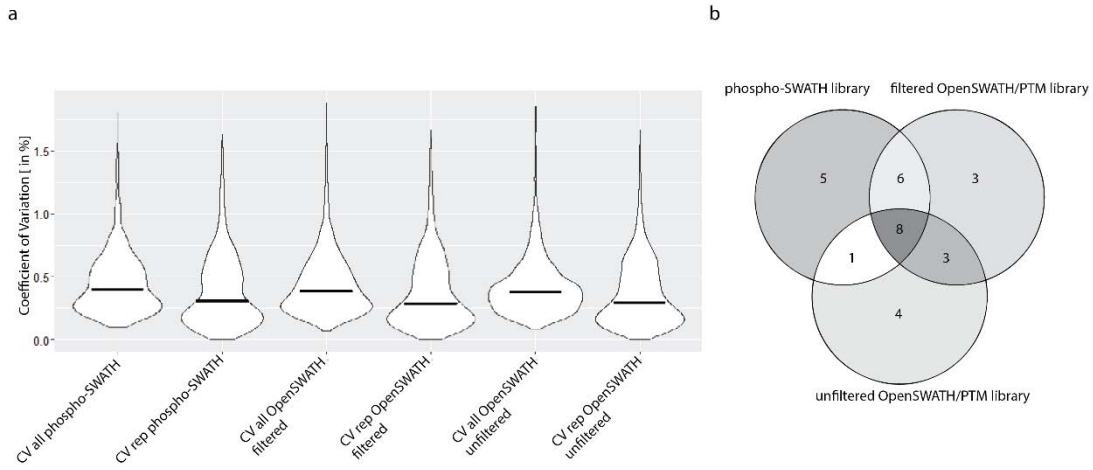


Figure 13: CVs for all SWATH assay libraries and Venn-Diagram of the regulated phosphopeptides for all SWATH assay libraries. a) *CV plots for the three SWATH assay libraries among all 12 samples (all) and among each of the four biological replicates (rep). The median value is represented by a black horizontal line. For all three libraries the median CVs over all samples analyzed with one library was on average 38.79 % (+/- 0.91 %). The libraries performed also comparable in terms of the median CVs within the replicates with an average of 26.68 % (+/- 0.79 %)* ***b)*** *Venn-Diagram of regulated phosphopeptides: 53 % of the phosphoproteins categorized in the phospho-SWATH assay library as regulated, were also categorized as regulated in the filtered OpenSWATH/PTM assay library. The similarity between the unfiltered OpenSWATH/PTM library and the phospho-SWATH library, in terms of regulated phosphoproteins, was only 33 %.*

Correlation of the intensities between the libraries

To reveal if the libraries quantified the intensities of the phosphopeptides in the same range, we correlated the intensities of all three mapDIA results with each other. The mouse, and replicate identifiers were added to the protein, and peptide identifiers to correlate each single detected phosphopeptide of each mouse replicate sample, with the intensity of the same identifier of another library. For the correlation the Pearson's product-moment was used. The three different mapDIA results correlated with an average Pearson's correlation of 0.88 (Figure 13a – c).

It is assumed, that the high correlation coefficient is achieved, because the data were processed by mapDIA, which performed outlier deletion and inferred quantities for missing values in the data matrix.

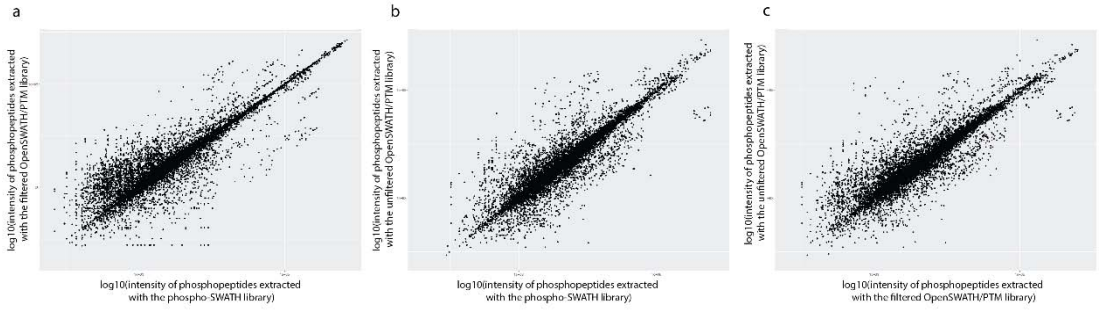


Figure 14: Correlation plots of the quantified phosphopeptides before and after mapDIA analysis. *The correlation plots* ***a) – e)*** *show the correlation of each of the three libraries with each other, after filtering by SWATH2stats, and mapDIA analysis including normalization and removal of outliers. The Pearson's product-moment correlation of the mapDIA results was on average R^2 0.88 (+/- 0.06).*

In summary the libraries performed nearly equally and there were no large differences between the OpenSWATH and the OpenSWATH/PTM approach. The final mapDIA results on peptide level, were for all three libraries quite comparable. The intensities of all three mapDIA results, correlate over 80 % with each other. In fact, the number of as regulated classified phosphopeptides, substantiated the proposition that the different libraries performed equally. For the analysis of the “aging” and “BXD-mouse reference population” experiments, the phospho-SWATH assay library with enabled re-quantification, was used. This library was selected due to the fact that the OpenSWATH/PTM approach was under ongoing development. Further, the refinement with LuciPHOr2 should have removed a large part of the phosphopeptides with ambiguous phosphosite localization. Therefore we gained an assay library, which consisted only of high-quality phosphopeptide assays.

4.3. Aging experiment

In order to estimate the technical variation of our final phosphopeptide enrichment procedure, we designed and conducted the “aging” experiment. As it was assumed, that the enrichment increases the variability, also the total proteome was measured. To achieve a dataset which allowed us to quantify the variability of our enrichment strategy, the two young, and two old mouse liver tissue samples were carried out in technical triplicates. By measuring the same phosphopeptide enriched samples with the DIA method SWATH-MS on the TripleToF 5600+ and with DDA on the OrbitrapElite, a comparison of DDA and DIA results was achieved. The total proteome was measured again with DIA method SWATH-MS.

For the characterization of the age related changes, we employed C57BL/6J mice samples, from 3 months old, so called “young”, and 22 months old mice, so called “old”. The mouse samples were already well characterized, for age related alterations on the metabolic and gen-expression level, as described in Houtkooper et al. [21].

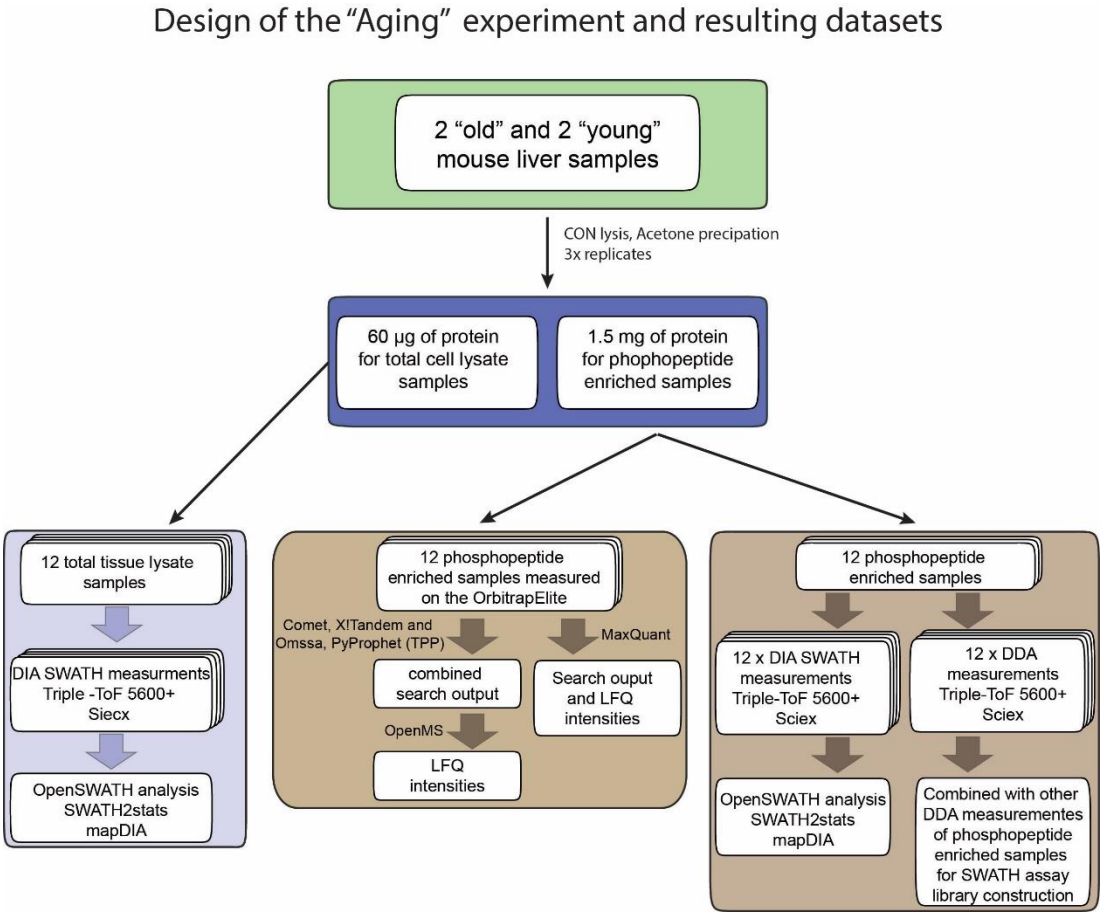


Figure 15: Experimental Design of the "aging" experiment, including the data analysis strategy. *For the total tissue lysate the three replicates of all four samples were measured in DIA mode on the TripleToF 5600+ and further processed as described in the methods chapter. The phosphopeptide enriched samples were also measured with SWATH-MS. The data were extracted by making us of the phospho-SWATH assay library. As control to check how the reproducibility for the phosphopeptide enrichment procedure was, the samples were also analyzed on the OrbitrapElite and processed with MaxQuant and OpenMS for a label free quantification of the phosphopeptide abundances.*

The phosphopeptide enriched samples were analyzed via the OpenSWATH pipeline using the phospho-SWATH library described above. For the analysis of the acquired total tissue lysate SWATH raw files and a SWATH assay library for total mouse liver tissue lysate was used [19]. The CV was calculated for the three replicates of one biological sample, and among all samples. Thus, the data provided a base for the estimation of variation for the phosphopeptide enrichment protocol, which was later on used for the "BXD-mouse reference population" experiment. Further the age related abundance changes of proteins and phosphoproteins were investigated. High-potentially differently regulated proteins or phosphoproteins, in old mouse tissue, were analyzed and if possible, related to known age related processes.

Analysis of total tissue lysate of mouse liver tissue

The total tissue lysate samples were measured on the TripleToF-MS in DIA mode. SWATH-MS allowed us to generate within a single measurement, a complete recording of the high resolution fragment ion spectra of all analytes, in the total lysate

mouse liver tissue. Therefore, we expected a low variability within the technical replicates. The multiplex MS2spectra were extracted with a sample specific assay library of total lysate mouse liver tissue [19]. The generated data were analyzed by using the recently published R package SWATH2stats. Outlier were removed by mapDIA, which also was used for signal normalization. During the analysis, the unique number of peptides was monitored. Table 6 shows the exact number for each analysis step of the total tissue lysate samples.

Table 6: Peptides identified and filtered during the analysis of the total tissue lysate of the DIA TripleToF 5600+ acquired spectra of the aging experiment.

Analysis step	Unique phosphopeptides	Unique phosphoproteins
Union of all measurements before filtering	16'136	3064
Extracted from phosphopeptide enriched SWATH "aging" experiment data	$\bar{x}$ = 12'722	$\bar{x}$ = 2583
SWATH2stats for the "aging" experiment (at least in 2 replicates, m-score = 0.01)	$\bar{x}$ = 12'562	$\bar{x}$ = 1257
Union of all measurements before filtering	14'851	2863
Union in all measurements after mapDIA outlier Filtering	12'157	2749

Analysis of phosphopeptide enriched samples of mouse liver tissue

The phosphopeptide enriched samples were measured in DDA mode on the TripleToF 5600+ and on the OrbitrapElite. For the peptide identification search of the DDA data from the OrbitrapElite we used MaxQuant and the iPortal platform with Comet, X!Tandem and Omssa as enabled search engines. The DDA data from the TripleToF 5600+ were analyzed via the iPortal platform with the same enabled search engines as for the Orbitrap analysis.

We first examined the number of unique identified phosphopeptides in each of the three different peptide identification search results. As the OrbitrapElite is optimized for DDA measurements we were able to detect about three times more phosphopeptides, compared to the average of 1050 (+/- 172) phosphopeptides detected in the DDA measurements of the TripleToF 5600+. This was expected, as the TripleToF 5600+ is not optimized for the measurement in DDA mode. However, the DDA measurements were needed to construct an assay library for the SWATH-MS measurements on the TripleToF 5600+. The differences between the two search engines, which were used for the DDA data were minor. With MaxQuant we could identify 3386 (+/- 90) phosphopeptides and with the iPortal platform with the enabled three search engines 3344 (+/- 269) phosphopeptides were detected. If the data of the iPortal output were not filtered, we detected in total 1966 unique phosphoproteins of 8685 phosphopeptides, within the 12 runs.

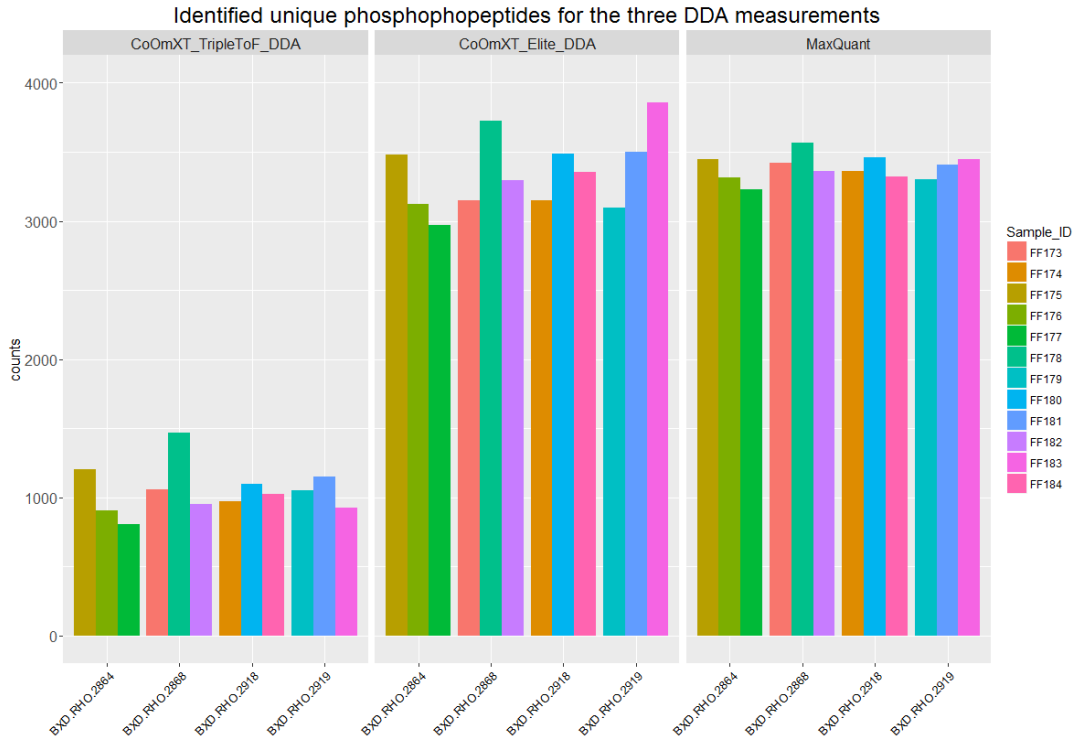


Figure 16: Unique identified phosphopeptides of different DDA measurements of the aging dataset search. *The illustration of the unique identified phosphopeptides shows, the expected performance differences in DDA mode on the TirpleToF 5600+ and the OrbitrapElite.*

SWATH and Label Free Quantification (LFQ)

The phosphopeptide enriched samples were measured a second time in DIA mode on the TripleToF 5600+. The acquired data were quantified by using the OpenSWATH workflow and the phospho-SWATH-MS library. The phospho-SWATH-MS library consisted in total of 31 DDA measurements, whereas 12 of them are the above described DDA measurements of the aging dataset measured with the TripleToF 5600+. The results for the extraction are shown in Table 7. The unique number of identified phosphopeptides was higher compared to DDA measurements with the TripleToF 5600+, as we combined all identified phosphopeptides in the 31 DDA measurements to construct an assay library.

Table 7: Phosphopeptides identified and filtered during the analysis pipeline of the DIA TripleToF 5600+ acquired spectra of the aging experiment.

Analysis step	Unique phosphopeptides	Unique phosphoproteins
Union of all measurements before filtering	2230	1151
Extracted from phosphopeptide enriched SWATH “aging” experiment data	$\bar{x}$ = 1489	$\bar{x}$ = 830
SWATH2stats for the “aging” experiment (at least in 2 replicates, m-score = 0.01)	$\bar{x}$ = 1414	$\bar{x}$ = 796
Union of all measurements after filtering in SWATH2stats	1823	889
Union in all measurements after mapDIA outlier correction	1802	882

Also the spectra acquired on the OrbitrapElite were further analyzed by using two different LFQ pipelines to gain quantitative information for all detected phosphopeptides. MaxQuant allowed detection and quantification in one analysis step. In addition, the peptide identification search result of the iPortal platform with the three enabled search engines, was quantified, by making use of the OpenMS LFQ pipeline. The exact parameters for both LFQ methods are described in chapter 2.2.1 Computational tools for proteomic data analysis (Page 22). The output of both tools was a data table containing protein identifiers, phosphopeptide sequence with suggested phosphorylation site(s), and the corresponding intensity signals. In Table 8 the unique number of phosphopeptides and phosphoproteins in the final result of the two methods are shown.

Table 8: Phosphopeptides and phosphoproteins detected and quantified in mouse liver tissue samples by MaxQuant and the iPortal integrated search engines of Comet, X!Tandem and Omssa by followed LFQ with OpenMS.

Analysis step	Unique phosphopeptides	Unique phosphoproteins
Union of all 12 measurements for the MaxQuant LFQ analysis	5286	2160
Union of all 12 measurements for the OpenMS LFQ analysis	3912	1610

With the MaxQuant proteomics identification and quantification tool it was possible to identify an additional 1300 phosphopeptides, compared to the peptide identification result of the three search engines, Comet, X!Tandem, and Omssa. As Figure 17a illustrates, the two different peptide identification searches had only 1001 delocalized phosphopeptides in common. These results represent the high variability of detected phosphopeptides among different search engine results.

The quantification of the found phosphopeptides also exhibited a vast variability, as the correlation between the quantified intensities of both LFQ signal intensities showed only a Pearson product-moment correlation of 0.66. Another criteria, for the quality of the search result was the data completeness. The OpenMS LFQ analysis contained 46.04 % missing values, compared to 35.92 % missing values for the MaxQuant LFQ output. In both cases, the major reason for the incomplete result table was due to semi-stochastic precursor selection of the DDA method. Therefore not in each of the 12 samples, the same precursors were selected for fragmentation. If a precursor was not measured within a sample, the peptide identification search could not find any peak and also LFQ was not possible.

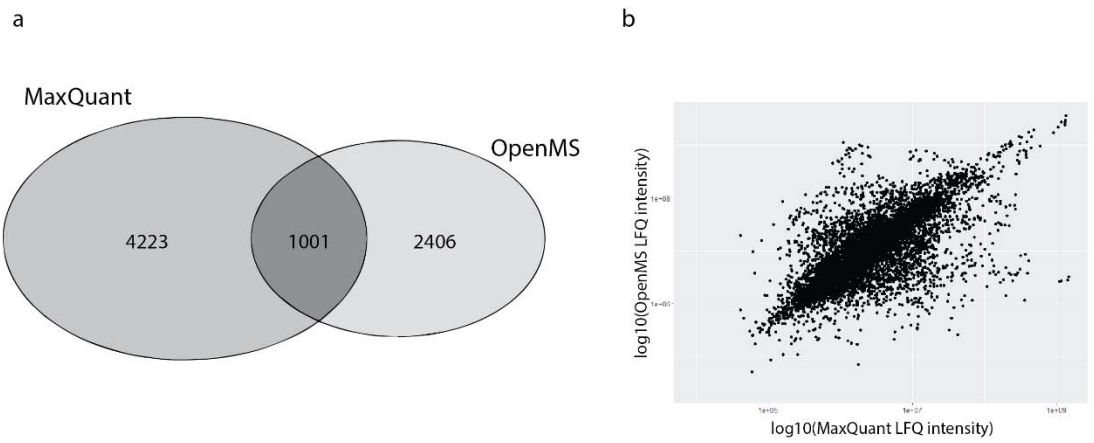


Figure 17: Venn-Diagram of the delocalized phosphopeptides and correlation of the intensities of the LFQ outputs. a) *For the Venn diagram the detected and quantified delocalized phosphopeptides of the MaxQuant and OpenMS output were used. The phosphorylation site identifier was removed within the peptide sequence and the counted phosphorylated sites were added after the sequence with a "_P" as identifier. 1001 delocalized phosphopeptides were identified and quantified via both LFQ pipelines. For the OpenMS the peptide identification search were done with the engines Comet, X!Tandem and Omssa.* ***b)*** *The intensities of the OpenMS and MaxQuant LFQ were correlated. The Pearson's product-moment correlation coefficient was R^2 0.66 between the intensities of both LFQ results.*

4.3.1. Reproducibility for DIA and DDA measurements of mouse liver tissue samples

The main focus of the "aging" experiment was, to estimate the CV among all 12 samples and within technical replicates, for all different MS techniques, used. Therefore the mean CV results of the phospho-SWATH MS data and the two LFQ data were calculated and each of the CV distribution was plotted as violin plot (Figure 18).

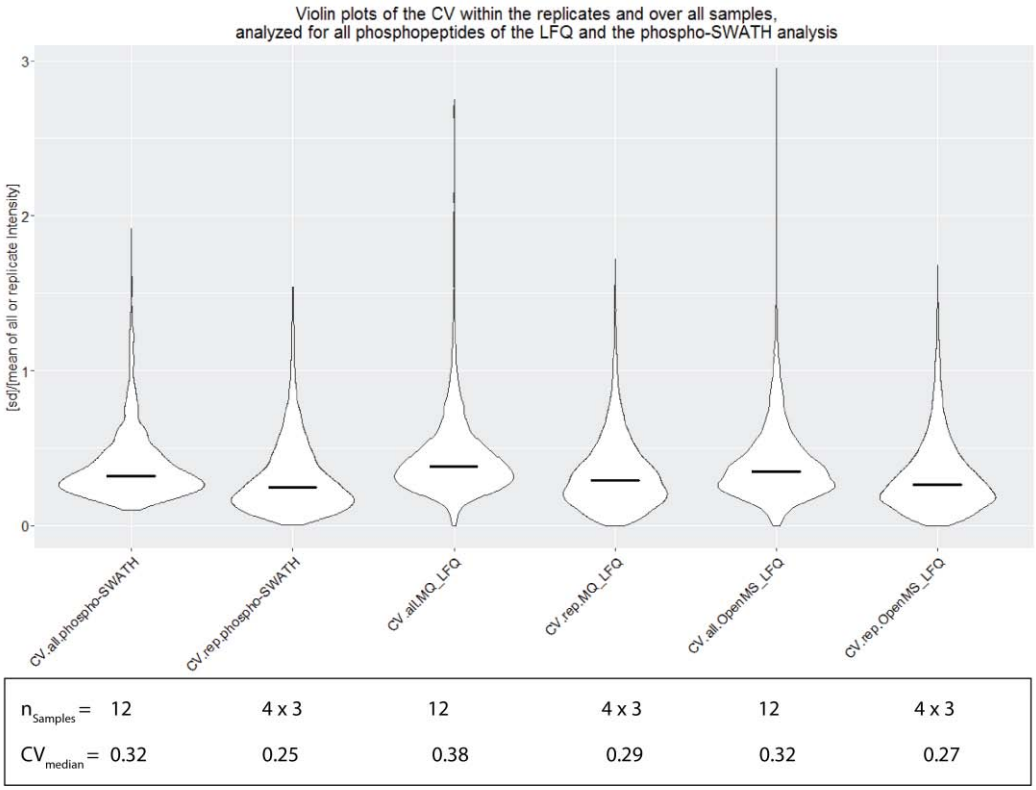


Figure 18: Violin plots of the CVs among all samples and within replicates for the phospho-SWATH-MS and the LFQ results of the DDA measurements. *The CVs over all data were calculated over all 12 samples. The CVs for the replicates were calculated within the three technical replicates per biological samples. The median CV for each violin plot is represented by the black line. The box shows the amount of samples and the median CVs for all violin plots.*

The data demonstrated that we were able to gain only a slightly better reproducibility between the technical and biological replicates with the phospho-SWATH library, compared to the LFQ results. The CVs in % were determined of the integrated peak areas (intensities) either across all 12 samples or within the replicates. The median CV for the SWATH-MS analyzed phosphopeptide enriched samples was 32.29 %, which was the lowest median CV among the three phosphopeptide datasets of the "aging" experiment. The same accounted for the CVs within the technical replicates, as again the SWATH-MS data showed with 24.84 % the lowest variability. Overall, the data indicated that with the phospho-SWATH-MS approach we obtained the capability to detect sufficient phosphopeptides across multiple measurements at an acceptable degree of reproducibility. However, the data also show, that the variability was for all MS techniques in the same range.

In addition we estimated the CV for the total tissue lysate samples of the "aging" experiment. We used these data to characterize the differences in variation between the SWATH-MS acquired total tissue lysate and the phosphopeptide enriched samples.

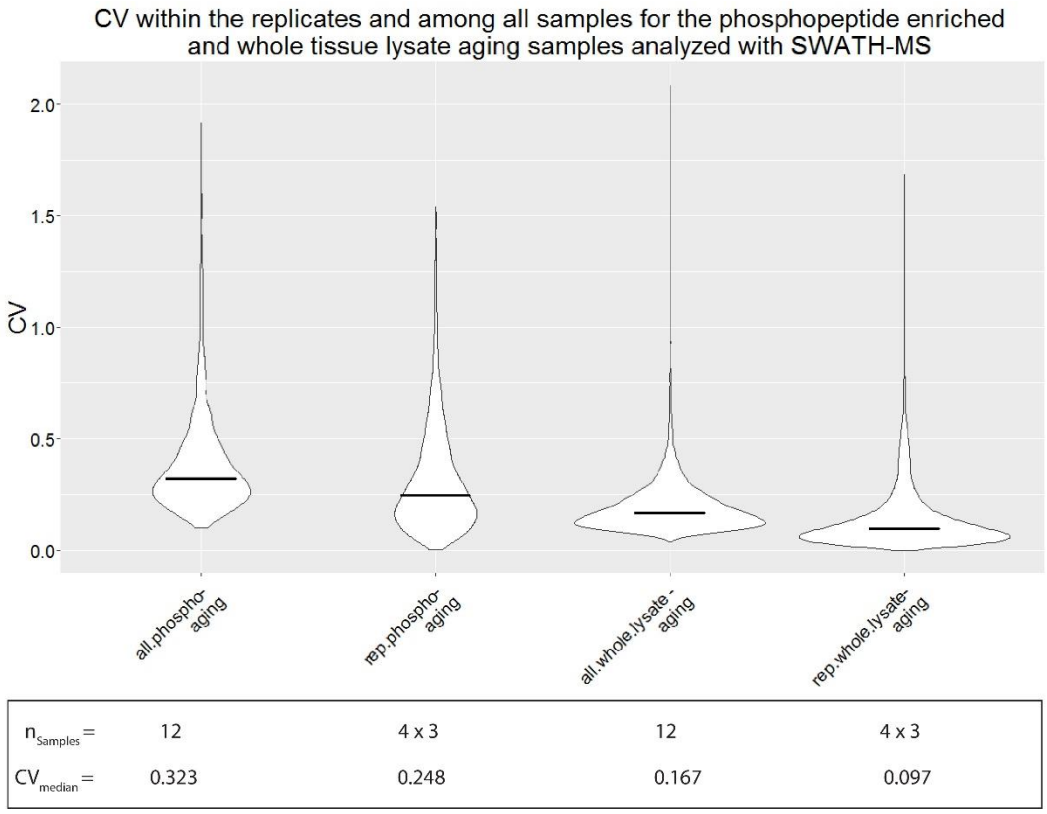


Figure 19: Violin plots of the CVs of the phosphopeptide enriched and total tissue lysate samples of the "aging" experiment. *The CVs among all samples and within the replicates were calculated for the total tissue lysate and the phosphopeptide enriched samples of the "aging" experiment. The median CVs are represented as black line within each violin plot.*

The median CV for the total proteome among all 12 samples was 16.73 %, whereas the median CV within the technical replicates of the four biological samples was only 9.72 % (Figure 19). By considering that the variability within technical replicates was lower, we concluded that we can derive meaningful biological effects, from the "aging" dataset. The CV of the total proteome was comparable to other total cell lysate SWATH-MS studies e.g. shown in SWATH-MS study of total yeast lysate [35]. The median CV of the phosphopeptide enriched samples was around 15 % higher, which could be explained by the enrichment procedure, which vastly increases the variability [42].

For the "BXD-mouse reference population" experiment it was planned to quantify the changes in abundance of a single phosphopeptides over 76 phosphopeptide enriched samples. In the aging dataset the variability among replicates was found to be lower than over biological samples. Thus, we inferred that the enrichment strategy, combined with the SWATH-MS approach, found to be suitable for the purpose of the large cohort mouse reference population.

4.3.2. Regulated proteins and phosphopeptides in old mouse liver tissue

To investigate changes in protein abundances due to aging in mouse liver tissue, we further analyzed the phosphopeptide enriched and total tissue lysate data, which were acquired with SWATH-MS. We expected that changes in the protein abundance can be identified and that we are able to detect regulated proteins within the total proteome data. However, for the phosphoproteins we measured single phosphopeptides with their phosphorylation site. Abundance changes within the phosphopeptide enriched samples were considered as indication for a regulated phosphosite.

Thus, the protein level output of the mapDIA analysis was used as an input to calculate the log2FC and an adjusted p-value to select for the regulated proteins and phosphoproteins, in the total tissue lysate and the phosphopeptide enriched samples, respectively. The pairwise t-test was conducted by comparing the technical replicates of the young mouse liver tissue with the old mouse liver tissue. The p-values were corrected by using Benjamini-Hochberg [76]. All phosphopeptides with an effect size +/- 0 .5 and adjusted p-value of 0.1 were considered as regulated.

In old mouse liver tissue 24 proteotypic phosphoproteins, of 32 phosphopeptides, were upregulated and 5 proteotypic phosphoproteins, of 5 phosphopeptides, were downregulated. The volcano plot in Figure 20c shows all, according to the used filter criteria, as regulated classified phosphoproteins in filled blue circles, while the filled red circles represent the phosphoproteins, which did not pass the thresholds. The volcano plot of Figure 20d shows the result for the total proteome. In the total proteome we were able to quantify 73 differently regulated proteotypic proteins in old mouse liver tissue. Form these regulated proteins, 44 and 29 proteotypic proteins were identified as up- and downregulated, respectively.

For the “aging” dataset we gained SWATH-MS data for the total proteome, as well as for the phosphoproteome. The Venn-diagram in Figure 20a shows, that we detected and quantified 289 unique proteotypic proteins in both datasets, which were only 10.76 % of the union of all proteins. Figure 20b shows that only one protein was characterized as regulated in the total- and phosphoproteome. Calcium-regulated heat stable protein 1 (UniProtKB/SwissProt - Q9CR86) was upregulated in the phosphopeptide enriched samples (3 phosphopeptides, two harbored the same phosphosite; one had a missed cleavage) and also in total proteome. This specific protein is an example for an overall protein upregulation. Our data pointed to the case, that the increased abundance in the phosphoproteome origins from the overall increase, and not from an increased phosphorylation.

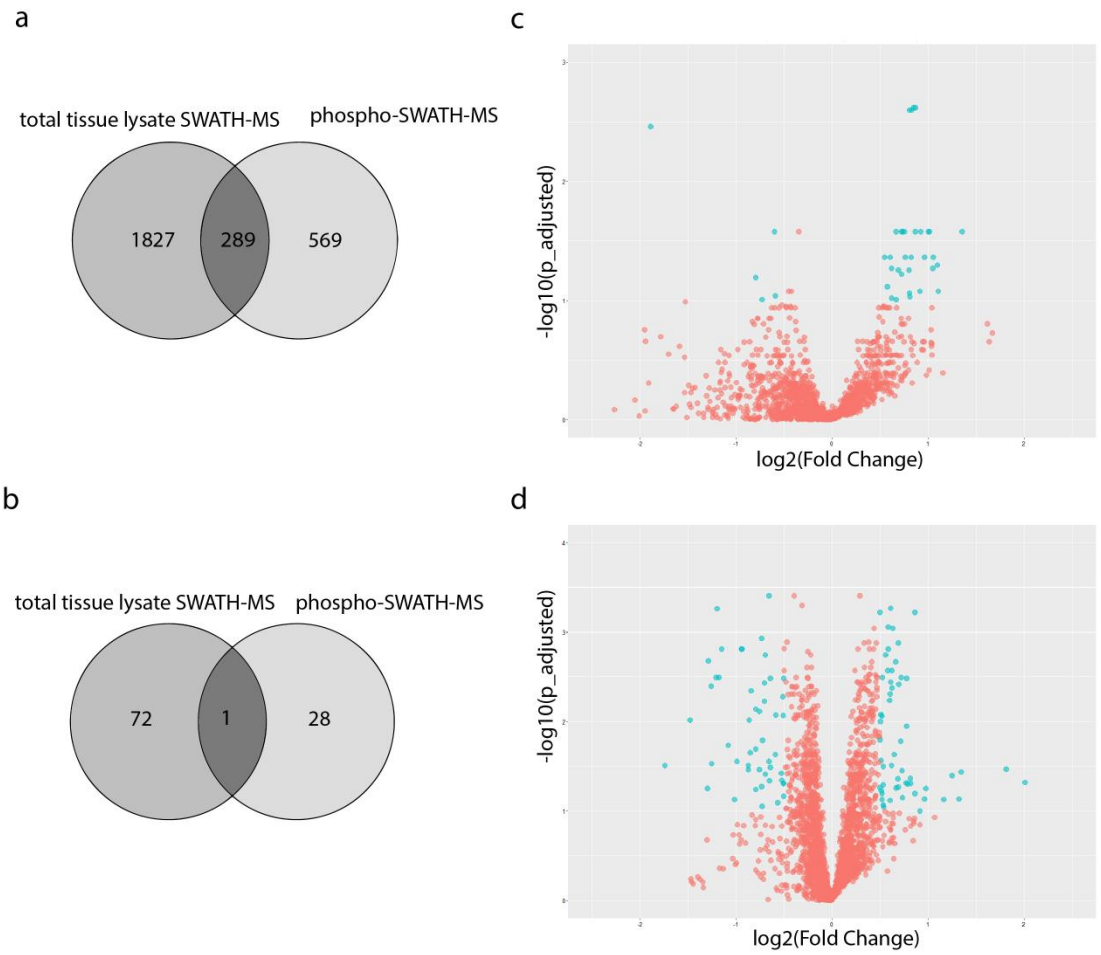


Figure 20: Regulated phosphoproteins and proteins in mouse liver tissue due to aging. a) *Venn-Diagram of unique proteotypic proteins and phosphoproteins detected in total tissue lysate and phosphopeptide enriched "aging" samples of mouse liver tissue by SWATH-MS and phospho-SWATH-MS, respectively.* ***b)*** *Regulated unique proteins and phosphoproteins in "aging" mouse liver tissue (log2(FC) > 0.5 and adjusted p-value < 0.1) in SWATH-MS and phospho-SWATH-MS analysis.* ***c)*** *and* ***d)*** *Volcano-plots were in blue filled circles the regulated phosphoproteins (c) and regulated proteins (d), in mouse liver tissue, are shown.*

Gene Ontology enrichment and molecular function revealed from the regulated proteins

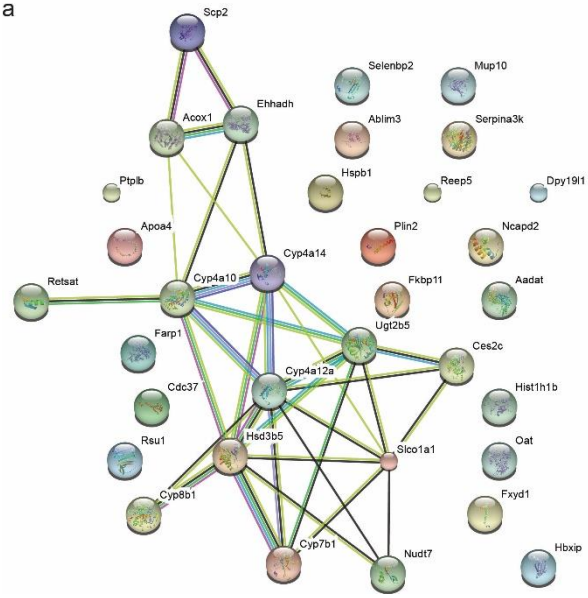
Next, we asked the question, which biological processes and pathways were differently regulated within old mouse liver tissue. Thus, we performed a functional annotation of the proteins regulated in total and mouse liver tissue. The functional enrichment analysis was conducted against the union of all detected proteins in the total and in the phosphopeptide enriched samples, which were 2685 proteins. First two subsets were passed for protein-protein interaction (PPI), pathway and molecular function analysis, to the STRING database [68]. The PPI-network is based on functional association and calculated by the STRING database. The PPI-network was exported from string without any alterations. For further information see chapter 2.2.4 . One subset for PPI-network and molecular function analysis, contained all upregulated proteins and phosphoproteins, and did not showed any enrichment for molecular functions or pathways against the union of all detected proteins. The

second dataset consisted of all in old mouse liver tissue, downregulated proteins and phosphoproteins. This uploaded dataset consisted of 34 different UniProtKB/SwissProt identifiers. The downregulated data were again tested for functional and pathway enrichment.

Table 9: Significantly enriched molecular functions (GO) and KEGG pathways in the subset of all downregulated proteins and phosphoproteins in old mouse liver tissue.

Molecular function (GO)		
Pathway description	Count in gene set	FDR
Oxidoreductase activity, acting on paired donors, with incorporation of reduction of molecular oxygen, reduced iron-sulfur protein as one donor, and incorporation of one atom of oxygen	3	0.00381
Alkane 1-monooxygenase activity	3	0.00381
KEGG Pathways		
PPAR signaling pathway	7	$5.1e^{-5}$
Retinol metabolism	5	0.00394
Fatty acid degradation	5	0.00495
Primary bile acid biosynthesis	3	0.0271

a



b

Network stats

number of nodes	34
number of edges	34
average node degree	2
clustering coefficient	0.901
expected number of edges	5
PPI enrichment p-value	0
network has significantley more interactions as expected	

Figure 21: Protein-protein interaction network of all downregulated proteins and phosphoproteins analyzed via the STRING database PPI-framework. *In* ***part a,*** *of the illustration, the PPI-network is shown. The different colors of the interaction network represent the different levels of evidence for protein-protein associations. Known interactions from curated databases were draw in blue and experimentally determined in purple. The predicted interactions were classified in three categories: from gene neighborhood in green, gene fusion in red, and gene co-occurrence in blue. Other categories of interactions were also considered and the edges were colored in different colors. Including yellow for textmining, black for co-expression and light blue for protein homology.* ***b)*** *Network statistics for the PPI-network shown in part a of the illustration. The network showed more interactions than expected from a randomly drawn network of the mouse genome.*

The subset showed an enrichment for two molecular functions, namely oxidoreductase activity and alkane 1-monooxgenase activity. The molecular functions were annotated via the Gene Ontology pathway database [77],[78]. Both molecular function annotations originated from the same proteins, the three cytochrome P450 family 4 proteins (CYP4A10, CYP4A12a and CYP4A14), which were all

downregulated in old mouse liver tissue. They are capable of oxidizing a variety of structurally unrelated compounds, including fatty acids and xenobiotics, and are therefore part of the xenobiotic metabolizing enzymes and transporters (XMETs). As shown by a study of Lee et al. (PLoS one, 2011), which investigated the gene expression changes XMETs through the life stages of the mouse, the cytochrome (CPY) P450 family changed significantly. XMETs can be separated into three categories, due to their reaction, and in which phase they are involved in the metabolism of xenobiotics. The phase I proteins consists mainly of monooxygenases, such as CPY which catalyzes oxidative metabolism, including also ω-hydroxylase activity, which uses an O_2 and 2 $NADP^+$, to oxidize in a three step reaction, the ω carbon of the fatty acid. Further the products of phase I are converted by phase II enzymes into amphiphilic anionic conjugates. This is achieved through proteins like glutathione transferases, sulfotransferases and UDP-glucuronosyltransferase [79]. Consistently we found the UDP-glucuronosyltransferase (UGT2B5) downregulated in old mouse liver tissue (Compare PPI- network in Figure 21). The glutathione S-transferases (GST) family, another phase II protein category were in our dataset found to be upregulated. We found three GST of two different families to be upregulated (GST theta 1 & 2, and GST mu 6) in the total proteome mouse liver samples. Further some transferases were found to be downregulated in aged mouse liver tissue, including ornithine aminotransferase (Oat), aminoadipate aminotransferase (Aadat) and retinol saturase (Retsat). Last mentioned showed a direct edge to the CYP protein cluster. The phase III deals mostly with the excreation and consist therefore of ATP binding cassette subfamily members, organic anion and cation transporters, and solute carrier [80]. And indeed, a few downregulated carriers were found in the dataset, including solute carrier organic anion transporter family member 1a1 (Slco1a1) and the sterol carrier protein 2 (Scp2).

Table 9 listed further four KEGG (Kyoto Encyclopedia of Genes and Genomes) pathways which were enriched in the downregulated proteins and phosphoproteins [81],[82]. In all pathways the different members of the CYP family were involved. The peroxisome proliferate activated receptor (PPAR) is one of the major regulators of the CPY family. Another, closely to the PPAR receptor linked pathway is the bile acid biosynthesis, which was downregulated in our dataset [83]. The pathway included the in the data detected two members of the Cytochrome P450 family (Cyp8b1, Cyp7b1) and the sterol carrier protein 2 (Scp2). This downregulation of these two specific CYP family members in the bile acid biosynthesis were also found in small fold changes on mRNA level due to aging [79]. Also significantly enriched in the downregulated subset, was the fatty acid degradation, which is well known and characterized as an age related process [21].

Finally all regulated phosphoproteins and proteins were submitted for PPI analysis to the STRING-database. The enrichment analysis of the 101 proteins was done against the union of all proteins detected in the SWATH-MS analysis of the aging samples. The resulting network corresponding statistics are shown in Figure 22. As indicated, the proteins in the network had significantly more interactions among themselves than what would be expected for a random set of proteins of similar size, drawn from the mouse genome. This means, the protein set used for the PPI seem to share more relation, than randomly picked proteins from the mouse genome.

Table 10: Significantly enriched KEGG pathways for regulated proteins in old mouse liver tissue.*As input all regulated proteins and phosphoproteins were taken into consideration. In total 101 proteins were used for the pathway analysis against the union of all detected proteins and phosphoproteins detected in the SWATH-MS analysis in this study.*

KEGG Pathways		
Pathway description	Observed gene (protein) count	FDR
PPAR signaling pathways	8	0.0112
Vascular smooth muscle contraction	6	0.0275

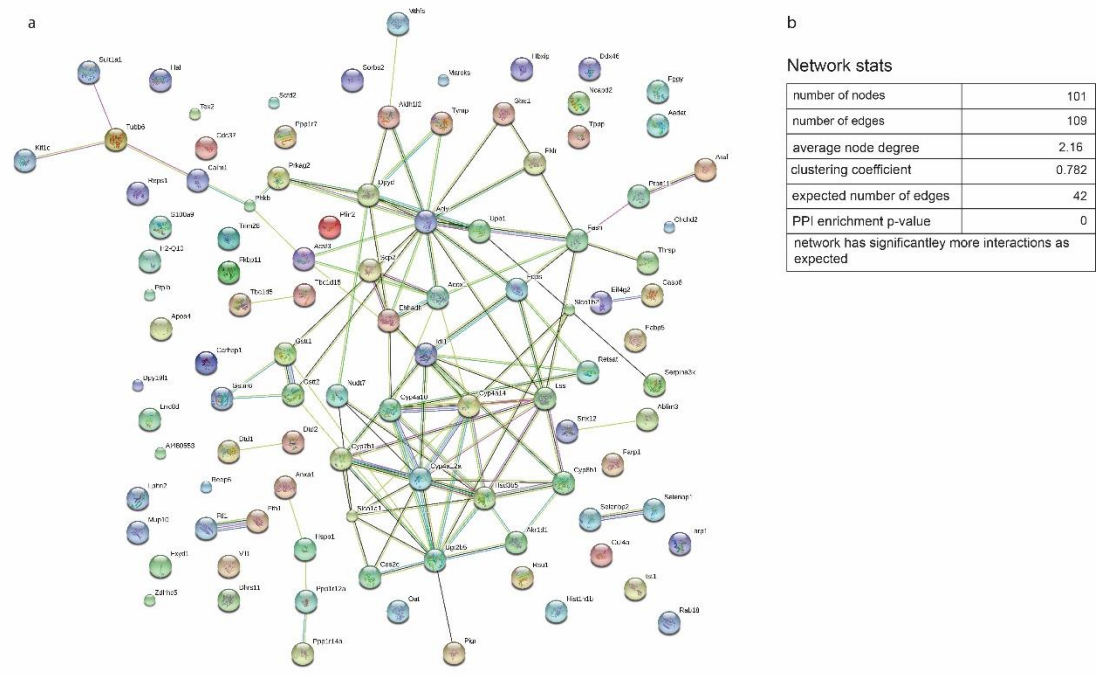


Figure 22: Protein-protein interaction network analyzed via the STRING database PPI-framework. ***a)*** *Shows the protein-protein interaction network of all regulated proteins and phosphoproteins of the SWATH-analysis results. The explanation for the colors of the connecting edges can be found in the caption of Figure 21.* ***b)*** *The statistical data for the drawn PPI network.*

For all regulated proteins only two KEGG pathways were identified. Again the PPAR signaling pathway was significantly enriched and the vascular smooth muscle contraction pathway, which included the three CYP of the family 4, subfamily a in the center of the network. The networks illustrate, that besides the downregulated members of the CYP families also metabolic proteins involved in fatty acid metabolism are differently regulated. The primary enzyme in synthesis of cytosolic-CoA, ATP

citrate lyase (Acly) was upregulated. Acly is also responsible for de novo lipid synthesis. Also the fatty acid synthase (FASN), which catalyzes for the formation of long-chain fatty acids from acetyl-CoA and other substrates, was upregulated. As described by Houtkooper et al. (Scientific Reports, 2011), during aging the fatty acid metabolism is perturbed, and it comes to an accumulation of plasma free fatty acids, whereas long chain acylcarnitines were lower in old mice [21]. Thus, the differently regulated proteins involved in fatty acid metabolism needs further investigation to come up with a consistent explanation.

Protein regulated by phosphorylation

All proteins, which were detected and quantified in the total tissue lysate and the phosphopeptide enriched samples, were further analyzed. We were checking if the phosphorylation changed due to protein abundances or due to increased phosphorylation. Therefore, the log2FC of the 12 proteins, detected in both SWATH-MS analysis and regarded as regulated in the quantified phosphoproteome, were compared to the log2FC of the total proteome and the resulting log2FC of the phosphopeptide in comparison to the protein level was calculated (Table 11). Only one protein, Q9CR86 (Calcium-regulated heat stable protein 1) was upregulated in the total and the phosphoproteome.

The other 11 proteins were not regulated above the 0.5 threshold of the total proteome. Thus, we used a further calculation to select the most probable due to phosphorylation regulated phosphoproteins. We calculated the log2FC difference between the total proteome and the phosphoproteome and set again a threshold of 0.5. All phosphoprotein with a log2FC difference above the threshold were considered as regulated due to phosphorylation (marked in gray in Table 11). One have to be aware, that for a high valid comparison one showed compared the abundances of the same phosphorylated and the non-phosphorylated peptide to each other. We could not compare exact the same peptide sequences to each other, as our datasets did not contain the same peptides for the protein and phosphoprotein data. Thus, we used the protein abundance from the protein level and compared it with the abundance of the phosphopeptide.

Table 11: List of phosphoproteins, for which the non-phosphorylated protein was also detected in the total proteome. *Marked in gray: Strong indices that a phosphosite was regulated through phosphorylation. Not marked: The abundance change*

Regulated in phospho-proteome and quantified in the total proteome	Regulated in phospho-proteome	Regulated in total proteome	Effect size in total proteome for old mouse liver tissue in log2	Effect size in phospho-proteome for old mouse liver in log2	Log2 (FC)	Fold Change
Q3UTJ2	Up		-0.133	1.098	1.23	2.35
Q9JJL3	Up		-0.288	0.920	1.21	2.31
Q62318	Up		0.138	1.110	0.97	1.96
Q62448	Up		0.011	0.846	0.83	1.78
P26645	Up		0.160	0.837	0.68	1.60
Q6ZQ58	Up		0.003	0.623	0.62	1.54
Q3UM45	Up		0.253	0.755	0.50	1.42
Q91V92	Up		0.414	0.830	0.42	1.33
Q9D1L0	Up		0.247	0.609	0.36	1.29
P35492	Up		0.394	0.670	0.28	1.21
Q9CR86	Up	Up	0.509	0.729	0.22	1.16
P32020	Down		-0.391	-0.726	-0.34	0.79

To summarize the “aging” experiment, we were able to estimate the variability of phosphopeptide enriched samples, we identified and quantified proteins and phosphoproteins, which showed differential abundance due to aging in mouse liver tissue. We further identified differently regulated proteins due to aging, and could identify known and unknown proteins involved, in the biological processes of aging. As the variability among biological replicates was higher than for the technical replicates, we proved that the current enrichment procedure was suitable to apply it to other experiments and studies.

4.4. BXD mouse reference population

In this experiment, we aimed to quantify the phosphoproteome of 40 strains of the BXD genetic mouse reference population treated with two different diets, to discover new phospho-pQTLs and phosphopeptides regulated by diet. We used the optimized phosphopeptide enrichment protocol and measured the phosphopeptide enriched samples via SWATH-MS. First, we looked for phosphopeptides regulated by diet by comparing the phosphopeptide abundance in the same strain between the two diets. Second, we identified phosphoproteins which most probably were genetically regulated, by correlating the abundances of both diets, per mouse strain and phosphoprotein, to all the other strains. With this subset of genetically regulated

phosphoproteins we performed a Haley-Knott regression analysis to identify cis- and trans-phospho-pQTLs [73].

Experimental design of the "BXD mouse reference population" experiment

In total, the experiment consisted of 76 mouse liver tissue samples, derived from 40 different strains of the BXD mouse reference population. For 37 BXD mouse strains, we obtained two mice samples, for which one mouse was on chow diet (CD) and one on high fat diet (HFD), from our collaboration partner (Auwerx laboratory, IPFL Genova). For three of the remaining strains, we obtained only the CD diet samples, and respectively for one the HFD. The used BXD mouse strains, with the exact identifiers are listed in the appendix Table 18.

For the lysis and enzymatic treatment, the samples were randomized and divided into 9 batches of 8 samples and one batch with 4 samples. For each mouse, 50 mg of pre-cut and weighted mouse liver tissue was lysed with the conventional method as described in the chapter 4.1.3 Final experimental workflow (Page 50). The data was analyzed as described in chapter 2.2.3 SWATH-MS analysis workflow (Page 30) and we were able to quantify 1997 phosphopeptides from 969 phosphoproteins (Table 12). As expected, the largest reduction in unique phosphopeptides was during the step of the extraction of the result with the SWATH assay library.

Table 12: Unique identified phosphopeptides and proteins for the data processing steps of the "BXD mouse reference population experiment". *Due to the strict filtering criteria in SWATH2stats, that the assay had to be quantified at least in 60 % of the data, we lost around 600 phosphopeptides. In the mapDIA analysis only a few outliers were removed.*

Type	Unique phosphopeptides	Unique phosphoproteins
Union of detected within all runs	2648	1190
Extracted from SWATH "BXD-mouse reference population " experiment data	$\bar{x}$ = 2033	$\bar{x}$ = 1073
SWATH2stats for the "BXD-mouse reference population" experiment (in at least 60 % of the samples, m-score = 0.01)	$\bar{x}$ = 1790	$\bar{x}$ = 979
Union in all measurements after mapDIA outlier correction	1997	969

4.4.1. Reproducibility within the BXD mouse liver samples

To compare the variability of the phosphopeptide abundance to the variation of observed in the samples of the previous experiments, the coefficient of variation of the data for all BXD samples, or separated for each diet, was calculated.

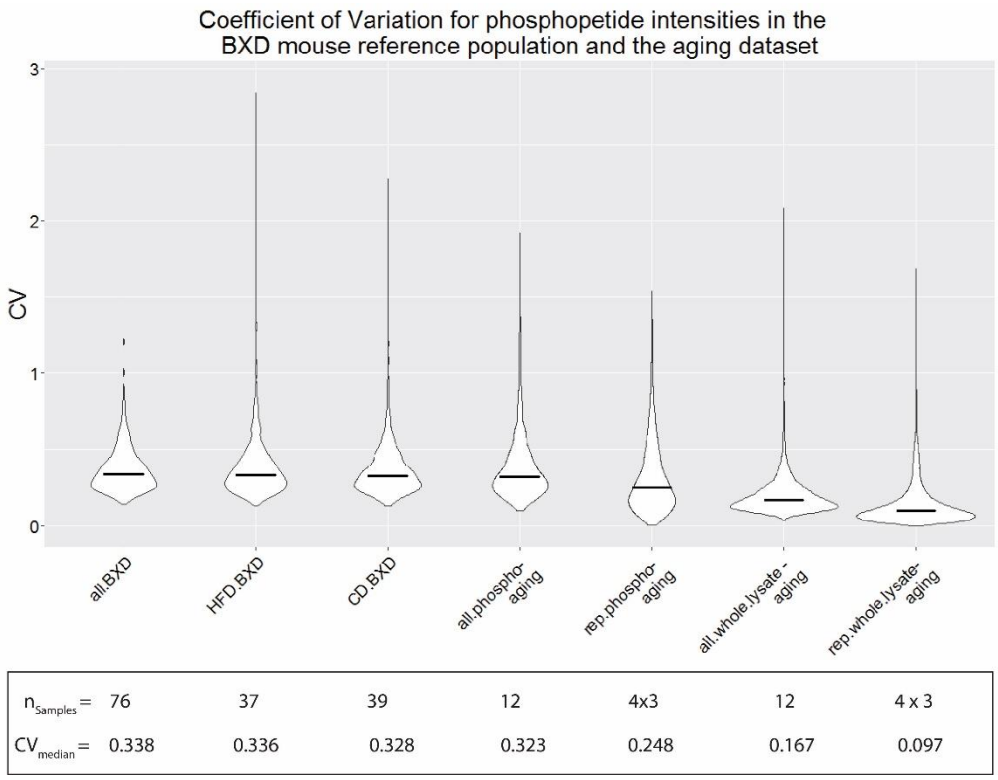


Figure 23: Violin plots of the CVs of the phosphopeptide enriched samples measured with SWATH-MS and analyzed by SWATH2stats and mapDIA. *In the plot, the median CVs are represented by a black line within each violin plot. The median CV values and amount of samples are shown in the box below the plot.*

As illustrated in Figure 23, the median CV among all BXD mouse reference population samples was 0.338, which was coherent with the median CV of 0.323 obtained in the various biological samples of the phosphopeptide "aging" experiment samples. Furthermore, the CVs of the two diverse diets of the BXD mouse reference population samples were separately plotted to exclude the influence of diet on the total variation. As expected, the reproducibility of the BXD mouse reference population was in the range of the "aging" experiment.

4.4.2. Mapping of phosphoprotein-QTLs

For all phosphopeptides we analyzed the two factors which were assumed to mainly influence the phosphoprotein abundance respectively the regulation of phosphosites. The two influencing factors were diet and genotype.

Differently regulated by diet

To test for the influence of diet, we performed a pairwise t-test between the log2FC transformed phosphopeptide intensities of all mice strains, for which we obtained HFD and CD mouse liver tissue samples. The p-value was corrected by the Benjamini Hochberg method. We identified 46 proteotypic phosphopeptides of 33 phosphoproteins, by filtering with a log2FC threshold of +/- 0.5 and an adjusted p-value below 0.01. For one phosphoprotein the result is illustrated in Figure 24a. This set of proteins regulated by diet did not show a PPI-network with significantly more

interactions than expected from a random set of genes of the same size, if the PPI analysis was performed against all phosphoproteins detected in the BXD mouse reference population experiment. Further, no functional or molecular pathway enrichment was obtained

Differently regulated by the genotype

Of more interest for us was to identify phosphoproteins for which the phosphosite showed a high chance to be regulated due to the genotypes of the different mouse strains. Thus, we correlated the intensities of the phosphopeptides of all mouse strains, for which we received both diverse diets. The data were filtered by a Spearman correlation coefficient > 0.5, leading in total to 65 phosphopeptides. From these identified phosphopeptides 63 were proteotypic and originated from 49 phosphoproteins.

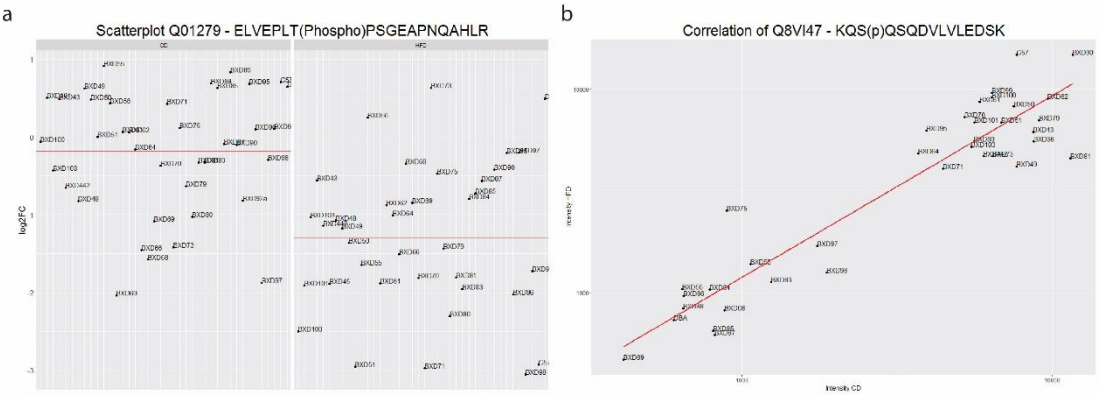


Figure 24: Graphical illustration of an example of a phosphoprotein regulated due to diet and one due genotype. *a) Phosphoprotein regulated due to diet. The mean log2FC for CD and HFD are indicated by a red horizontal line. It was a randomly chosen phosphoprotein from the list of phosphoproteins, which were identified to be regulated by diet. The plot shows, that within both diverse diet, the phosphoprotein abundance of Q01279 did not follow in all BXD strains the same trend of being up- or downregulated. B) The Spearman correlation coefficient between the mouse pair samples was 0.81 for Q84I47, a randomly picked phosphoprotein form the list of genetically regulated phosphoproteins. A further evidence, that the changes in abundance were due to the genotype was, that the two parental strains clustered separately. The parental DBA strained clustered in the downregulated region (lower left corner in the graph), whereas the other strain C57/6B cluster in the upregulated cluster (upper right corner of the graph).*

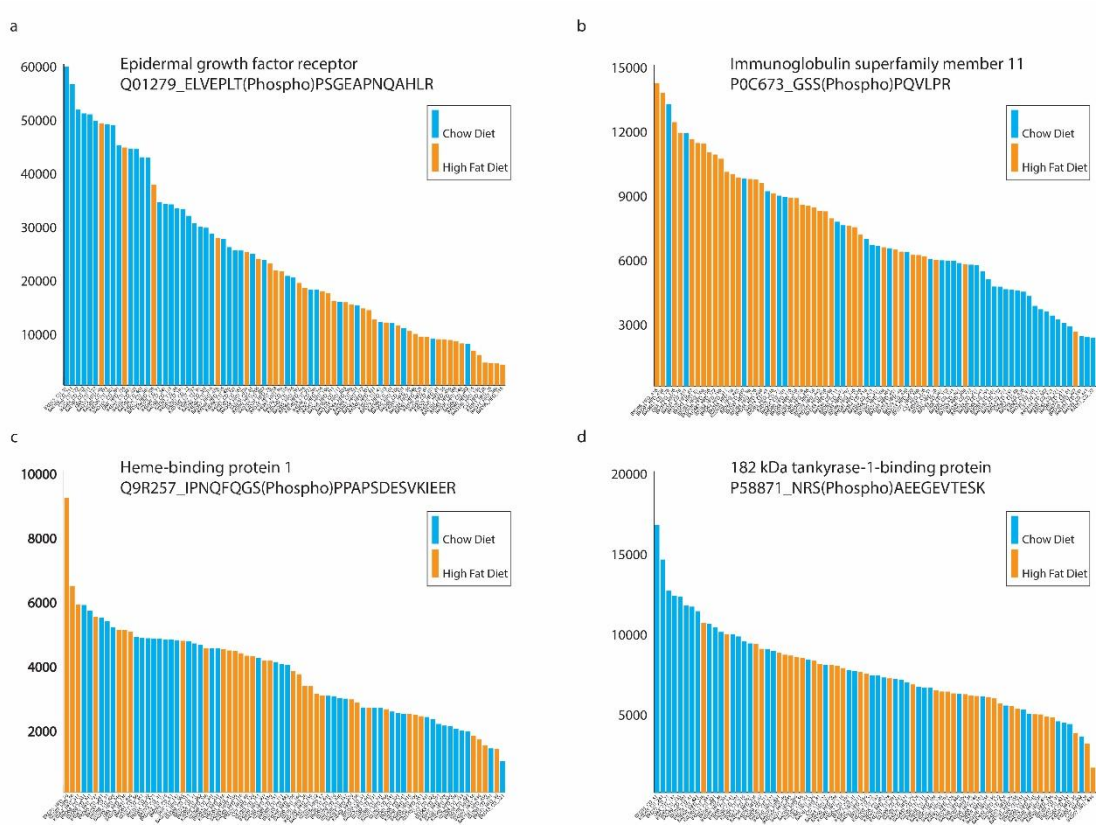


Figure 25: Sorted intensities of phosphoproteins regulated either due to diet or genotypes for all mouse liver tissue. *The column graphs in **a and b** show two randomly picked phosphoproteins which were identified to be regulated by diet. Intensity for all mouse strains were sorted from highest to lowest. The diet is indicted by color, whereby blue indicates CD and orange HFD. It can be seen, that except for some outliers either one of the feeding strategies was found to lead to higher abundances, for the phosphoproteins. In contrast, the column graphs in **c and d** show two genetically regulated phosphoproteins. Obviously, no separation between the diets can be seen, as for these phosphoproteins the changes in intensity originated from the genotype.*

Identified phospho-pQTLs

For the phosphoproteins, which were identified as genetically regulated, a phospho-pQTL analysis was conducted as described in 2.2.5 Mapping of phospho-pQTLs (Page 35).The phospho-pQTL analysis was separately performed for each diet. In CD we identified 27 trans-phospho-pQTLs. To account as trans-phospho-pQTL, the identified trait locus had to pass a p-value threshold less than or equal to 0.05. The location of the trait locus had to be either identified on another chromosome, or showing at least a distant difference of more than 10 Mb to the gene locus of the quantified phosphoprotein used for mapping. Further, within the CD analysis 15 cis-pQTLs were identified. The p-value threshold for a cis-pQTL was set to less than or equal to 0.67. The distant constrain was set to the maximal resolution of the identified marker and was therefore set to 10 Mb around the gene locus of the phosphoprotein used for QTL mapping. The identified cis-pQTLs detected within the CD mouse samples of the current phosphopeptide enrichment study were further evaluated by checking if they were discovered also as cis-eQTLs or cis-pQTLs in the previous study by Wu et al. [18]. For 8 of the mapped 15 cis-pQTLs we identified cis-eQTLs or cis-

pQTLs, either previously detected in HFD or CD samples of the BXD mouse reference population. The results of the comparison of the previously mapped cis-QTLs are shown in Table 13.

Table 13: Discovered cis-pQTLs in phosphopeptide enriched samples of the BXD mouse samples fed with CD, which were also identified in a previous study of Wu et al. (Cell, 2014). To validate the mapped cis-pQTLs in the phosphopeptide enriched samples, they were compared to the cis-eQTLs and cis-pQTLs of HFD and CD detected in the previous study. If a cis-QTL was found in the previous, the table indicated this with TRUE. These results showed that for 8 cis-QTLs we had evidence either on the gene expression or on the total *proteome level.*

Protein	Gene	cis-eQTL_CD	cis-eQTL_HFD	cis-pQTL_CD	cis-pQTL_HFD
Q9R257	Hebp1		TRUE	TRUE	TRUE
Q9CR00	Psmd9		TRUE	TRUE	TRUE
Q8VI47	Abcc2	TRUE	TRUE		TRUE
Q8BYU6	Tor1aip2	TRUE		TRUE	TRUE
Q80XI3	Eif4g3	TRUE	TRUE		
Q64464	Cyp3a13	TRUE	TRUE	TRUE	TRUE
P46935	Nedd4			TRUE	
P10518	Alad	TRUE	TRUE	TRUE	TRUE

The mapping of cis- and trans-QTLs in the HFD BXD mouse samples were filtered with the same thresholds, used for the CD BXD mouse samples. We identified 21 trans-pQTLs in the current study. In addition, 17 cis-pQTLs were mapped within 10 Mb of the gene locus of the analyzed phosphoprotein. The cis-pQTLs identified in HFD samples in the current study were as compared to the cis-eQTLs and cis-pQTLs of the previous study. 8 of the cis-pQTLs had been mapped also mapped in the previous study as cis-QTLs. The comparison results are shown Table 14.

Table 14: Discovered cis-pQTLs in phosphopeptide enriched samples of the BXD mouse samples fed with HFD, which were also identified in a previous study of Wu et al. [18] *The table shows if the detected cis-pQTLs were previously detected as cis-eQTLs and cis-pQTLs in one of both diets.*

Protein	Gene	cis-eQTL_CD	cis-eQTL_HFD	cis-pQTL_CD	cis-pQTL_HFD
P10518	Alad	TRUE	TRUE	TRUE	TRUE
Q8VI47	Abcc2	TRUE	TRUE		TRUE
Q64464	Cyp3a13	TRUE	TRUE	TRUE	TRUE
Q80XI3	Eif4g3	TRUE	TRUE		
Q9WU19	Hao1			TRUE	TRUE
Q9CR00	Psmd9		TRUE	TRUE	TRUE
P58871	Tnks1bp1	TRUE	TRUE		
Q8BYU6	Tor1aip2	TRUE		TRUE	TRUE

From the 1997 quantified phosphopeptides we used a subset of 65 phosphopeptides for the QTL analysis. With this subset we were able to identify 63 genome loci, which correlated to the quantitative signals of the phosphopeptides. We discover 6 trans-

phosphoprotein-QTLs in both diets, 21 only in the CD and 15 only in the HFD data. Further we analyzed 11 cis-phosphoprotein-QTLs in both diets, leaving 4 which were only detected in CD and 6 in HFD. Overall we were able to detected 42 different phosphoprotein-trans-QTLs and 21 cis-phosphoprotein-QTLs. In the BXD samples on CD diet, 12 proteins mapped to multiple loci, and 15 proteins only mapped to one loci. In the HFD samples, we correlated 5 proteins to multiple loci on the genome, and identified 26 proteins which correlated only to one distinct locus. For the BXD mice on HFD diet, 5 proteins were linked to multiple loci and we further identified 26 proteins which correlated only to one distinct loci. If the two data sets are combined 31 phosphoprotein map to only a single phospho-pQTL and 18 phosphoproteins were linked to multiple loci.

5. Discussion

In this study, we quantified the liver tissue phosphoproteome from 76 mouse strains of the BXD mouse genetic reference population under two different dietary conditions. As far as we can tell, no phosphopeptide specific DIA dataset is published yet, and the used workflow represents a feasible way to study the phosphopeptide enriched samples over a large cohort of samples. The acquired data enabled us to estimate abundance changes on the phosphoproteome due to genotype, diet, and gene-by-environment interactions. We were able to link 63 different phosphoprotein QTLs with phenotypes, including 5 newly discovered cis-phospho-pQTLs and unknown trans-phospho-QTLs. Further, 16 phosphoproteins were mapped to eQTLs or pQTLs which had been identified by the previous multi-omics studies of the transcriptome, proteome, and metabolome [19], [18].

5.1. Identification of a high-quality phosphoproteome in mouse liver tissue

To quantify the phosphoproteome we used selective enrichment of phosphopeptides and measured the samples on high-resolution LC-MS/MS systems. Benefiting from advances in enrichment strategies, phosphoproteomics techniques now permit the large-scale identification and quantification of protein phosphorylation sites. These improvements were used to quantitatively measure 10’000 of phosphorylation sites for more than 1000 proteins in different mouse tissue [84]. Furthermore, previous studies of the phosphoproteome in mouse liver were able to reveal 5600 to 7400 non redundant phosphorylation sites of 2300 phosphoproteins [43], [46]. The phosphopeptide enriched samples of the BXD mouse reference population were measured in DIA mode on the TripleToF 5600+. The acquired spectra were quantified by using OpenSWATH and the phospho-SWATH assay library. With our workflow, we were able to quantify 1997 unique phosphopeptides of 969 phosphoproteins in all samples of the BXD mouse reference population using one injection per sample. The huge difference in number of quantified phosphopeptides between the studies stem from various issues.

First of all, we used a single enrichment step and did not fractionate the samples. A common approach in phosphoproteomics for large-scale analysis is to perform first strong cation exchange chromatography (SCX) and do a subsequent phosphopeptide enrichment of each fraction. Such methods highly increase process time on the experimental side as on the machine time for acquiring the data. The advantage of this method is the reduction of the complexity per sample and the increased sampling time per biological sample, which leads to a huge increase in detected

phosphopeptides. This approach with TiO_2 and IMAC as second enrichment step lead, with the offset of longer measuring times, to more comprehensive analysis of the phosphoproteome [43], [46], [84]. It should also be mentioned, that other studies, only using a single-step phosphopeptide enrichment strategy with Ti^{4+}-IMAC or TiO_2 beads, reported quite impressive results by detecting 12,799 unique phosphosites in Jurkat T cells or with the EasyPhos method in mouse liver tissue more than 20,000 distinct phosphopeptides [85],[86]. However, the coverage was achieved by multiple injections or vast number of technical replicates, and both studies characterized the changes of a distinct biological pathway with more than 100 MS runs. In this study we used only one injection for each of the 76 samples.

Not only fractionation and the number of injections contribute largely to the coverage of the phosphoproteome, another factor is the gradient of the MS analysis. For the analysis of the SWATH-MS data we used a 90 min gradient. By including the 31 DDA measurements used for the assay library construction, which were measured with a 120 min gradient, we had a total measurement time of 176 hours. We were able to use a short gradient for the DIA measurements, as theoretical all precursors were selected for MS2 measurements. Conversely, in DDA a longer gradient significantly increases the number of detected phosphopeptides. Thus, an approach is to increase the gradient to 180 min or even to 240 minutes and further injection samples twice to increase the coverage of the phosphoproteome [46], [86].

Another factor which hugely influences the number of detected phosphorylation sites is the measurement method. For our approach we used a DIA method with the enabled re-quantification in OpenSWATH, which provides us with a resulting matrix 0% missing values, as the abundance re-quantification algorithm of OpenSWATH allows to infer peak-group information (e.g. retention time) to search for peaks which were not directly identified by OpenSWATH [36]. Whereas, the above mentioned large-scale studies all used a DDA approach and therefore had to deal with the inherent semi-stochastic sampling issue of this method. Some studies addressed this issue, by using “match between runs” in MaxQuant, which can transfer MS/MS identifications between measurements [86], [85]. Another way to overcome this issue is the above mentioned multiple injection of a samples and combining the results [46].

For the current study, we aimed to generate high-quality data, with confident localization of the phosphosites. Several tools and approaches are available to estimate a site localization score. Thus, we filtered the peptide-identification search result, which were controlled by a ProteinProphet FDR of 0.01, with a LuciPHOr2 FLR threshold of 0.1, considering only localization sites with a high score. This rather strict filtering provides us with a high-quality phospho-SWATH assay library. It is rather

difficult to compare the different studies in terms of site localization confidence, as they all used different peptide identification tools and various scores for filtering or classifying the phosphorylation sites. Various studies used MaxQuant site localization scores and filtered or sort the phosphorylation sites by different thresholds [85], [86]. One has to be aware of the site localization issue, and carefully interpret the results of large-scale studies. A rule of thumb is, that a sizeable percentage of 20 – 40 % of identified phosphopeptides cannot be automatically localized with a high confidence [42]. We filtered about 30 % of the phosphopeptides due ambiguous localization by applying the LuciPHOr2 threshold.

In summary, we can state that the number of identified phosphopeptides and phosphoproteins in our dataset is significantly lower compared to other published studies of the phosphoproteome of mouse liver. Nevertheless, the used DIA approach suited perfectly for our intention, as we aimed to obtain a high-quality and complete dataset over the 76 samples, instead of gaining more identified phosphopeptides with less confident site localization, and missing values in the dataset. Further improvement on the analysis will probably also result in higher number for phosphopeptides for DIA measurements. To the best of our knowledge, no large-scale DIA phosphoproteomics study has been published, thus the employed techniques show a feasible way to reproducibly acquire the phosphoproteome in a systems biology study.

5.2. Increased variability due to phosphopeptide enrichment

We compared the variability among different SWATH-MS data and LFQ data which were obtained with two different computational analysis pipelines. The median CV for phosphopeptide enriched samples, quantified with the phospho-SWATH-MS pipeline was among biological samples around 0.33. For the aging dataset we calculated a median CV of 0.25. For the DDA acquired aging data the median CV values were in the same range. If this CV of the phosphopeptide enriched samples is compared to the median CV of the total proteome data of the aging experiment, which were among all samples 0.17, and within technical replicates 0.1, a large increase in variably can be observed. It is well known, that the phosphopeptide enrichment step increases the variability [42].

Several strategies are suggested to lower the variability. A spin-tip based enrichment in a single step, with Ti^{4+}-IMAC allows high parallelization of the enrichment reduces variation. With such an enrichment procedure a median CV of 0.2 within replicates has been achieved [85]. One has to mention, that in this study, Jurkat T lymphoma cells were used. A part of the variation within our samples originates from the fact,

that the mouse liver tissue is not as homogeneous as cells grown in a plate. Nevertheless a downscaling to make the usage of a spin-tip based enrichment would be a way to try to remove the variation among replicates. Another study, conducted with the EasyPhos enrichment protocol, which allows the parallelization of 96 phosphopeptide enrichment procedures achieves a mean CV between 0.2 and 0.25 [86]. It seems, that automation and parallelization of our workflow would probably help to decrease the variability, origin from the phosphopeptide enrichment procedure.

In order to introduce a higher automation, the amount of starting material must be downscaled to 250 µg of peptides to make tip-based phosphopeptide enrichment feasible [85]. By doing so, another source of variability can be removed, by conducting the lysis with pressure cycling technology (PCT). For the currently used protocol we used the conventional lysis with the glass dounce homogenizer as for large amounts of tissue, PCT lysis leads to a multiple increase of bench working time. However, reducing the amount of starting material will result in lower quantified phosphopeptides.

5.3. PTM-SWATH assay library

From a data analysis perspective, the establishment and subsequent assessment of different SWATH assay libraries which contain phosphopeptide assays for mouse liver tissue was a key factor for the establishment of the phospho-SWATH-MS workflow described in this thesis. Analysis of phosphopeptide enriched SWATH-MS data can be achieved either by constructing a sample specific library or using, an already existing library. As we aimed to study the phosphoproteome in liver, we used the approach to construct a sample specific library. It should also be mentioned, that to our knowledge no SWATH assay libraries for phosphopeptide samples of any mouse tissue are published. Recently developments in the field of PTM SWATH were made by the developed extension SWATHProphet^{PTM} or the unpublished OpenSWATH/PTM extension [87] (and George A. Rosenberger, unpublished). However, we decided to quantify the data with the library, for which the LuciPHOr2 site localization scoring tool was used, as the OpenSWATH/PTM extension is still being developed. The specific mouse liver tissue library consisted of 2859 phosphopeptide from 1253 phosphoproteins. Despite that size of the library is rather small, this library enables us to quantify the phosphoproteome in mouse liver tissue very reproducible and lead to high-quality data. Compared to repository assay library like the pan-human library, which consist of transition assays for roughly 10’000 proteins, the mouse specific library is very small [88]. A sample specific library has the advantage that the control of false positives is easier feasible.

Although the library used for quantification performed well, further improvements considering the library, should be considered, as we are far away from a comprehensive sampling of the phosphoproteome. By adding further DDA measurements of the samples, the amount of assays within the library could be increased. Also a less stringent FLR threshold would lead to more phosphopeptides in the library.

5.4. Aging in mouse liver tissue

Aging is characterized by a progressive decline in physiological function and an increase propensity to degenerative diseases. With aging the risk to suffer from complex diseases, including nervous, immune, cardio-vascular and metabolic system increases. Thus, revealing the changes of the protein and phosphoprotein in young and old mouse liver tissue allows us to further investigate the functional und molecular mechanisms behind aging. In the aging experiment, we were able to characterize 101 significantly differently regulated proteins and phosphoprotein, between young and old mouse liver tissue samples. In both the total proteome and phosphoproteome, 3.4 % of all measurements were found to be significantly altered due to aging after multiple testing correction. This low amount of significantly altered proteins was also found in another study of aging in mouse liver tissue, were less than 1 % of the proteins were significantly altered [89].

Several proteins involved in metabolic pathways were altered in old mouse liver tissue samples. The altered pathways were consistent with the found metabolic changes in a previous study of the same mouse liver tissue samples [21]. The alterations in the xenobiotic metabolizing enzyme and transporters can be explained as reaction to increased levels of ROS in old liver tissue. The pathways are upregulated to detoxify the cells from truncated proteins and shuttle out waste products through the upregulation of carriers. We also found glutathione S-transferases (GST) to be upregulated in old mouse liver tissue. The GSTs are involved in the redox homeostasis and detoxification process. We were not able to confirm with significantly regulated phosphoproteins, the stated alterations of the oxidative phosphorylation and fatty acid oxidation pathway found in the previous study.

Over all this small pilot study of 2 young and 2 old moue liver tissue samples has already lead to promising results, and has verified parts of the previously observed alteration on the metabolome and transcript level. It seems promising that with a larger dataset, additional differently regulated proteins and altered pathways can be identified.

5.5. Phosphoprotein-QTLs in a mouse reference population

In this study, we quantified 969 phosphoproteins in livers of 76 BXD mice of the BXD mouse genetic reference panel. As environmental factor the 40 different strains were fed with two different diets. We were able to link 63 of phosphoprotein-QTLs with the distinct phenotypes. It seems, this is a relatively low number of identified pQTLs compared to a study for which a multi-omics approach was used. For the study, the transcript and proteome levels of 192 metabolism genes were combined, and with the dataset it was possible to reveal around 50 significantly regulated pQTLs in the same mouse liver tissue samples [18]. It is assumed, that the phosphoproteomics layer did not discover more pQTLs due to higher within-strain variability among the large-scale experiment. It seems, that with more precise measurements more differently regulated phosphoproteins would be detected and thereby increasing the number of pQTLs. Another issue, is that we used only a subset of the most promising phosphoproteins, which show a Pearson's correlation coefficient more or equal to 0.5, between the two diets and between all strains, for the mapping to the genome. Thus, a mapping with all data would maybe lead to discovery of more phospho-pQTLs. Another criteria could be, that we set a relative harsh filter in the SWATH2stats step of the analysis, as we only considered the phosphopeptides, which were detected at in least 60 % of the samples. Also promising seems the further enhancement of phospho-SWATH-MS library and the reanalyzing of the data to increase the number of phosphopeptide in the data. The described dataset is the first systemic analysis of phospho-pQTLs for more than thousand phosphopeptides. Yet, no large-scale phospho-pQTL study was published. To our knowledge, only specific loci, which contain kinases or phosphoproteins were analyzed in studies [90], [91], [92].

Potential validation strategy for identified phosphoprotein-QTLs

The results of the large-scale phosphoprotein-QTL analysis indicate, that this omics-layer can help to further characterize the BXD mouse reference population. However, the thesis did not provide any validation of the discovered phospho-pQTLs. The ongoing analysis is focused on the study of the molecular and mechanistic pathways, which are affected due to the different genotypes within the BXD mouse reference population. Another important part is the validation of the identified phosphoprotein-QTLs, which is a quite tricky and not easy to achieve task.

In a first step, one should aim to validate that the change in abundances on the phosphoprotein level is due to phosphorylation and not due to a changes in the total proteome. This can be achieved, by controlling if the same protein respectively peptide is detected in the total proteome and was also found there to be differently regulated due to genetics. If so, one should check, if it was possible to identify eQTLs

and pQTLs, within the QTL analysis of the total proteome, as this provides strong evidence, that the protein is upregulated and therefore we detected a higher phosphoprotein signal [19].

Secondly, the trans-phospho-pQTLs should be analyzed by checking if the genomic loci harbors obvious gene sequences of kinases. To conduct such an analysis, the data need to be searched for known kinase-substrate interactions by using the differently regulated phosphoproteins as substrate. If there is a perfect match, and the kinase can be found in the trans-phosphoprotein-QTL, it provides a good hint, that the phosphorylation site is directly regulated by this kinase.

Third, for a few of the most interesting phosphorylation sites which were identified as highly potential regulated through genetics, and the corresponding kinases could be identified, these kinases should be experimentally validated. As, this would consume a lot of time to knock-down the kinases in mouse samples, the validations should be conducted in mouse liver tissue derived cell lines.

To summarize, the phosphoprotein-QTLs discovered in the phosphoprotein-QTL study still need to be validated by taken into account already assigned pQTLs and data of the total proteome data, identify and validate kinase-substrate interactions with bioinformatics tools, and further conduct a few experimental validations in mouse liver derived cell line by knocking out the kinases. However, we have identified for the first time phospho-pQTLs in large numbers, as to our knowledge, until now only a few studies have found single phospho-pQTLs. Therefore, the used technique provides an approach to add another important omics-layer, for more comprehensive characterizations of genetic reference populations.

5.6. Implications for further research

Our data show that the integration of systems phosphoproteomics data sets provides another important layer, to study the mechanistic regulation of complex systems. The study of phosphoprotein-QTLs should help to understand specific regulation mechanisms, as phosphorylation is one of the key functions how cells regulate important basic process, including metabolism, growth, cell division and differentiation, organelle trafficking and immune responses [29]. A promising strategy to further elucidate the fine and dynamic regulation through phosphorylation can maybe be achieved by integrating the phosphoproteome with other omics-layers of the biological system as described [18]. This indicates that a better understanding of how the phosphoproteome – as major regulatory process in mammals – is influenced by environmental and genotypic factors, will be essential to fully understand complex

diseases, improve diagnostics by identifying biomarkers, and find well-tailored treatments.

6. References

1. Altshuler, D., M.J. Daly, and E.S. Lander, *Genetic mapping in human disease.* Science, 2008. **322**(5903): p. 881-8.
2. Bailey-Wilson, J.E. and A.F. Wilson, *Linkage analysis in the next-generation sequencing era.* Hum Hered, 2011. **72**(4): p. 228-36.
3. Mackay, T.F., E.A. Stone, and J.F. Ayroles, *The genetics of quantitative traits: challenges and prospects.* Nat Rev Genet, 2009. **10**(8): p. 565-77.
4. Lander, E.S. and D. Botstein, *Mapping mendelian factors underlying quantitative traits using RFLP linkage maps.* Genetics, 1989. **121**(1): p. 185-99.
5. Donnelly, P., *Progress and challenges in genome-wide association studies in humans.* Nature, 2008. **456**(7223): p. 728-31.
6. Hunter, D.J., *Gene-environment interactions in human diseases.* Nat Rev Genet, 2005. **6**(4): p. 287-98.
7. Dempfle, A., et al., *Gene-environment interactions for complex traits: definitions, methodological requirements and challenges.* Eur J Hum Genet, 2008. **16**(10): p. 1164-72.
8. Pericak-Vance, M.A., et al., *Linkage studies in familial Alzheimer disease: evidence for chromosome 19 linkage.* Am J Hum Genet, 1991. **48**(6): p. 1034-50.
9. Jeunemaitre, X., et al., *Molecular basis of human hypertension: role of angiotensinogen.* Cell, 1992. **71**(1): p. 169-80.
10. Manolio, T.A., *Genomewide association studies and assessment of the risk of disease.* N Engl J Med, 2010. **363**(2): p. 166-76.
11. Ott, J., J. Wang, and S.M. Leal, *Genetic linkage analysis in the age of whole-genome sequencing.* Nat Rev Genet, 2015. **16**(5): p. 275-84.
12. Doerge, R.W., *Mapping and analysis of quantitative trait loci in experimental populations.* Nat Rev Genet, 2002. **3**(1): p. 43-52.
13. Peirce, J.L., et al., *A new set of BXD recombinant inbred lines from advanced intercross populations in mice.* BMC Genet, 2004. **5**: p. 7.
14. Potter, M. and B. Mock, *Mouse Genetics - Concepts and Applications - Silver,Lm.* Science, 1995. **270**(5242): p. 1692-1693.
15. Gatti, D., et al., *Genome-level analysis of genetic regulation of liver gene expression networks.* Hepatology, 2007. **46**(2): p. 548-57.
16. Williams, E.G. and J. Auwerx, *The Convergence of Systems and Reductionist Approaches in Complex Trait Analysis.* Cell, 2015. **162**(1): p. 23-32.
17. Andreux, P.A., et al., *Systems genetics of metabolism: the use of the BXD murine reference panel for multiscalar integration of traits.* Cell, 2012. **150**(6): p. 1287-99.
18. Wu, Y., et al., *Multilayered genetic and omics dissection of mitochondrial activity in a mouse reference population.* Cell, 2014. **158**(6): p. 1415-30.
19. Williams, E.G., and Wu, Y., et al., *Systems proteomics and trans-omic data integration illuminate new mechanisms in mitochondrial function.* Science, 2016.
20. Hayflick, L., *How and why we age.* Experimental Gerontology, 1998. **33**(7-8): p. 639-653.
21. Houtkooper, R.H., et al., *The metabolic footprint of aging in mice.* Sci Rep, 2011. **1**: p. 134.
22. Cui, H., Y. Kong, and H. Zhang, *Oxidative stress, mitochondrial dysfunction, and aging.* J Signal Transduct, 2012. **2012**: p. 646354.
23. Aebersold, R. and M. Mann, *Mass spectrometry-based proteomics.* Nature, 2003. **422**(6928): p. 198-207.
24. Zhang, G., et al., *Overview of peptide and protein analysis by mass spectrometry.* Curr Protoc Protein Sci, 2010. **Chapter 16**: p. Unit16 1.

25. Yates, J.R., C.I. Ruse, and A. Nakorchevsky, *Proteomics by Mass Spectrometry: Approaches, Advances, and Applications.* Annual Review of Biomedical Engineering, 2009. **11**: p. 49-79.
26. Banerjee, S. and S. Mazumdar, *Electrospray Ionization Mass Spectrometry: A Technique to Access the Information beyond the Molecular Weight of the Analyte.* International Journal of Analytical Chemistry, 2012.
27. Eliuk, S. and A. Makarov, *Evolution of Orbitrap Mass Spectrometry Instrumentation.* Annual Review of Analytical Chemistry, Vol 8, 2015. **8**: p. 61-80.
28. Olsen, J.V., et al., *A Dual Pressure Linear Ion Trap Orbitrap Instrument with Very High Sequencing Speed.* Molecular & Cellular Proteomics, 2009. **8**(12): p. 2759-2769.
29. Domon, B. and R. Aebersold, *Options and considerations when selecting a quantitative proteomics strategy.* Nature Biotechnology, 2010. **28**(7): p. 710-721.
30. Heaven, M.R., et al., *Systematic evaluation of data-independent acquisition for sensitive and reproducible proteomics-a prototype design for a single injection assay.* J Mass Spectrom, 2016. **51**(1): p. ii.
31. Picotti, P., R. Aebersold, and B. Domon, *The implications of proteolytic background for shotgun proteomics.* Mol Cell Proteomics, 2007. **6**(9): p. 1589-98.
32. Picotti, P., et al., *High-throughput generation of selected reaction-monitoring assays for proteins and proteomes.* Nat Methods, 2010. **7**(1): p. 43-6.
33. Gillet, L.C., et al., *Targeted data extraction of the MS/MS spectra generated by data-independent acquisition: a new concept for consistent and accurate proteome analysis.* Mol Cell Proteomics, 2012. **11**(6): p. O111 016717.
34. Venable, J.D., et al., *Automated approach for quantitative analysis of complex peptide mixtures from tandem mass spectra.* Nat Methods, 2004. **1**(1): p. 39-45.
35. Selevsek, N., et al., *Reproducible and consistent quantification of the Saccharomyces cerevisiae proteome by SWATH-mass spectrometry.* Mol Cell Proteomics, 2015. **14**(3): p. 739-49.
36. Rost, H.L., et al., *OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data.* Nature Biotechnology, 2014. **32**(3): p. 219-223.
37. de la Fuente van Bentem, S., et al., *Towards functional phosphoproteomics by mapping differential phosphorylation events in signaling networks.* Proteomics, 2008. **8**(21): p. 4453-65.
38. Ubersax, J.A. and J.E. Ferrell, Jr., *Mechanisms of specificity in protein phosphorylation.* Nat Rev Mol Cell Biol, 2007. **8**(7): p. 530-41.
39. Engholm-Keller, K. and M.R. Larsen, *Technologies and challenges in large-scale phosphoproteomics.* Proteomics, 2013. **13**(6): p. 910-31.
40. Zhou, H., et al., *Specific phosphopeptide enrichment with immobilized titanium ion affinity chromatography adsorbent for phosphoproteome analysis.* J Proteome Res, 2008. **7**(9): p. 3957-67.
41. Zhou, H., et al., *Robust phosphoproteome enrichment using monodisperse microsphere-based immobilized titanium (IV) ion affinity chromatography.* Nat Protoc, 2013. **8**(3): p. 461-80.
42. Riley, N.M. and J.J. Coon, *Phosphoproteomics in the Age of Rapid and Deep Proteome Profiling.* Anal Chem, 2016. **88**(1): p. 74-94.
43. Villen, J., et al., *Large-scale phosphorylation analysis of mouse liver.* Proc Natl Acad Sci U S A, 2007. **104**(5): p. 1488-93.
44. Fermin, D., et al., *LuciPHOr2: site localization of generic post-translational modifications from tandem mass spectrometry data.* Bioinformatics, 2015. **31**(7): p. 1141-1143.
45. Monetti, M., et al., *Large-scale phosphosite quantification in tissues by a spike-in SILAC method.* Nat Methods, 2011. **8**(8): p. 655-8.

46. Wilson-Grady, J.T., W. Haas, and S.P. Gygi, *Quantitative comparison of the fasted and re-fed mouse liver phosphoproteomes using lower pH reductive dimethylation.* Methods, 2013. **61**(3): p. 277-86.
47. Pan, C., et al., *Quantitative phosphoproteome analysis of a mouse liver cell line reveals specificity of phosphatase inhibitors.* Proteomics, 2008. **8**(21): p. 4534-46.
48. Guo, T., et al., *Rapid mass spectrometric conversion of tissue biopsy samples into permanent quantitative digital proteome maps.* Nat Med, 2015. **21**(4): p. 407-13.
49. Resynbio, *MagReSyn® Ti-IMAC.* 2012-15.
50. Kessner, D., et al., *ProteoWizard: open source software for rapid proteomics tools development.* Bioinformatics, 2008. **24**(21): p. 2534-6.
51. Bauch, A., et al., *openBIS: a flexible framework for managing and analyzing complex data in biology research.* BMC Bioinformatics, 2011. **12**: p. 468.
52. Deutsch, E.W., et al., *A guided tour of the Trans-Proteomic Pipeline.* Proteomics, 2010. **10**(6): p. 1150-1159.
53. Kunszt, P., et al., *iPortal: the swiss grid proteomics portal: Requirements and new features based on experience and usability considerations.* Concurrency and Computation-Practice & Experience, 2015. **27**(2): p. 433-445.
54. Geer, L.Y., et al., *Open mass spectrometry search algorithm.* Journal of Proteome Research, 2004. **3**(5): p. 958-964.
55. MacLean, B., et al., *General framework for developing and evaluating database scoring algorithms using the TANDEM search engine.* Bioinformatics, 2006. **22**(22): p. 2830-2832.
56. Eng, J.K., T.A. Jahan, and M.R. Hoopmann, *Comet: An open-source MS/MS sequence database search tool.* Proteomics, 2013. **13**(1): p. 22-24.
57. Weisser, H., et al., *An Automated Pipeline for High-Throughput Label-Free Quantitative Proteomics.* Journal of Proteome Research, 2013. **12**(4): p. 1628-1644.
58. Cox, J. and M. Mann, *MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification.* Nat Biotechnol, 2008. **26**(12): p. 1367-72.
59. Cox, J., et al., *Andromeda: a peptide search engine integrated into the MaxQuant environment.* J Proteome Res, 2011. **10**(4): p. 1794-805.
60. Cox, J., et al., *A practical guide to the MaxQuant computational platform for SILAC-based quantitative proteomics.* Nat Protoc, 2009. **4**(5): p. 698-705.
61. Schubert, O.T., et al., *Building high-quality assay libraries for targeted analysis of SWATH MS data.* Nature Protocols, 2015. **10**(3).
62. Fermin, D., et al., *LuciPHOr: algorithm for phosphorylation site localization with false localization rate estimation using modified target-decoy approach.* Mol Cell Proteomics, 2013. **12**(11): p. 3409-19.
63. Lam, H., et al., *Development and validation of a spectral library searching method for peptide identification from MS/MS.* Proteomics, 2007. **7**(5): p. 655-667.
64. Teleman, J., et al., *DIANA-algorithmic improvements for analysis of data-independent acquisition MS data.* Bioinformatics, 2015. **31**(4): p. 555-562.
65. Reiter, L., et al., *mProphet: automated data processing and statistical validation for large-scale SRM experiments.* Nat Methods, 2011. **8**(5): p. 430-5.
66. Blattmann, P., M. Heusel, and R. Aebersold, *SWATH2stats: An R/Bioconductor Package to Process and Convert Quantitative SWATH-MS Proteomics Data for Downstream Analysis Tools.* PLoS One, 2016. **11**(4): p. e0153160.
67. Teo, G.S., et al., *mapDIA: Preprocessing and statistical analysis of quantitative proteomics data from data independent acquisition mass spectrometry.* Journal of Proteomics, 2015. **129**: p. 108-120.

68. Szklarczyk, D., et al., *STRING v10: protein-protein interaction networks, integrated over the tree of life.* Nucleic Acids Res, 2015. **43**(Database issue): p. D447-52.
69. Janga, S.C., J.J. Diaz-Mejia, and G. Moreno-Hagelsieb, *Network-based function prediction and interactomics: the case for metabolic enzymes.* Metab Eng, 2011. **13**(1): p. 1-10.
70. Broman, K.W., et al., *R/qtl: QTL mapping in experimental crosses.* Bioinformatics, 2003. **19**(7): p. 889-90.
71. Mouse Genome Sequencing, C., et al., *Initial sequencing and comparative analysis of the mouse genome.* Nature, 2002. **420**(6915): p. 520-62.
72. Wang, X.S., et al., *High-throughput sequencing of the DBA/2J mouse genome.* Bmc Bioinformatics, 2010. **11**.
73. Haley, C.S. and S.A. Knott, *A simple regression method for mapping quantitative trait loci in line crosses using flanking markers.* Heredity (Edinb), 1992. **69**(4): p. 315-24.
74. Kruglyak, L. and E.S. Lander, *A Nonparametric Approach for Mapping Quantitative Trait Loci.* Genetics, 1995. **139**(3): p. 1421-1428.
75. Li, Q.R., et al., *Effect of peptide-to-TiO2 beads ratio on phosphopeptide enrichment selectivity.* J Proteome Res, 2009. **8**(11): p. 5375-81.
76. Benjamini, Y. and Y. Hochberg, *Controlling the False Discovery Rate - a Practical and Powerful Approach to Multiple Testing.* Journal of the Royal Statistical Society Series B-Methodological, 1995. **57**(1): p. 289-300.
77. Ashburner, M., et al., *Gene Ontology: tool for the unification of biology.* Nature Genetics, 2000. **25**(1): p. 25-29.
78. Blake, J.A., et al., *Gene Ontology Consortium: going forward.* Nucleic Acids Research, 2015. **43**(D1): p. D1049-D1056.
79. Lee, J.S., et al., *Hepatic xenobiotic metabolizing enzyme and transporter gene expression through the life stages of the mouse.* PLoS One, 2011. **6**(9): p. e24381.
80. Francis, G.A., et al., *Nuclear receptors and the control of metabolism.* Annu Rev Physiol, 2003. **65**: p. 261-311.
81. Kanehisa, M., et al., *KEGG as a reference resource for gene and protein annotation.* Nucleic Acids Research, 2016. **44**(D1): p. D457-D462.
82. Kanehisa, M. and S. Goto, *KEGG: kyoto encyclopedia of genes and genomes.* Nucleic Acids Res, 2000. **28**(1): p. 27-30.
83. Modica, S., E. Bellafante, and A. Moschetta, *Master regulation of bile acid and xenobiotic metabolism via the FXR, PXR and CAR trio.* Front Biosci (Landmark Ed), 2009. **14**: p. 4719-45.
84. Huttlin, E.L., et al., *A tissue-specific atlas of mouse protein phosphorylation and expression.* Cell, 2010. **143**(7): p. 1174-89.
85. de Graaf, E.L., et al., *Single-step enrichment by Ti4+-IMAC and label-free quantitation enables in-depth monitoring of phosphorylation dynamics with high reproducibility and temporal resolution.* Mol Cell Proteomics, 2014. **13**(9): p. 2426-34.
86. Humphrey, S.J., S.B. Azimifar, and M. Mann, *High-throughput phosphoproteomics reveals in vivo insulin signaling dynamics.* Nat Biotechnol, 2015. **33**(9): p. 990-5.
87. Keller, A., et al., *Opening a SWATH Window on Posttranslational Modifications: Automated Pursuit of Modified Peptides.* Mol Cell Proteomics, 2016. **15**(3): p. 1151-63.
88. Rosenberger, G., et al., *A repository of assays to quantify 10,000 human proteins by SWATH-MS.* Sci Data, 2014. **1**: p. 140031.
89. Walther, D.M. and M. Mann, *Accurate quantification of more than 4000 mouse tissue proteins reveals minimal proteome changes during aging.* Mol Cell Proteomics, 2011. **10**(2): p. M110 004523.
90. Hernandez, S.C., et al., *A genome-wide linkage analysis for reproductive traits in F2 Large White x Meishan cross gilts.* Anim Genet, 2014. **45**(2): p. 191-7.

91. Qi, P., et al., *The novel quantitative trait locus GL3.1 controls rice grain size and yield by regulating Cyclin-T1;3.* Cell Res, 2012. **22**(12): p. 1666-80.
92. Brich, J., et al., *Genetic modulation of tau phosphorylation in the mouse.* J Neurosci, 2003. **23**(1): p. 187-92.

6.1. List of Abbreviations

Table 15: Summary of abbreviations

ABC	Ammonium bicarbonate
BCA	Pierce™ Bicinchoninic Acid Assay
BH	Benjamini Hochberg
BSA	Bovine Serum Albumin
CD	Chow Diet
CID	Collision-induced Dissociation
CON	Conventional lysis with glass dounce homogenizer
CoOmXT	Comet, Omssa and X!Tandem
CV	Coefficient of Variation
CYP	Cytochrome P450
DDA	Data dependent acquisition
DIA	Data independent acquisition
eQTL	Expression Quantitative Trait Locus
ESI	Electrospray Ionization
FC	Fold Change
FDR	False Discovery Rate
FLR	False Localization Rate
GO	Gen Ontology
GST	Glutathione S transferases
GWAS	Genome-wide Association Study
HCD	Higher-energy Collisional Dissociation
HFD	High Fat Diet
HPLC	High Pressure Liquid Chromatography
HPLC	High-Performance Liquid Chromatography
IAA	Iodoacetamide
IEX	Ion Exchange
IMAC	Immobilized Metal Ion Affinity Chromatography
KEGG	Kyoto Encyclopedia of Genes and Genomes
LC	Liquid Chromatography
LFQ	Label Free Quantification
Log2FC	Log base 2 Fold Change
LTQ	Linear Ion Trap Mass Spectrometer
Lys-C	Lysyl Endopeptidase®
MRM	Multiple Reaction Monitoring
MS	Mass Spectrometry
OmXT	Omssa and X!Tandem
PCT	Pressure Cycling Technology
phospho-pQTL	Phosphoprotein Quantitative Trait Locus
PPI	Protein-Protein Interaction
pQTL	Protein Quantitative Trait Locus
PTMs	Post Transcriptional Modifications
QQQ	Triple Quadrupole
Q1	Quadrupole 1
q2	Quadrupole 2
RI or RIS	Recombinant Inbred Strain
RIL	Recombinant Inbred Line
ROS	Reactive oxygen species
SEC	Size Exclusion Chromatography
SNPs	Single Nucleotide Polymorphisms
SRM	Selected Reaction Monitoring
SWATH	Sequential Window Acquisition of all Theoretical fragment ion spectra
TCEP	Tris(2-carboxyethyl)phosphine

TFA	Trifluoroacetic Aacid
ToF	Time of Flight mass analyzer
ToF-MS	Time of Flight mass spectrometer
TTP	Trans Proteomic Pipeline
XMETs	Xenobiotic Metabolizing Enzymes and Transporters
m/z	Mass To Charge Ratio
ppm	Parts Per Million
Glu-Fib	Glu-1-Fibrinopeptide B
mM	Milli-Molar
M	Molar
nm	Nanometer
H	Hours
rpm	Rounds Per Minute
g	G-Force
µg	Microgram
µL	Microliter
mg	Millimeter
µm	Micrometer
mL	Milliliter
MS1	Precursor Ion Scan
MS2	Product Ion Scan
Ms	Millisecond

6.2. Index of Figures

6.3. Index of Tables

7. Appendix

7.1. Supplementary tables

7.1.1. List of all mouse liver tissue samples

Table 16: Weighted pieces of mouse liver tissue of C57-1 for the "beads and buffer combinations" experiment. *The samples were done in triplicates for CON and PCT lysis. The portions for PCT were smaller as approximately half of the tissue was sufficient to get enough lysed protein. For the CON lysis spared lysed protein solution was stored at -20 °C as a backup.*

	Lysis replicate	Lysis method	Technical Replicate	Weighted tissue [mg]
Lysis_1	1	CON	1	50
Lysis_2	2	CON	1	50
Lysis_3	3	CON	2	50
Lysis_4	4	CON	2	50
Lysis_5	5	PCT	1	24.8
Lysis_6	6	PCT	2	24.6
Lysis_7	7	CON	3	50
Lysis_8	8	CON	3	50
Lysis_9	9	PCT	3	46.6

Table 17: Statistical data for the amount of cut mouse liver tissue of the aging experiment CON lysis.

	Mass [mg]
Average	46.37
Standard Deviation	7.99
Minimum	30.01
25 % Quartile	45.04
Median	49.82
75 % Quartile	51.19
Maximum	53.60

Table 18: Samples used for the BXD mouse genetic reference population with indication of the two diverse diet. *For 37 BXD mouse strains, we obtained two mice samples, fed with CD and HFD. For four sample the collaboration partner (Auwerx laboratory, IPFL Genova), we obtained only one condition.*

Strain	HFD	MS-Sample	CD	MS-Sample
DBA	DBA_HFD_136	FF243	DBA_CD_132	FF192
C57	C57_HFD_126	FF251	C57_CD_122	FF239
BXD99	BXD99_HFD_346	FF256	BXD99_CD_341	FF208
BXD98	BXD98_HFD_416	FF237	BXD98_CD_426	FF191
BXD97	BXD97_HFD_436	FF220	BXD97_CD_431	FF240
BXD96	BXD96_HFD_146	FF248	BXD96_CD_141	FF228
BXD95	BXD95_HFD_246	FF217	BXD95_CD_241	FF257
BXD92a			BXD92a_CD_231	FF194
BXD90	BXD90_HFD_206	FF258	BXD90_CD_201	FF227
BXD89			BXD89_CD_211	FF187
BXD87	BXD87_HFD_96	FF230	BXD87_CD_91	FF199

BXD85	BXD85_HFD_797	FF242	BXD85_CD_371	FF223
BXD84	BXD84_HFD_276	FF250	BXD84_CD_733	FF221
BXD83	BXD83_HFD_166	FF218	BXD83_CD_161	FF215
BXD81	BXD81_HFD_326	FF233	BXD81_CD_321	FF231
BXD80	BXD80_HFD_116	FF201	BXD80_CD_111	FF236
BXD79	BXD79_HFD_445	FF189	BXD79_CD_440	FF219
BXD75	BXD75_HFD_76	FF205	BXD75_CD_71	FF260
BXD73	BXD73_HFD_46	FF259	BXD73_CD_41	FF241
BXD71	BXD71_HFD_366	FF252	BXD71_CD_361	FF204
BXD70	BXD70_HFD_67	FF212	BXD70_CD_61	FF253
BXD69	BXD69_HFD_256	FF245	BXD69_CD_251	FF224
BXD68	BXD68_HFD_336	FF235	BXD68_CD_331	FF202
BXD66	BXD66_HFD_106	FF209	BXD66_CD_101	FF226
BXD64	BXD64_HFD_311	FF211	BXD64_CD_316	FF238
BXD62	BXD62_HFD_196	FF254	BXD62_CD_191	FF203
BXD61	BXD61_HFD_81	FF222	BXD61_CD_86	FF196
BXD60			BXD60_CD_282	FF186
BXD56	BXD56_HFD_406	FF197	BXD56_CD_401	FF249
BXD55	BXD55_HFD_56	FF225	BXD55_CD_51	FF207
BXD51	BXD51_HFD_26	FF190	BXD51_CD_22	FF261
BXD50	BXD50_HFD_451	FF216	BXD50_CD_421	FF193
BXD49	BXD49_HFD_307	FF214	BXD49_CD_301	FF246
BXD48	BXD48_HFD_296	FF255	BXD48_CD_291	FF244
BXD45	BXD45_HFD_36	FF188		
BXD442	BXD442_HFD_172	FF206	BXD442_CD_176	FF195
BXD43	BXD43_HFD_228	FF229	BXD43_CD_221	FF198
BXD103	BXD103_HFD_885	FF234	BXD103_CD_882	FF232
BXD101	BXD101_HFD_456	FF213	BXD101_CD_391	FF200
BXD100	BXD100_HFD_156	FF247	BXD100_CD_152	FF210

Table 19: Statistical data for the amount of cut mouse liver tissue of the mouse reference population experiment lysed by conventional lysis.

	Mass [mg]
Average	49.25
Standard Deviation	4.03
Minimum	34.80
25 % Quartile	47.93
Median	49.95
75 % Quartile	51.63
Maximum	58.00

Table 20: Obtained lysis efficiency for all mouse liver tissue samples. *In the table for each mouse, the number of the lysis batch, the weighted portion, the by BCA measurement obtained concentration, the volume gained by pipetting from the glass tube of the CON lysis, the total amount of lysed protein and the lysis efficiency is shown. The total lysed protein was calculated by multiplying the concentration*

of the lysis with the lysis volume. The lysis efficiency was calculated by dividing the total lysed protein amount through the weighted portion. For the phosphopeptide 1.5 mg of protein were used per sample.

Sample_ID	Lysis batch	weighted portion [mg]	concentration [mg mL^{-1}]	Lysis Volume [mL]	Total lysed protein [mg]	Lysis efficiency [%]
BXD60_CD_282	a1_1	53.7	1.116	5.52	6.16	11.47
BXD89_CD_211	a1_2	50.8	1.372	5.52	7.57	14.91
BXD45_HFD_36	a1_3	57	1.129	5.58	6.30	11.05
BXD79_HFD_445	a1_4	48.2	0.882	5.54	4.89	10.14
BXD51_HFD_26	a1_5	51	0.827	5.52	4.57	8.95
BXD98_CD_426	a1_6	49.8	1.226	5.5	6.74	13.54
DBA_CD_132	a1_7	52.3	1.194	5.49	6.56	12.53
BXD50_CD_421	a1_8	53.1	1.282	5.58	7.15	13.47
BXD90_HFD_206	a10_1	46.5	1.499	5.53	8.29	17.83
BXD73_HFD_46	a10_2	53.8	1.285	5.42	6.96	12.95
BXD75_CD_71	a10_3	48.1	1.18	5.38	6.35	13.20
BXD51_CD_22	a10_4	52.8	1.248	5.58	6.96	13.19
BXD92a_CD_231	a2_1	53.1	1.23	5.389	6.63	12.48
BXD442_CD_176	a2_2	37.5	1.053	5.338	5.62	14.99
BXD61_CD_86	a2_3	47.9	1.335	5.04	6.73	14.05
BXD56_HFD_406	a2_4	50.6	1.051	5.447	5.72	11.31
BXD43_CD_221	a2_5	50.1	1.654	5.348	8.85	17.66
BXD87_CD_91	a2_6	42.1	1.139	5.4	6.15	14.61
BXD101_CD_391	a2_7	48.9	1.073	5.13	5.50	11.26
BXD80_HFD_116	a2_8	48	1.146	5.405	6.19	12.90
BXD68_CD_331	a3_1	52.9	1.163	5.4	6.28	11.87
BXD62_CD_191	a3_2	52	1.537	5.37	8.25	15.87
BXD71_CD_361	a3_3	50	1.748	5.25	9.18	18.35
BXD75_HFD_76	a3_4	53.6	1.087	5.23	5.69	10.61
BXD442_HFD_172	a3_5	44	0.958	6.05	5.80	13.17
BXD442_HFD_172	a3_5b	42.5	0.951	5.41	5.14	12.11
BXD55_CD_51	a3_6	50.5	1.411	5.14	7.25	14.36
BXD99_CD_341	a3_7	53.7	1.425	5.07	7.22	13.45
BXD66_HFD_106	a3_8	34.8	0.748	5.21	3.90	11.20
BXD100_CD_152	a4_1	53.6	1.359	5.18	7.04	13.13
BXD64_HFD_311	a4_2	50.1	1.019	5.47	5.57	11.13
BXD70_HFD_67	a4_3	51.9	1.135	5.45	6.19	11.92
BXD101_HFD_456	a4_4	48.3	1.356	5.29	7.17	14.85
BXD49_HFD_307	a4_5	51.2	1.321	5.15	6.80	13.29
BXD83_CD_161	a4_6	45.5	1.594	5.32	8.48	18.64
BXD50_HFD_451	a4_7	51.4	1.48	5.22	7.73	15.03
BXD95_HFD_246	a4_8	50.4	1.357	5.35	7.26	14.40
BXD83_HFD_166	a5_1	51.4	1.063	5.36	5.70	11.08
BXD79_CD_440	a5_2b	47.3	0.987	5.4	5.33	11.27
BXD84_CD_733	a5_4	50.9	1.316	5.52	7.26	14.27
BXD61_HFD_81	a5_5b	58	0.934	5.4	5.04	8.70

BXD85_CD_371	a5_6b	47	1.142	5.39	6.16	13.10
BXD69_CD_251	a5_7	49.9	1.222	5.46	6.67	13.37
BXD55_HFD_56	a5_8	49.8	1.264	5.32	6.72	13.50
BXD66_CD_101	a6_1b	51	1.166	5.44	6.34	12.44
BXD90_CD_201	a6_2b	48.1	1.502	5.24	7.87	16.36
BXD96_CD_141	a6_3b	46.7	1.1	5.26	5.79	12.39
BXD43_HFD_228	a6_4b	48.3	1.197	5.25	6.28	13.01
BXD87_HFD_96	a6_5b	46	0.861	5.23	4.50	9.79
BXD81_CD_321	a6_6b	48.4	1.312	5.27	6.91	14.29
BXD103_CD_882	a6_7b	48.5	1.24	4.76	5.90	12.17
BXD81_HFD_326	a6_8b	52.1	1.053	5.33	5.61	10.77
BXD103_HFD_885	a7_1	49.9	1.069	5.31	5.68	11.38
BXD68_HFD_336	a7_2	52.6	1.359	5.29	7.19	13.67
BXD80_CD_111	a7_3	44.5	1.083	5.37	5.82	13.07
BXD98_HFD_416	a7_4	50	0.812	5.33	4.33	8.66
BXD64_CD_316	a7_5	49	1.265	5.38	6.81	13.89
C57_CD_122	a7_6	49.5	1.111	5.39	5.99	12.10
BXD97_CD_431	a7_7	50.4	0.978	5.54	5.42	10.75
BXD73_CD_41	a7_8	48.8	1.246	5.28	6.58	13.48
BXD85_HFD_797	a8_1	54.9	1.128	5.12	5.78	10.52
DBA_HFD_136	a8_2	56.6	1.339	5.3	7.10	12.54
BXD48_CD_291	a8_3	50.9	1.228	5.34	6.56	12.88
BXD69_HFD_256	a8_4	50.4	1.233	5.34	6.58	13.06
BXD49_CD_301	a8_5	45.7	1.15	5.35	6.15	13.46
BXD100_HFD_156	a8_6	50.4	1.039	5.2	5.40	10.72
BXD96_HFD_146	a8_7	41.1	0.465	5.35	2.49	6.05
BXD56_CD_401	a8_8	51.8	1.559	5.23	8.15	15.74
BXD84_HFD_276	a9_1	47.2	1.17	5.43	6.35	13.46
C57_HFD_126	a9_2	51.7	0.794	5.37	4.26	8.25
BXD71_HFD_366	a9_3	50.9	1.04	5.31	5.52	10.85
BXD70_CD_61	a9_4	47.5	1.011	5.4	5.46	11.49
BXD62_HFD_196	a9_5	41.4	0.957	5.28	5.05	12.21
BXD48_HFD_296	a9_6	48.2	0.859	5.19	4.46	9.25
BXD99_HFD_346	a9_7	52.3	0.669	5.41	3.62	6.92
BXD95_CD_241	a9_8	48.7	1.399	5.27	7.37	15.14

7.1.2. Parameters used for the SWATH-MS analysis with the openSWATH/PTM method

Copied from the internal wiki on the 10th of March 2016.

OpenSWATH/PTM Workflow

Use MS1 traces: Select true if you generated MS1-specific assays

Use UIS scores: Select true

UIS S/N threshold: -1
UIS peak area threshold: 0

PyProphet
main_var: xx_swath_prelim_score
vars: bseries_score elution_model_fit_score intensity_score
isotope_correlation_score isotope_overlap_score library_corr library_rmsd
log_sn_score massdev_score massdev_score_weighted norm_rt_score
xcorr_coelution xcorr_coelution_weighted xcorr_shape xcorr_shape_weighted
yseries_score
qvality.enable: True
qvality.epsilon-cross-validation: 0.0001
qvality.epsilon-step: 0.00000001
qvality.generalized: True
qvality.number-of-bins: 500
qvality.q-value: False
ms1_scoring.enable: MS1: True MS2: True
ms1.final_statistics.emp_p: MS1: True MS2: True
ms2_scoring.enable: True
ms2.final_statistics.emp_p: True
ms2_scoring.detection_fdr_ms1: 1.0
uis_scoring.enable: True
uis.final_statistics.emp_p: True
uis_scoring.detection_fdr_ms1: MS1: 0.05 MS2: 1.0
uis_scoring.detection_fdr_ms2: 0.05
uis_scoring.disable_h0: False
uis_scoring.identification_fdr: 1.0
uis_scoring.identification_probability: 0.4

FeatureAlignment
Realign method: lowess
clustering method: LocalMST
max RT diff: 30
Target FDR: -1
seeding m_score cutoff: 0.01
extension m_score cutoff: 0.1
Min. fraction for select.: 0
mst: useRTcorrection: True
mst:Stddev_multiplier: 3.0

Isotopic grouping: false

alignment_score 0.0001

Requant:

Do requantification: False

Isotopic transfer: False

7.1.3. Regulated proteins in the total cell lysate of the aging samples

Table 21: Proteins up- and downregulated in the total proteome SWATH-MS analysis. *Proteins were categorized as regulated if they had an adjusted p-value below 0.1 and an effect size of +/- 0.5 (log2 FC). For the analysis only the proteins which were identified with proteotypic peptides were taken into consideration. These proteins were used for PPI, and molecular and functional enrichment analysis.*

Protein	Regulation	effect size (log2 FC mean old)	p-adjusted
A2AJL3	Up	0.06	0.50
O08738	Up	0.06	0.52
O35660	Up	0.00	0.58
O70493	Up	0.05	0.77
O70570	Up	0.00	0.61
P01898	Up	0.05	0.82
P09528	Up	0.00	0.66
P10107	Up	0.04	1.35
P17563	Up	0.01	0.50
P19096	Up	0.01	0.60
P29391	Up	0.01	0.51
P31725	Up	0.06	0.98
P35235	Up	0.07	1.17
P35293	Up	0.05	0.54
P52840	Up	0.00	0.87
P53657	Up	0.00	0.50
P58044	Up	0.00	0.61
P62204	Up	0.03	0.62
Q05816	Up	0.02	0.72
Q3TCH7	Up	0.06	0.87
Q3U0B3	Up	0.05	0.69
Q3U4G3	Up	0.02	0.65
Q3URE1	Up	0.07	1.32
Q569Z5	Up	0.07	0.97
Q61133	Up	0.00	0.52
Q62264	Up	0.03	0.54
Q62468	Up	0.06	0.67
Q64471	Up	0.00	0.59
Q8BGR2	Up	0.05	2.01
Q8BHA3	Up	0.04	1.25
Q8BLN5	Up	0.04	0.82
Q8BTY8	Up	0.08	0.61

Q8CHR6	Up	0.00	0.63
Q8K009	Up	0.07	0.52
Q8VC97	Up	0.00	0.64
Q8VCX1	Up	0.00	0.69
Q920E5	Up	0.00	0.70
Q922F4	Up	0.09	0.54
Q99N42	Up	0.04	0.73
Q9CR86	Up	0.03	0.51
Q9CX00	Up	0.02	0.50
Q9CXF4	Up	0.00	0.61
Q9D110	Up	0.05	0.79
Q9D6Y9	Up	0.00	0.56
2/P13745/P10648	Up	0.00	0.72
2/P28650/P46664	Up	0.04	0.69
2/P49945/P29391	Up	0.00	0.78
2/P68510/P61982	Up	0.01	0.78
2/Q61115/Q9ET01	Up	0.00	0.59
2/Q80W2P15626	Up	0.00	0.63
2/Q8VHX6/Q8BTM8	Up	0.03	1.82
2/Q91Z98/O35744	Up	0.07	0.74
2/Q99LD8/Q9CWS0	Up	0.01	0.52
3/P30115/P13745 /P10648	Up	0.09	0.53
3/Q99020/Q60668 /Q9Z130	Up	0.10	0.92
4/Q9JK88/Q60854 /Q8VHP7/Q9D154	Up	0.06	0.52
O35728	Down	0.01	-0.70
O88833	Down	0.02	-0.72
O88962	Down	0.00	-0.51
P06728	Down	0.00	-1.18
P07759	Down	0.02	-0.85
P11589	Down	0.00	-0.66
P17717	Down	0.01	-0.58
P29758	Down	0.05	-0.65
P43276	Down	0.00	-0.84
P43883	Down	0.05	-0.50
Q01730	Down	0.04	-0.53
Q60870	Down	0.03	-1.25
Q60991	Down	0.05	-0.73
Q61081	Down	0.09	-0.73
Q61694	Down	0.03	-0.87
Q63836	Down	0.03	-1.74
Q64FW2	Down	0.01	-0.51
Q8K2Z4	Down	0.03	-0.87
Q91WG0	Down	0.00	-0.64
Q91WL5	Down	0.00	-1.20

Q99P30	Down	0.00	-0.73
Q9D1L9	Down	0.07	-1.01
Q9D1M7	Down	0.03	-0.66
Q9D3B1	Down	0.06	-1.30
Q9DBM2	Down	0.00	-0.94
Q9QXZ6	Down	0.00	-0.70
Q9R0H0	Down	0.00	-0.69
Q9WVM8	Down	0.01	-0.51
Q9Z239	Down	0.03	-0.76
2/O35728/O88833	Down	0.02	-0.79
2/O35728/Q91WL5	Down	0.01	-0.76
2/O55143/Q8R429	Down	0.05	-0.52
2/O88833/O35728	Down	0.00	-0.94
2/Q03734/P07759	Down	0.06	-0.79
2/Q91WC3/P41216	Down	0.01	-0.80
2/Q91WL5/O35728	Down	0.00	-1.20
2/Q91YY5/Q9QXZ6	Down	0.03	-0.64
3/P04938/P02762 /B5X0G2	Down	0.01	-0.86
3/Q00898/Q00897 /P22599	Down	0.07	-0.50
4/P04938/P02762 /P11589/P11588	Down	0.01	-1.48
4/P04938/P11589 /P11588/P02762	Down	0.03	-0.99
5/P04938/P02762 /P11589/B5X0G2 /P11588	Down	0.00	-1.26
5/P04938/P02762 /P11589/P11588 /B5X0G2	Down	0.00	-1.15
5/Q61694/Q61767 /P26150/P26149 /O35469	Down	0.08	-0.57
5/Q91WP6/Q03734 /P07759/Q5I2A0 /P29621	Down	0.02	-1.08
6/P04938/P02762 /P11591/P11589 /P11588/B5X0G2	Down	0.00	-1.29
6/Q61694/Q61767 /P26150/P26149 /P24815/O35469	Down	0.04	-0.70
6/Q64436/Q9WV27 /Q6PIC6/Q6PIE5 /Q8VDN2/Q9Z1W8	Down	0.02	-0.59

8/P61028/Q6PHN9 /Q9D1G1/P62821 /Q8K386/Q9DD03 /P61027/P55258	Down	0.05	-0.51

7.1.4. Regulated phosphopeptides in the phosphopeptide enriched aging samples

Table 22: Phosphoproteins up- and downregulated in the phosphopeptide enriched samples SWATH-MS analysis of the aging dataset. *Phosphoproteins were categorized as regulated if they had an adjusted p-value below 0.1 and an effect size of +/- 0.5 (log2 FC). As for some phosphoproteins several regulated phosphosites were in the dataset, only the unique phosphoproteins were taken into consideration for the PPI, and molecular and functional enrichment analysis.*

Proteins	Phosphopeptide	Regu-lation	p_ad-justed	mean old
O35071	YPPYTT(Phospho)PPR	Up	0.03	1.01
O54916	RT(Phospho)SSDHTNPTSPLLVKPSDLSEENK	Up	0.04	0.83
O54916	RTS(Phospho)SDHTNPTSPLLVKPSDLSEENK	Up	0.03	0.93
O54916	RTSS(Phospho)DHTNPTSPLLVKPSDLSEENK	Up	0.04	0.76
O54916	RTSSDHT(Phospho)NPTSPLLVKPSDLSEENK	Up	0.06	0.80
P04627	GGDGAPRGS(Phospho)PSPASVSSGR	Up	0.03	1.36
P26645	AEDGAAPSPS(Phospho)SETPK	Up	0.00	0.87
P26645	AEDGAAPSPSS(Phospho)ETPK	Up	0.00	0.81
P26645	AEDGAAPSPSSET(Phospho)PK	Up	0.00	0.85
P35492	MEHIPESRPLS(Phospho)PTAFSLESLR	Up	0.10	0.67
Q3UM45	HGGGGIVANLS(Phospho)EQSLK	Up	0.03	0.76
Q3UTJ2	S(Phospho)EPAVGPLR	Up	0.05	1.10
Q62318	S(Phospho)GEGEVSGLLRK	Up	0.08	1.11
Q62448	T(Phospho)QTPPLGQTPQLGLK	Up	0.06	0.73
Q62448	TQT(Phospho)PPLGQTPQLGLK	Up	0.04	0.96
Q6ZPJ0	TAPSS(Phospho)PLTSPSDTR	Up	0.03	0.74
Q7TQD2	AVSS(Phospho)PTVSR	Up	0.04	1.06
Q7TSH2	RQS(Phospho)STADAPEAQHEPGITITEWK	Up	0.03	0.67
Q8JZZ7	S(Phospho)MPNLGAGR	Up	0.03	0.73
Q8VDZ4	HPS(Phospho)YRSEPSLEPESFR	Up	0.03	1.02
Q91V92	TAS(Phospho)FSESRADEVAPAK	Up	0.00	0.83
Q91VC7	GPGGS(Phospho)PSGLQK	Up	0.09	0.81
Q91WG5	KVDSPFSSGS(Phospho)PSR	Up	0.08	0.58
Q9CR86	GNVVPS(Phospho)PLPTR	Up	0.03	0.87
Q9CR86	GNVVPS(Phospho)PLPTRR	Up	0.10	0.62
Q9CR86	TFS(Phospho)ATVR	Up	0.06	0.69
Q9D1L0	RAPAAQPPAAAAPSAVGS(Phospho)PAAAPR	Up	0.04	0.61
Q9DBR7	SAS(Phospho)YSYLEDR	Up	0.09	0.81
Q9DD18	SASS(Phospho)GAEGDVSSEREP	Up	0.04	0.55
Q9JJL3	NSAS(Phospho)LHVLK	Up	0.08	0.92
Q6ZQ58	S(Phospho)LPTTVPESPNYR	Up	0.05	0.62

Q80XQ2	SES(Phospho)MPVQLNK	Up	0.05	1.05
A6X919	S(Phospho)ARSSPPPLSGASEVDAGELGSER	Down	0.09	-0.59
F8VPU2	LGAPENSGIST(Phospho)LER	Down	0.06	-0.79
P14602	QLS(Phospho)SGVSEIR	Down	0.00	-1.89
P32020	THQVSAAPTS(Phospho)SAGDGFK	Down	0.10	-0.73
Q69ZX8	RFS(Phospho)SGGEEEDFDR	Down	0.03	-0.60

7.1.5. Potentially due to diet regulated phosphopeptides in the BXD mouse reference population

Table 23: Regulated proteins with the quantified phosphopeptides due to diet in the BXD mouse genetic reference population. *To be considered as regulated by diet, the phosphoproteins were filtered by an effect size of +/- 0.5 and an adjusted p-value below 0.01. For the calculation of the p-value a pairwise t-test between each mice strain, which were on two diverse diet. The adjusted p-value was corrected with the Benjamini Hochberg method.*

Proteins	Phosphopeptide	effect size	adjusted p-value
O70475	IPYT(Phospho)PGEIPK	-0.80	3E-07
O70475	RIPYT(Phospho)PGEIPK	-0.63	1E-06
O88343	NLTS(Phospho)SSLNDISDKPEKDQLK	0.79	8E-04
O88343	NLTSS(Phospho)SLNDISDKPEKDQLK	0.80	3E-03
O88343	NLTSSS(Phospho)LNDISDKPEK	0.57	6E-05
O88343	NLTSSSLNDIS(Phospho)DKPEK	0.51	3E-03
P0C673	GSS(Phospho)PQVLPR	0.71	2E-04
P15105	IPRT(Phospho)VGQEK	0.56	6E-05
P35492	MEHIPESRPLS(Phospho)PTAFSLESLR	-0.93	6E-03
P50136	IGHHSTSDDSS(Phospho)AYRSVDEVNYWDK	0.57	9E-03
P51660	VDSEGISPNRTS(Phospho)HAAPAATSGFVGAVGHK	0.64	4E-03
P54310	SVS(Phospho)EAALAQPEGLLGTDTLKK	0.52	9E-03
P58735	GGT(Phospho)LVLVR	-1.13	1E-06
P70429	SNS(Phospho)VEKPVSSLLSR	0.58	4E-04
Q01279	ELVEPLT(Phospho)PSGEAPNQAHLR	-1.13	5E-04
Q01279	ELVEPLTPS(Phospho)GEAPNQAHLR	-1.07	7E-04
Q60953	ALDES(Phospho)LAEPHLEDR	0.75	3E-03
Q62261	SALPAQSAAT(Phospho)LPAR	-0.58	5E-03
Q6TCG2	SHPASASAPRS(Phospho)PPAATTKPLLR	0.83	2E-03
Q8BHI7	GHQNGSVAAVNGHT(Phospho)NSFPSLENSVKPR	1.12	3E-03
Q8C0N2	NSAS(Phospho)VGIIQR	0.72	5E-04
Q8C5H8	ELAGGGS(Phospho)PADGGFRPSR	0.53	6E-05
Q8CC35	AAS(Phospho)PAKPSSLDLVPNLPR	0.62	4E-04
Q8R1G6	VLLHSPGRPS(Phospho)SPR	0.86	1E-05
Q921G7	NLS(Phospho)IYDGPEQR	0.82	6E-05
Q9DBS9	LHS(Phospho)SNPNLSTLDFGEEK	0.80	1E-03
Q9DBS9	LHSS(Phospho)NPNLSTLDFGEEK	0.84	7E-04
Q9DCM0	RLS(Phospho)QQSASGAPVLLR	-1.02	2E-08

Q9DCP2	SPS(Phospho)KEPHFTDFEGK	-0.87	4E-04
Q9JIG4	FAFQLPFAEGAS(Phospho)DGARLDFVVR	-0.81	7E-03
Q9JJL3	NSAS(Phospho)LHVLK	0.58	2E-03
Q9JLF6	GS(Phospho)GVNAAGDGTIR	-0.63	6E-05
Q9JLF6	RPES(Phospho)RGSGVNAAGDGTIR	-0.63	4E-04
Q9QXG4	GWS(Phospho)PPPEVR	-0.71	4E-04
Q9QXG4	GWS(Phospho)PPPEVRR	-0.56	2E-03
Q9QXG4	VRGWS(Phospho)PPPEVR	-0.67	1E-04
Q9QY06	AQDKPES(Phospho)PSGSTQIQR	0.52	6E-04
Q9QY30	SQLSHLS(Phospho)HEPPLAIGDHK	-0.91	7E-04
Q9QZW0	ASDSLSARPS(Phospho)VRPLLLR	0.66	9E-03
Subgroup_0_2 /Q64459 /Q9JMA7	ALLSPTFTS(Phospho)GK	-1.21	3E-07
Subgroup_0_5 /Q32Q92 /Q6Q2Z6 /Q8BWN8 /Q9QYR7 /O55137	YRADS(Phospho)HGELDLAR	0.67	8E-03
Subgroup_1_1 /P10649	RYT(Phospho)MGDAPDFDR	-0.52	6E-03
Subgroup_1_1 /Q61425	SMS(Phospho)SSSSASAAAK	1.07	2E-04
Subgroup_1_1 /Q8C0N2	NSAS(Phospho)VGIIQR	0.71	7E-04
Subgroup_1_1 /Q8C0N2	NSAS(Phospho)VGIIQRDESPMEK	0.80	3E-04
Subgroup_1_1 /Q99K28	HGTDLWIDSMNSAPS(Phospho)HSPEKK	0.52	3E-03
Subgroup_1_1 /Q99K28	HGTDLWIDSMNSAPSHS(Phospho)PEKK	0.53	1E-03
Subgroup_2_1 /O88343	MFSNPDNGS(Phospho)PAMTHR	0.78	1E-06

7.1.6. Potentially due to genetics regulated phosphopeptides in the BXD mouse reference population

Table 24: Due genetics regulated phosphopeptides in the BXD mouse reference population. *The list contain all phosphopeptides for which the Spearman correlation coefficient, between CD and HFD of the BXD mouse samples, were higher than 0.5. This list of potentially genetically regulated phosphoproteins were used for further analysis of the phospho-pQTLs.*

Proteins	Phosphopeptide	Spearman
Q8BYU6	AQEHT(Phospho)DTGDRSESPEEPALEKPPLDK	0.87

Q8BYU6	AQEHTDT(Phospho)GDRSESPEEPALEKPPLDK	0.83
Q8BYU6	AQEHTDTGDRS(Phospho)ESPEEPALEKPPLDK	0.82
Q8VI47	KQS(Phospho)QSQDVLVLEDSK	0.81
Q6P542	SKPAAADS(Phospho)EGEEEEEDTAK	0.80
Q8C0N2	NSAS(Phospho)VGIIQR	0.80
Subgroup_1_1 /Q8C0N2	NSAS(Phospho)VGIIQR	0.79
Q8C5R2	LAGNEALSPTS(Phospho)PSK	0.77
Q9JLF6	SPWQPPTTPS(Phospho)PEPEPEPEPDRR	0.74
Q9WU19	NFETNDLAFS(Phospho)PK	0.73
Q9CR00	RLASNS(Phospho)PVLPQAFAR	0.72
Q9CR00	LAS(Phospho)NSPVLPQAFAR	0.71
P58871	NRS(Phospho)AEEGEVTESK	0.71
Q3UJU9	SHS(Phospho)LPNSLDYAQASER	0.69
Q6PGL7	ARPAQAPVSEELPPS(Phospho)PKPGK	0.68
Subgroup_0_2 /Q61301 /P26231	SRT(Phospho)SVQTEDDQLIAGQSAR	0.68
Q9CR00	LASNS(Phospho)PVLPQAFAR	0.67
Subgroup_0_2 /Q61301 /P26231	SRTS(Phospho)VQTEDDQLIAGQSAR	0.67
Q80XI3	TSS(Phospho)PTSLPPLAR	0.66
Q3UJU9	S(Phospho)HSLPNSLDYAQASER	0.66
P46935	RQIS(Phospho)EDVDGPDNR	0.65
Q64464	VVSRDETVS(Phospho)DE	0.65
Subgroup_1_1 /Q8C0N2	NSAS(Phospho)VGIIQRDESPMEK	0.64
Q99P72	GPLPAAPPTAPERQPS(Phospho)WER	0.64
O08547	NLGS(Phospho)INTELQDVQR	0.64
O08547	RNLGS(Phospho)INTELQDVQR	0.64
P83093	RAS(Phospho)GSAGAAASPSAAAAGER	0.64
Q99M51	RKPS(Phospho)VPDTASPADDSFVDPGER	0.64
Q3UEI1	SSHT(Phospho)SLPTAAIPR	0.63
Q8BJ37	KHVSS(Phospho)PDVTTAQK	0.62
Q99KU0	DQHNGS(Phospho)LTDPSSVHEK	0.62
Q3UPH7	RIQQQLGEEAS(Phospho)PR	0.61
Q91X91	LFAEGDT(Phospho)PVPHAR	0.60
Subgroup_1_1 /P50136	IGHHSTS(Phospho)DDSSAYR	0.59
Subgroup_1_1 /P27546	DMS(Phospho)PSAETEAPLAK	0.59
Q8K3K8	KNS(Phospho)ATPSELNEK	0.59
Q05915	ELPRPGAS(Phospho)PPAEK	0.58
O08705	AAATEDAT(Phospho)PAALEK	0.56
Q05915	HRS(Phospho)EEENQVNLPK	0.55
Q3UEI1	VREPVDSGVAPVS(Phospho)PLGGGVILR	0.55
Q9R257	IPNQFQGS(Phospho)PPAPSDESVKIEER	0.55

Q9JMH9	SSSPTSHWKPLAPDPS(Phospho)DDEHDPVDSISRPR	0.54
Q8BTI8	RS(Phospho)SSELSPEVVEK	0.54
P10637	SGYSSPGS(Phospho)PGTPGSR	0.54
Q9JLF6	RPESRGS(Phospho)GVNAAGDGTIR	0.54
Q9DBC7	TDSREDEIS(Phospho)PPPPNPVVK	0.53
Subgroup_0_2 /P26231 /Q61301	S(Phospho)RTSVQTEDDQLIAGQSAR	0.53
Q9QZW0	ASDSLSARPS(Phospho)VRPLLLR	0.53
P11679	VGS(Phospho)SSSSFR	0.53
Q3UTJ2	TS(Phospho)PGRADLPGSSSTFTK	0.52
P10518	DAAQSS(Phospho)PAFGDRR	0.52
Q63918	SSPFKVS(Phospho)PLSFGR	0.52
P16015	HDPSLQPWS(Phospho)ASYDPGSAK	0.52
Q8K4G5	STS(Phospho)QGSINSPVYSR	0.52
P50136	IGHHST(Phospho)SDDSSAYR	0.51
Q3UPH7	IQQQLGEEAS(Phospho)PR	0.51
Q9D8T7	ALHGAQTS(Phospho)DEER	0.51
P47963	KGDSS(Phospho)AEELK	0.51
Q9QXS1	SSS(Phospho)VGSSSSYPISSAGPR	0.51
Q6ZQA0	RIS(Phospho)QVSSGETEYNPGEAR	0.51
Q8BVZ1	GASPS(Phospho)PTFHPPK	0.51
P11862	EIEQEETLSAPSPS(Phospho)PSPSSK	0.51
Q8BK03	GDGGS(Phospho)TPTPGDSLQNPDTASEALSEPESQRR	0.51
Q9JLF6	RPES(Phospho)RGSGVNAAGDGTIR	0.51
Subgroup_1_1 /P97351	ADGYEPPVQES(Phospho)V	0.50
Q9QZQ1	S(Phospho)QEELREEK	0.50

7.2. R-scripts

The R-scripts for the data analysis are added. The whole statistical analysis was conducted via R.

```
##############################################################################################################################################################
##############################################################################################################################################################
##                                                                                                                                                          ##
##        R - code used for the thesis "Phosphoproteomic analysis of liver in mouse reference population" (ETH, BOKU 2016)                                 ##
##                                                                                                                                                          ##
##                                                                                                                                                          ##
##############################################################################################################################################################
##############################################################################################################################################################
##                                                                                                                                                          ##
##                                                                                                                                                          ##
## Contains the following scripts:  "Beads and buffer combinations analysis"                                                                                ##
##                                  "Paramter optimization for phosphopeptide enrichment with Ti4+-IMAC"                                                    ##
##                                  "SWATH2stats for OpenSWATH data - one example"                                                                          ##
##                                  "Refine a phospho-SWATH-MS library"                                                                                     ##
##                                  "Comparision of the three SWATH-MS libraries"                                                                           ##
##                                  "LFQ & phospho-SWATH-MS comparison in the aging dataset"                                                                ##
##                                  "Analysis of peptide identification results in DDA & DIA results of the aging experiment"                               ##
##                                  "Analysis mapDIA output of the BXD-mouse reference population samples"                                                  ##
##                                  "QTL analysis with the R/qtl package"                                                                                   ##
##                                  "Summary plots of the CV in different MS-measurements and experiments"                                                  ##
##                                                                                                                                                          ##
##############################################################################################################################################################
##############################################################################################################################################################
```

```
##############################################################################################################################################################
#                                                                                                                                                            #
#                 "Beads and buffer combinations analysis"                                                                                                   #
#                                                                                                                                                            #
# Author: Fabian Frommelt                                                                                                                                    #
# Date: 11.03.2016                                                                                                                                           #
# Summary: Several Beads and Buffer combinations to optimize phosphopeptide enrichment of mouse liver tissue were tested and the                             #
#          resulting data were plotted                                                                                                                       #
##############################################################################################################################################################

# Uses iPortal and MaxQuant search engine results for analysis of the best performing beads to peptide combination

setwd("Y:\\160311_Beads_to_peptide/")

# load required R packages
library(gplots)
library(ggplot2)
library(stringr)
library(gridExtra)
library(reshape2)
library(VennDiagram)

# import the peptide search results of iportal
file.name <- "peptides.tsv"
data_Comet_1 <- read.table(file.path("Y:\\html\\openBIS\\20150902134302405-1096947\\", file.name), header=TRUE, sep="\t",
                           fill=TRUE, stringsAsFactors = FALSE)
data_Comet_2 <- read.table(file.path("Y:\\html\\openBIS\\20150917165705123-1100254\\", file.name), header=TRUE, sep="\t",
                           fill=TRUE, stringsAsFactors = FALSE)
data_Comet <- rbind(data_Comet_1, data_Comet_2)

data_OmXT_1 <- read.table(file.path("Y:\\html\\openBIS\\20150902143744739-1096988\\", file.name), header=TRUE, sep="\t",
                          fill=TRUE, stringsAsFactors = FALSE)
data_OmXT_2 <- read.table(file.path("Y:\\html\\openBIS\\20150917171924962-1100273\\", file.name), header=TRUE, sep="\t",
                          fill=TRUE, stringsAsFactors = FALSE)
data_OmXT <- rbind(data_OmXT_1, data_OmXT_2)

data_CoOmXT_1 <- read.table(file.path("Y:\\html\\openBIS\\20150917173026063-1100287\\", file.name), header=TRUE, sep="\t",
                            fill=TRUE, stringsAsFactors = FALSE)
data_CoOmXT_2 <- read.table(file.path("Y:\\html\\openBIS\\20150901180138930-1096667\\", file.name), header=TRUE, sep="\t",
                            fill=TRUE, stringsAsFactors = FALSE)
data_CoOmXT <- rbind(data_CoOmXT_1,data_CoOmXT_2)

# import the results of MaxQuant
file.name <- "Phospho (STY)Sites.txt"
data_mq_phospho_1 <- read.table(file.path("Y:\\Max_Quant\\MaxQuant_analysis_rep1_2\\txt", file.name), header=TRUE, sep="\t",
                                fill=TRUE, stringsAsFactors = FALSE,quote = "")
data_mq_phospho_2 <- read.table(file.path("Y:\\Max_Quant\\MaxQuant_analysis_rep3\\txt", file.name), header=TRUE, sep="\t",
                                fill=TRUE, stringsAsFactors = FALSE,quote = "")
file.name <- "peptides.txt"
data_mq_peptide_1 <- read.table(file.path("Y:\\Max_Quant\\MaxQuant_analysis_rep1_2\\txt", file.name), header=TRUE, sep="\t",
                                fill=TRUE, stringsAsFactors = FALSE,quote = "")
data_mq_peptide_2 <- read.table(file.path("Y:\\Max_Quant\\MaxQuant_analysis_rep3\\txt", file.name), header=TRUE, sep="\t",
                                fill=TRUE, stringsAsFactors = FALSE,quote = "")

pat <- "^[[:space:]]*$"
data_mq_peptide_1 <- data_mq_peptide_1[grepl(pat, data_mq_peptide_1$Phospho..STY..site.IDs),]
data_mq_peptide_2 <- data_mq_peptide_2[grepl(pat, data_mq_peptide_2$Phospho..STY..site.IDs),]

## annotate the phospho-sites of the MaxQuant output
phospho.mq.annotate <- function(data = dataframe(), threshhold = numeric())
{
  x <- (data[,c("Protein","Phospho..STY..Probabilities",colnames(data)[grep("Intensity.FF[[:digit:]]{3}$",colnames(data))])])
  colnames(x) <- gsub("Intensity.",replacement = "", x= colnames(x))
  th <- threshhold
  for (i in row(x)) {
    print(i)
    row <- i
    k <- x[i,]
    # save the pattern which should be grep
    grx <- c("\\([0-9].[0-9]+\\)|\\([0-9]{1}\\)")
    # count the amount of patterns


    count <- sapply("\\([0-9].[0-9]+\\)|\\([0-9]{1}\\)", str_count, string =k$Phospho..STY..Probabilities)
    count <- as.integer(count)

    for (q in 1:count) {
      q <- regmatches(k$Phospho..STY..Probabilities, regexpr(grx, k$Phospho..STY..Probabilities))
      q <- regmatches(q,regexpr("[0-9].[0-9]+|[0-9]{1}",q))

      logic <- q >= th


      if (logic == TRUE) {
        k$Phospho..STY..Probabilities <- sub(grx, replacement = "(Phospho)",x = k$Phospho..STY..Probabilities)
```

```
      } else if (logic == FALSE) {
        k$Phospho..STY..Probabilities <- sub(grx, replacement = "",x = k$Phospho..STY..Probabilities)
      }
    }
    x[i,] <- k
    if (i == length(x$Phospho..STY..Probabilities)) break
  }
  return(x)
}

data_mq_phospho_1 <- phospho.mq.annotate(data=data_mq_phospho_1, threshhold = 0.0)
data_mq_phospho_2 <- phospho.mq.annotate(data=data_mq_phospho_2, threshhold = 0.0)

# two mergeing and reshaping functions for the MaxQuant output
merge.mq.phospho.data <- function (data_1, data_2)
{
  data_a <- melt(data_1, id=c("Protein","Phospho..STY..Probabilities"))
  data_b <- melt(data_2, id=c("Protein","Phospho..STY..Probabilities"))
  data <- rbind(data_a, data_b)
  colnames(data)[colnames(data) == "Phospho..STY..Probabilities"] <- "modified_peptide"
  colnames(data)[colnames(data) == "Protein"] <- "protein"
  colnames(data)[colnames(data) == "variable"] <- "Sample_ID"
  colnames(data)[colnames(data) == "value"] <- "Intensity_phospho"
  return(data)
}

# merge the MaxQuant files, which contain the information of the unphosphorylated peptides detected in the dataset
merge.mq.peptide <- function(data_1, data_2)
{
  data_a <- (data_1[,c("Leading.razor.protein","Sequence",colnames(data_1)[grep("Intensity.FF[[:digit:]]{3}$",colnames(data_1))])])
  data_b <- (data_2[,c("Leading.razor.protein","Sequence",colnames(data_2)[grep("Intensity.FF[[:digit:]]{3}$",colnames(data_2))])])
  colnames(data_a) <- gsub("Intensity.",replacement = "", x= colnames(data_a))
  colnames(data_b) <- gsub("Intensity.",replacement = "", x= colnames(data_b))
  data_a <- melt(data_a, id=c("Leading.razor.protein","Sequence"))
  data_b <- melt(data_b, id=c("Leading.razor.protein","Sequence"))
  data <-rbind(data_a, data_b)
  colnames(data)[colnames(data) == "Leading.razor.protein"] <- "protein"
  colnames(data)[colnames(data) == "Sequence"] <- "peptide"
  colnames(data)[colnames(data) == "variable"] <- "Sample_ID"
  colnames(data)[colnames(data) == "value"] <- "Intensity_peptide"
  return(data)
}

data_mq_phospho_all <- merge.mq.phospho.data(data_mq_phospho_1, data_mq_phospho_2)
data_mq_peptide_all <- merge.mq.peptide(data_mq_peptide_1, data_mq_peptide_2)

# create one output file for the MaxQuant data
data_mq_phospho_all <- data_mq_phospho_all[data_mq_phospho_all[,4] > 0, ]
data_mq_peptide_all <- data_mq_peptide_all[data_mq_peptide_all[,4] > 0, ]
data_mq_phospho_all$peptide <- data_mq_phospho_all$modified_peptide
data_mq_phospho_all$peptide <- sapply(data_mq_phospho_all$peptide,gsub,pattern="*\\(Phospho\\)*",replacement="")

data_mq_all <- merge(data_mq_peptide_all, data_mq_phospho_all, by=c("protein", "Sample_ID", "peptide"), all.x = TRUE, all.y = TRUE)
data_mq_all <- data_mq_all[!grepl("CON__", data_mq_all$protein),]
data_mq_all <- data_mq_all[!grepl("REV__", data_mq_all$protein),]
data_mq_all$protein <- sub("(sp\\|)([[:alnum:]]+)(\\|[[:alnum:]]+_MOUSE)","\\2", data_mq_all$protein)

# annotate the phosphosites in a uniform way for all different outputs
annotate.phospho <- function(x, ...){
  x[, c("S_167", "T_181", "Y_243", "DECOY")] <- FALSE

  x$modified_peptide <- gsub("\\[167\\]", "(Phospho)", x$modified_peptide)
  x$modified_peptide <- gsub("\\[181\\]", "(Phospho)", x$modified_peptide)
  x$modified_peptide <- gsub("\\[243\\]", "(Phospho)", x$modified_peptide)
  x$modified_peptide <- gsub("\\[147\\]", "(Oxidation)", x$modified_peptide)
  x$modified_peptide <- gsub("\\[160\\]", "", x$modified_peptide)

  x[grep("S\\(Phospho\\)", x$modified_peptide),"S_167"] <- TRUE
  x[grep("T\\(Phospho\\)", x$modified_peptide),"T_181"] <- TRUE
  x[grep("Y\\(Phospho\\)", x$modified_peptide),"Y_243"] <- TRUE
  x[grep("\\DECOY\\_*", x$protein),"DECOY"] <- TRUE
  x$PHOSPHO <- rowSums(subset(x,select=c("S_167", "T_181", "Y_243")))
  x$PHOSPHO <- as.logical(x$PHOSPHO)
  return(x)
}

# Calculate the delocalized forms of the phosphopeptides
delocalize.phospho <- function(data = dataframe())
{
  # http://stackoverflow.com/questions/19666965/count-pattern-matching-in-r
  # from this site I got the hint with the str_count command
  x <-data
  x$Count_Phospho <- sapply("(Phospho)", str_count, string =x$modified_peptide)
  x$Deloc <- x$modified_peptide
  x$Deloc <- sapply(x$Deloc,gsub,pattern="*\\(Phospho\\)*",replacement="")
  x$Deloc <- sapply(x$Deloc,gsub,pattern="*\\(Oxidation\\)*",replacement="")
  x$Delocalized <- paste(x$Deloc, x$Count_Phospho, sep="_P")
  x <- subset(x, select = -c(Deloc))
  x$Delocalized <- gsub(x$Delocalized, pattern = "NA_PNA", replacement = NA )
  x$Delocalized <- gsub(x$Delocalized, pattern = "\\_P0", replacement = "" )
  return(x)
}

# apply the functions to the datasets
data_Comet <- annotate.phospho(data_Comet)
data_Comet <- delocalize.phospho(data_Comet)

data_OmXT <- annotate.phospho(data_OmXT)
data_OmXT <- delocalize.phospho(data_OmXT)

data_CoOmXT <- annotate.phospho(data_CoOmXT)
data_CoOmXT <- delocalize.phospho(data_CoOmXT)

data_mq_all <- annotate.phospho(data_mq_all)
data_mq_all <- delocalize.phospho(data_mq_all)

# annotate the data
annotation.file <- "Study_design_MaxQuant.txt"
Study_design <- read.delim2(file.path(getwd(), annotation.file), dec=".", sep ="\t", header=TRUE)
data_mq_all <- merge(data_mq_all, Study_design, by = "Sample_ID")

# function adapted from the SWATH2stats package for the annotation of the iPortal data
annotation.iportal <-

  function (data, sample.annotation, data.type = "iportal", column.file = "spectrum",
```

```
          change.run.id = TRUE, verbose = FALSE)
  {

    if (!(column.file %in% colnames(data))) {
      warning("Warning: column for spectrum is not present in data file")
    }


    if (nlevels(factor(paste(sample.annotation$spectrum))) !=
        nlevels(factor(data[, column.file]))) {
      stop("Warning: the number of sample annotation condition and spectrum in data are not balanced.",
           "\n", "Different filenames in sample annotation file: ",
           nlevels(factor(sample.annotation$Condition)), "\n",
           "Different filenames in data file: ", nlevels(factor(data[,
                                                                     column.file])))
    }

    if (data.type == "iportal") {
      colnames(data) <- gsub("Run", column.file, colnames(data))
      for (i in levels(sample.annotation$spectrum)) {

        coord <- grep(i, data[, column.file])
        if (length(coord) == 0) {
          warning("No measurement value found for this sample in the data file: ",
                  print(i))
        }

        data.subset <- sample.annotation[which(i == sample.annotation$spectrum),
                                         ]
        data[coord, "Sample_ID"] <- data.subset[, "Sample_ID"]
        data[coord, "TechReplicate"] <- data.subset[, "TechReplicate"]
        data[coord, "Lysis"] <- data.subset[, "Lysis"]
        data[coord, "Beads"] <- data.subset[, "Beads"]
        data[coord, "Loading_buffer"] <- data.subset[, "Loading_buffer"]
        data[coord, "Engine"] <- data.subset[, "Engine"]
      }
      add.colnames <- colnames(data)[!(colnames(data) %in%
                                         c("Sample_ID", "peptide", "modified_peptide", "protein","S_167", "T_181", "Y_243",
                                           "DECOY", "PHOSPHO", "Count_Phospho", "Delocalized",
                                           "TechReplicate", "Lysis", "Beads", "Loading_buffer", "Engine"))]
      data <- data[, c("Sample_ID", "protein","peptide", "modified_peptide", "S_167", "T_181", "Y_243",
                       "DECOY", "PHOSPHO", "Count_Phospho", "Delocalized",
                       "TechReplicate", "Lysis", "Beads", "Loading_buffer", "Engine",
                       add.colnames)]
      return(data)
    }
  }

# annotate the three iPortal outputs
annotation.file <- "Study_design_Comet.txt"
Study_design <- read.delim2(file.path(getwd(), annotation.file), dec=".", sep ="\t", header=TRUE)
data_Comet$spectrum <- gsub(pattern="*~.*",replacement="", data_Comet$spectrum )
data_Comet <- annotation.iportal(data_Comet, Study_design)

annotation.file <- "Study_design_OmXT.txt"
Study_design <- read.delim2(file.path(getwd(), annotation.file), dec=".", sep ="\t", header=TRUE)
data_OmXT$spectrum <- gsub(pattern="*~.*",replacement="", data_OmXT$spectrum )
data_OmXT <- annotation.iportal(data_OmXT, Study_design)

annotation.file <- "Study_design_CoOmXT.txt"
Study_design <- read.delim2(file.path(getwd(), annotation.file), dec=".", sep ="\t", header=TRUE)
data_CoOmXT$spectrum <- gsub(pattern="*~.*",replacement="", data_CoOmXT$spectrum )
data_CoOmXT <- annotation.iportal(data_CoOmXT, Study_design)

# Bind all iPortal lists together and subset all information of the dataset needed  for combining it with the MaxQuant output
data_iportal <- do.call("rbind", list(data_Comet, data_OmXT, data_CoOmXT))
n_data_iportal <- subset(data_iportal, select = c(Sample_ID, protein, peptide, modified_peptide, PHOSPHO, DECOY, Delocalized, Engine))
n_data_maxquant <- subset(data_mq_all, select = c(Sample_ID, protein, peptide, modified_peptide, PHOSPHO, DECOY, Delocalized, Engine))
n_data_all <- do.call("rbind", list(n_data_iportal, n_data_maxquant))
n_data_all <- unique(n_data_all)

# create single lists
sub_PEPIDE <- subset(n_data_all, DECOY == FALSE)
sub_DECOY <- subset(n_data_all, DECOY == TRUE)
sub_PHOSPHO <-subset(n_data_all, DECOY == FALSE & PHOSPHO == TRUE)
sub_PHOSPHO_Protein <- subset(n_data_all, DECOY == FALSE & PHOSPHO == TRUE)
sub_PHOSPHO_Protein <- unique(subset(sub_PHOSPHO_Protein, select = c(Sample_ID, protein, Engine)))
sub_DELOCALIZED <- subset(n_data_all, DECOY == FALSE & PHOSPHO == TRUE)
sub_DELOCALIZED <- unique(subset(sub_DELOCALIZED, select = c(Sample_ID, protein, Engine, Delocalized)))

# Plots for a quick overview over every single run

p1 <-ggplot(sub_PEPIDE, aes(factor(Sample_ID))) +
  geom_bar(aes(fill =Sample_ID)) +
  facet_wrap( ~Engine, ncol = 2) +
  ggtitle("Identified phopeptides for all iportal settings") +
  ylim(0, 11000) +
  labs(x = "", y = "counts")+
  theme(
    axis.text.x = element_text(size = 6, angle = 45),
    axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 10),
    axis.title.y = element_text(size = 10),
    plot.title = element_text(size = 14),
    strip.text.x = element_text(size = 10))
plot(p1)

p2 <-ggplot(sub_PHOSPHO, aes(factor(Sample_ID))) +
  geom_bar(aes(fill =Sample_ID)) +
  facet_wrap( ~Engine, ncol = 2) +
  ggtitle("Identified unique phosphophopeptides for all iportal settings") +
  ylim(0, 2700) +
  labs(x = "", y = "counts")+
  theme(
    axis.text.x = element_text(size = 6, angle = 45),
    axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 10),
    axis.title.y = element_text(size = 10),
    plot.title = element_text(size = 14),
    strip.text.x = element_text(size = 10))
plot(p2)

p3 <-ggplot(sub_DELOCALIZED, aes(factor(Sample_ID))) +
  geom_bar(aes(fill =Sample_ID)) +
  facet_wrap( ~Engine, ncol = 2) +
```

```
  ggtitle("Identified unique delocalized phosphopeptdies for all iportal settings") +
  ylim(0, 2300) +
  labs(x = "", y = "counts")+
  theme(
    axis.text.x = element_text(size = 6, angle = 45),
    axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 10),
    axis.title.y = element_text(size = 10),
    plot.title = element_text(size = 14),
    strip.text.x = element_text(size = 10))
plot(p3)


p4 <-ggplot(sub_PHOSPHO_Protein, aes(factor(Sample_ID))) +
  geom_bar(aes(fill =Sample_ID)) +
  facet_wrap( ~Engine, ncol = 2) +
  ggtitle("Identified unique phosphoproteins for all iportal settings") +
  ylim(0, 1190) +
  labs(x = "", y = "counts")+
  theme(
    axis.text.x = element_text(size = 6, angle = 45),
    axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 10),
    axis.title.y = element_text(size = 10),
    plot.title = element_text(size = 14),
    strip.text.x = element_text(size = 10))
plot(p4)

# function for calculating the different species in all datasets
count.species <- function(x, species=c()){

  if (species == "phospho_proteins") {
    x <- melt(x, id =c("protein", "Engine"))
    x$variable <- 1
    x <- dcast(x, value ~ protein + Engine, value.var = "variable")
  } else if (species == "peptide"){
    ## is a rather complex fragment of code ...
    ## in fact, there are some peptides, which have different modifications so therefore we need to combine the once with modifcation,
    ## and the once which do not have any modifaction in one row, to really count the number of detected peptides, because it can also be,
    ## so that we account for peptides with the phosphorylation on different sites.

    x <- subset(x, select = c(Sample_ID, peptide, Engine, protein, modified_peptide))
    index <- x$modified_peptide == is.na(TRUE)
    index[is.na(index)] <- TRUE
    x$modified_peptide[index] <- (x$peptide[index])
    x <- subset(x, select = c(Sample_ID, modified_peptide, Engine, protein))
    x <- melt(subset(x, select = c(Sample_ID, modified_peptide, Engine)), id =c("modified_peptide", "Engine"))
    x <- unique(x)
    x$variable <- 1
    x <- dcast(x, value ~ modified_peptide + Engine, value.var = "variable")
  } else if (species == "phospho_peptide") {

    ## produces the same result as in the earlier analysis. I am not so sure about the unique. The issue is,
    ## the number of phosphopeptide do not alter, if you take the Proteins into account and therefore I would rather suggest not to unique.
    ## In the protein list in the peptides.tsv list are only proteotpyic peptides (checked it in the excel)
    x <- subset(x, select = c(Sample_ID, modified_peptide, Engine, protein))
    x <- melt(subset(x, select = c(Sample_ID, modified_peptide, Engine)), id =c("modified_peptide", "Engine"))
    x <- unique(x)
    x$variable <- 1
    x <- dcast(x, value ~ modified_peptide + Engine, value.var = "variable")

  } else if (species == "delocalized") {
    ## same about the delocalized as for the peptides, if the protein is took into account, there is not one entrey removed.
    ## Therefore we should stick
    ## here also to the non reduced one.
    x <- melt(subset(x, select = c(Sample_ID, Delocalized, Engine)), id =c("Delocalized", "Engine"))
    x <- unique(x)
    x$variable <- 1
    x <- dcast(x, value ~ Delocalized + Engine, value.var = "variable")
  } else if (species == "decoy"){
    x <- sub_DECOY
    x <- melt(subset(x, select = c(Sample_ID, protein, Engine)), id =c("protein", "Engine"))
    x <- unique(x)
    x$variable <- 1
    x <- dcast(x, value ~ protein + Engine, value.var = "variable")
  }


  data <- x
  colname <- c()
  colname <- species
  data$Comet <- apply(data[,grep("*_Comet", colnames(data))], 1, function(x)sum(x, na.rm = TRUE))
  data$OmXT <- apply(data[,grep("*_OmXT", colnames(data))], 1, function(x)sum(x, na.rm = TRUE))
  data$CoOmXT <- apply(data[,grep("*_CoOmXT", colnames(data))], 1, function(x)sum(x, na.rm = TRUE))
  data$MaxQuant <- apply(data[,grep("*_MaxQuant", colnames(data))], 1, function(x)sum(x, na.rm = TRUE))
  data <- melt(subset(data, select =c(value, Comet, OmXT, CoOmXT, MaxQuant)), id="value")
  colnames(data)[3] <- colname[1]
  return(data)
}

# Count the Species
count_sub_PHOSPHO_Protein <- count.species(sub_PHOSPHO_Protein, species = "phospho_proteins")
count_sub_PEPIDE <- count.species(sub_PEPIDE, species = "peptide")
count_sub_PHOSPHO <- count.species(sub_PHOSPHO, species = "phospho_peptide")
count_sub_DELOCALIZED <- count.species(sub_DELOCALIZED, species = "delocalized")
count_sub_DECOY <- count.species(sub_DECOY, species = "decoy")

# merge function with reduce; the idea was taken from http://stackoverflow.com/questions/14096814/r-merging-a-lot-of-data-frames
count_merge <- Reduce(function(x,y) merge(x,y, all=TRUE), list(count_sub_PHOSPHO_Protein, count_sub_PEPIDE,
                                                               count_sub_PHOSPHO, count_sub_DELOCALIZED, count_sub_DECOY))
count_merge <- count_merge[order(count_merge$variable),]
count_merge$enrichment <- round(count_merge$phospho_peptide/count_merge$peptide, digits = 2)
colnames(count_merge)[colnames(count_merge) == "value"] <- "Sample_ID"
colnames(count_merge)[colnames(count_merge) == "variable"] <- "Engine"

# annotate the data again
annotation.file <- "Study_design_plotting.txt"
Study_design <- read.delim2(file.path(getwd(), annotation.file), dec=".", sep ="\t", header=TRUE)
count_merge <- merge(count_merge, Study_design, by = "Sample_ID")


# create plots of the data for visualization
p5 <-ggplot(count_merge, aes(factor(x = Beads), y = phospho_peptide)) +
  geom_boxplot(aes(fill =Loading_buffer)) +
  facet_wrap( ~Engine + Lysis, ncol = 4) +
```

```
  ggtitle("Unique phosphopeptide counts for all beads and buffer combinations split \n by lysis and search engine") +

  labs(x = "", y = "counts")+
  theme(
    axis.text.x = element_text(size = 14, angle = 45, hjust = 1, colour = "black"),
    axis.text.y = element_text(size = 14),
    axis.title = element_text(size = 14),
    axis.title.y = element_text(size = 14),
    plot.title = element_text(size = 18, face = "bold"),
    strip.text.x = element_text(size = 14))
plot(p5)

## delocalized phosopho peptides
p6 <-ggplot(count_merge, aes(factor(x = Beads), y = delocalized)) +
  geom_boxplot(aes(fill =Loading_buffer)) +
  facet_wrap( ~Engine + Lysis, ncol = 4) +
  ggtitle("Delocalized phosphopeptide counts for all beads and buffer combinations split \n by lysis and search engine") +

  labs(x = "", y = "counts")+
  theme(
    axis.text.x = element_text(size = 8, angle = 45, hjust = 1, colour = "black"),
    axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 10),
    axis.title.y = element_text(size = 10),
    plot.title = element_text(size = 12, face = "bold"),
    strip.text.x = element_text(size = 10))
plot(p6)

## enrichment factor for each of the combinations
p7 <-ggplot(count_merge, aes(factor(x = Beads), y = enrichment)) +
  geom_boxplot(aes(fill =Loading_buffer)) +
  facet_wrap( ~Engine + Lysis, ncol = 4) +
  ggtitle("Enrichment Factor for all beads and buffer combinations split \n by lysis and search engine") +

  labs(x = "", y = "percentages")+
  theme(
    axis.text.x = element_text(size = 14, angle = 45, hjust = 1, colour = "black"),
    axis.text.y = element_text(size = 14),
    axis.title = element_text(size = 14),
    axis.title.y = element_text(size = 14),
    plot.title = element_text(size = 18, face = "bold"),
    strip.text.x = element_text(size = 14))
plot(p7)

## Venn Diagrams
## For the Venn Diagrams the "delocalized" phosphopeptides are taken into account
head(sub_DELOCALIZED)
data <- sub_DELOCALIZED
cols <- c("Sample_ID", "Engine")
data$Sample_Engine <- do.call(paste, c(data[cols], sep="_"))
for (co in cols) data[co] <- NULL
data <-data[,-1]
data$Sample_Engine <- as.factor(data$Sample_Engine)

split_data<- split(data, f=data$Sample_Engine, drop =TRUE)
split_data <- unlist(split_data, recursive=FALSE)


venn.plot <- venn.diagram(
  x=c(list(split_data$FF143_CoOmXT.Delocalized),
      list(split_data$FF145_CoOmXT.Delocalized)),
  filename = NULL,
  sacaled = TRUE,
  main = "Beads Ti-IMAC, PCT lysis, CoOmXT, 6% TFA in 80% ACN - delocalized phosphopeptides",
  col = "black",
  fill = c("blue", "green"),
  category = c("FF143_CoOmXT",
               "FF145_CoOmXT"),
  alpha = 0.50,
  cat.col = c("Black"),
  cat.cex = 1.0,
  main.cex = 1.2,
  cat.fontface = "bold",
  margin = 0.15);
grid.draw(venn.plot);
dev.off()

venn.plot <- venn.diagram(
  x=c(list(split_data$FF147_CoOmXT.modified_peptide),
      list(split_data$FF146_CoOmXT.modified_peptide),
      list(split_data$FF137_CoOmXT.modified_peptide)),
  filename = NULL,
  sacaled = TRUE,
  main = "Beads Ti-IMAC, CON lysis, CoOmXT, 6% TFA in 80% ACN - delocalized phosphopeptides",
  col = "black",
  fill = c("blue", "green", "purple"),
  category = c("FF147_CoOmXT",
               "FF147_CoOmXT",
               "FF137_CoOmXT"),
  alpha = 0.50,
  cat.col = c("Black"),
  cat.cex = 1.0,
  main.cex = 1.2,
  cat.fontface = "bold",
  margin = 0.15);
grid.draw(venn.plot);
dev.off()


## Venn Plot with phosphopeptides
head(sub_PHOSPHO)
data <- sub_PHOSPHO
cols <- c("Sample_ID", "Engine")
data$Sample_Engine <- do.call(paste, c(data[cols], sep="_"))
for (co in cols) data[co] <- NULL
data <-data[,-c(1:2,4:6)]
data$Sample_Engine <- as.factor(data$Sample_Engine)

split_data<- split(data, f=data$Sample_Engine, drop =TRUE)
split_data <- unlist(split_data, recursive=FALSE)

venn.plot <- venn.diagram(
  x=c(list(split_data$FF143_CoOmXT.modified_peptide),
      list(split_data$FF145_CoOmXT.modified_peptide)),
  filename = NULL,
```

```
  sacaled = TRUE,
  main = "Beads Ti-IMAC, PCT lysis, CoOmXT, 6% TFA in 80% ACN - phosphopeptides",
  col = "black",
  fill = c("blue", "green"),
  category = c("FF143_CoOmXT",
               "FF145_CoOmXT"),
  alpha = 0.50,
  cat.col = c("Black"),
  cat.cex = 1.0,
  main.cex = 1.2,
  cat.fontface = "bold",
  margin = 0.15);
grid.draw(venn.plot);
dev.off()

## Venn plot for the three Ti-IMAC conventional lysis samples

venn.plot <- venn.diagram(
  x=c(list(split_data$FF147_CoOmXT.modified_peptide),
      list(split_data$FF146_CoOmXT.modified_peptide),
      list(split_data$FF137_CoOmXT.modified_peptide)),
  filename = NULL,
  sacaled = TRUE,
  main = "Beads Ti-IMAC, CON lysis, CoOmXT, 6% TFA in 80% ACN - phosphopeptides",
  col = "black",
  fill = c("blue", "green", "purple"),
  category = c("FF147_CoOmXT",
               "FF147_CoOmXT",
               "FF137_CoOmXT"),
  alpha = 0.50,
  cat.col = c("Black"),
  cat.cex = 1.0,
  main.cex = 1.2,
  cat.fontface = "bold",
  margin = 0.15);
grid.draw(venn.plot);
dev.off()


#########################################################################################################################################################
#                                                                                                                                                       #
#                 "Beads and buffer combinations analysis"                                                                                              #
#                                                                                                                                                       #
# Author: Fabian Frommelt                                                                                                                               #
# Date: 22.03.2016                                                                                                                                      #
# Summary: R-script based analysis of the second "optimization experiment" dealing with the optimization of the starting material                       #
#          and the optimization of the beads to peptide (or beads to starting material measured via BCA) ratio. The optimization is done               #
#          for Ti-IMAC beads with mouse liver tissue to find the optimal conditions for the aging and the BXD-mouse reference                           #
#          population experiment.                                                                                                                       #
#########################################################################################################################################################


# Uses iPortal and MaxQuant search engine results for analysis of the best performing condtions for Ti-IMAC

rm(list=ls())
setwd("Y:\\20160322_starting_beads_ratio_2nd/")

# load required R packages
library(gplots)
library(ggplot2)
library(stringr)
library(gridExtra)
library(reshape2)
library(VennDiagram)

# import the peptide search results of iportal
# also the two Hela-Control samples are imported to check if the result workded out in a proper way
file.name <- "peptides.tsv"
data_iportal_CoOmXT <- read.table(file.path("Y:\\html\\openBIS\\20151012175432790-1107749\\", file.name), header=TRUE, sep="\t", fill=TRUE, stringsAsFactors =
FALSE)
data_iportal_CoOmXT_Hela <- read.table(file.path("Y:\\html\\openBIS\\20151008125432718-1106512\\", file.name), header=TRUE, sep="\t", fill=TRUE,
stringsAsFactors = FALSE)
data_CoOmXT <- rbind(data_iportal_CoOmXT,data_iportal_CoOmXT_Hela)

# import the results of MaxQuant
file.name <- "Phospho (STY)Sites.txt"
data_mq_phospho <- read.table(file.path("Y:\\Max_Quant\\MaxQuant_analysis_151012\\txt", file.name), header=TRUE, sep="\t", fill=TRUE, stringsAsFactors =
FALSE,quote = "")
data_mq_phospho_Hela <- read.table(file.path("Y:\\Max_Quant\\MaxQuant_analysis_151008-Hela_control\\txt", file.name), header=TRUE, sep="\t", fill=TRUE,
stringsAsFactors = FALSE,quote = "")
file.name <- "peptides.txt"
data_mq_peptide <- read.table(file.path("Y:\\Max_Quant\\MaxQuant_analysis_151012\\txt", file.name), header=TRUE, sep="\t", fill=TRUE, stringsAsFactors =
FALSE,quote = "")
data_mq_peptide_Hela<- read.table(file.path("Y:\\Max_Quant\\MaxQuant_analysis_151008-Hela_control\\txt", file.name), header=TRUE, sep="\t", fill=TRUE,
stringsAsFactors = FALSE,quote = "")

# Subset df to remove rows fully composed of elements matching `pat`
pat <- "^[[:space:]]*$"
data_mq_peptide_Hela <- data_mq_peptide_Hela[grepl(pat, data_mq_peptide_Hela$Phospho..STY..site.IDs),]
data_mq_peptide <- data_mq_peptide[grepl(pat, data_mq_peptide$Phospho..STY..site.IDs),]


# annotation function (generic programmed in the "Beads and buffer combinations" analysis)
data_mq_phospho <- phospho.mq.annotate(data=data_mq_phospho, threshhold = 0.0)
data_mq_phospho_Hela <- phospho.mq.annotate(data=data_mq_phospho_Hela, threshhold = 0.0)

# merge functioons (generic programmed in the "Beads and buffer combinations" analysis)
data_mq_phospho_all <- merge.mq.phospho.data(data_mq_phospho, data_mq_phospho_Hela)
data_mq_peptide_all <- merge.mq.peptide(data_mq_peptide, data_mq_peptide_Hela)


data_mq_phospho_all <- data_mq_phospho_all[data_mq_phospho_all[,4] > 0, ]
data_mq_peptide_all <- data_mq_peptide_all[data_mq_peptide_all[,4] > 0, ]
data_mq_phospho_all$peptide <- data_mq_phospho_all$modified_peptide
data_mq_phospho_all$peptide <- sapply(data_mq_phospho_all$peptide,gsub,pattern="*\\(Phospho\\)*",replacement="")

data_mq_all <- merge(data_mq_peptide_all, data_mq_phospho_all, by=c("protein", "Sample_ID", "peptide"), all.x = TRUE, all.y = TRUE)
data_mq_all <- data_mq_all[!grepl("CON__", data_mq_all$protein),]
data_mq_all <- data_mq_all[!grepl("REV__", data_mq_all$protein),]
data_mq_all$protein <- sub("(sp\\|)([[:alnum:]]+)(\\|[[:alnum:]]+_MOUSE)","\\2", data_mq_all$protein)
data_mq_all$protein <- sub("(sp\\|)([[:alnum:]]+)(\\|[[:alnum:]]+_HUMAN)","\\2", data_mq_all$protein)
data_mq_all <- data_mq_all[!grepl("tr\\|*", data_mq_all$protein),]

# annoate phospho and delocalization functions (generic programmed in the "Beads and buffer combinations" analysis)
data_CoOmXT <- annotate.phospho(data_CoOmXT)
```

```
data_CoOmXT <- delocalize.phospho(data_CoOmXT)

data_mq_all <- annotate.phospho(data_mq_all)
data_mq_all <- delocalize.phospho(data_mq_all)

annotation.file <- "Study_design_MaxQuant.txt"
Study_design <- read.delim2(file.path(getwd(), annotation.file), dec=".", sep ="\t", header=TRUE)
data_mq_all <- merge(data_mq_all, Study_design, by = "Sample_ID")

# adapted from the SWATH2stats package
annotation.iportal <-

  function (data, sample.annotation, data.type = "iportal", column.file = "spectrum",
            change.run.id = TRUE, verbose = FALSE)
  {

    if (!(column.file %in% colnames(data))) {
      warning("Warning: column for spectrum is not present in data file")
    }


    if (nlevels(factor(paste(sample.annotation$spectrum))) !=
        nlevels(factor(data[, column.file]))) {
      stop("Warning: the number of sample annotation condition and spectrum in data are not balanced.",
           "\n", "Different filenames in sample annotation file: ",
           nlevels(factor(sample.annotation$Condition)), "\n",
           "Different filenames in data file: ", nlevels(factor(data[,
                                                                     column.file])))
    }

    if (data.type == "iportal") {
      colnames(data) <- gsub("Run", column.file, colnames(data))
      for (i in levels(sample.annotation$spectrum)) {

        coord <- grep(i, data[, column.file])
        if (length(coord) == 0) {
          warning("No measurement value found for this sample in the data file: ",
                  print(i))
        }

        data.subset <- sample.annotation[which(i == sample.annotation$spectrum),
                                         ]
        data[coord, "Sample_ID"] <- data.subset[, "Sample_ID"]
        data[coord, "TechReplicate"] <- data.subset[, "TechReplicate"]
        data[coord, "Lysis"] <- data.subset[, "Lysis"]
        data[coord, "Beads"] <- data.subset[, "Beads"]
        data[coord, "Loading_buffer"] <- data.subset[, "Loading_buffer"]
        data[coord, "Engine"] <- data.subset[, "Engine"]
        data[coord, "Beads_ratio"] <- data.subset[, "Beads_ratio"]
        data[coord, "Statring_material"] <- data.subset[, "Statring_material"]
        data[coord, "Volume"] <- data.subset[, "Volume"]
        data[coord, "Beads_amount"] <- data.subset[, "Beads_amount"]


      }
      add.colnames <- colnames(data)[!(colnames(data) %in%
                                         c("Sample_ID", "peptide", "modified_peptide", "protein","S_167", "T_181", "Y_243",
                                           "DECOY", "PHOSPHO", "Count_Phospho", "Delocalized",
                                           "TechReplicate", "Lysis", "Beads", "Loading_buffer", "Engine", "Beads_ratio",
                                           "Statring_material","Volume", "Beads_amount"))]
      data <- data[, c("Sample_ID", "protein","peptide", "modified_peptide", "S_167", "T_181", "Y_243",
                       "DECOY", "PHOSPHO", "Count_Phospho", "Delocalized",
                       "TechReplicate", "Lysis", "Beads", "Loading_buffer", "Engine","Beads_ratio",
                       "Statring_material","Volume", "Beads_amount",
                       add.colnames)]
      return(data)
    }
  }



annotation.file <- "Study_design_CoOmXT.txt"
Study_design <- read.delim2(file.path(getwd(), annotation.file), dec=".", sep ="\t", header=TRUE)
data_CoOmXT$spectrum <- gsub(pattern="*~.*",replacement="", data_CoOmXT$spectrum )
data_CoOmXT <- annotation.iportal(data_CoOmXT, Study_design)


data_iportal <- data_CoOmXT
n_data_iportal <- subset(data_iportal, select = c(Sample_ID, protein, peptide, modified_peptide, PHOSPHO, DECOY, Delocalized, Engine))
n_data_maxquant <- subset(data_mq_all, select = c(Sample_ID, protein, peptide, modified_peptide, PHOSPHO, DECOY, Delocalized, Engine))
n_data_all <- do.call("rbind", list(n_data_iportal, n_data_maxquant))
n_data_all <- unique(n_data_all)

# subsetting the combined data into subsets
sub_PEPIDE <- subset(n_data_all, DECOY == FALSE)
sub_DECOY <- subset(n_data_all, DECOY == TRUE)
sub_PHOSPHO <-subset(n_data_all, DECOY == FALSE & PHOSPHO == TRUE)
sub_PHOSPHO_Protein <- subset(n_data_all, DECOY == FALSE & PHOSPHO == TRUE)
sub_PHOSPHO_Protein <- unique(subset(sub_PHOSPHO_Protein, select = c(Sample_ID, protein, Engine)))
sub_DELOCALIZED <- subset(n_data_all, DECOY == FALSE & PHOSPHO == TRUE)
sub_DELOCALIZED <- unique(subset(sub_DELOCALIZED, select = c(Sample_ID, protein, Engine, Delocalized)))

# Various plots to give an overview over the data
p1 <-ggplot(sub_PEPIDE, aes(factor(Sample_ID))) +
  geom_bar(aes(fill =Sample_ID)) +
  facet_wrap( ~Engine, ncol = 2) +
  ggtitle("Identified phosphopeptides for the iPortal TPP and MaxQuant") +
  ylim(0, 4350) +
  labs(x = "", y = "counts")+
  theme(
    axis.text.x = element_text(size = 8, angle = 45),
    axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 10),
    axis.title.y = element_text(size = 10),
    plot.title = element_text(size = 14),
    strip.text.x = element_text(size = 10))
plot(p1)

p2 <-ggplot(sub_PHOSPHO, aes(factor(Sample_ID))) +
  geom_bar(aes(fill =Sample_ID)) +
  facet_wrap( ~Engine, ncol = 2) +
  ggtitle("Identified unique phosphophopeptides for the iPortal and MaxQuant TPP") +
  ylim(0, 4350) +
  labs(x = "", y = "counts")+
```

```
  theme(
    axis.text.x = element_text(size = 8, angle = 45),
    axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 10),
    axis.title.y = element_text(size = 10),
    plot.title = element_text(size = 14),
    strip.text.x = element_text(size = 10))
plot(p2)

p3 <-ggplot(sub_DELOCALIZED, aes(factor(Sample_ID))) +
  geom_bar(aes(fill =Sample_ID)) +
  facet_wrap( ~Engine, ncol = 2) +
  ggtitle("Identified unique delocalized phosphopeptdies for the iPortal and MaxQuant TPP") +
  ylim(0, 3600) +
  labs(x = "", y = "counts")+
  theme(
    axis.text.x = element_text(size = 8, angle = 45),
    axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 10),
    axis.title.y = element_text(size = 10),
    plot.title = element_text(size = 14),
    strip.text.x = element_text(size = 10))
plot(p3)


p4 <-ggplot(sub_PHOSPHO_Protein, aes(factor(Sample_ID))) +
  geom_bar(aes(fill =Sample_ID)) +
  facet_wrap( ~Engine, ncol = 2) +
  ggtitle("Identified unique phosphoproteins for the iPortal and MaxQuant TPP") +
  ylim(0, 1800) +
  labs(x = "", y = "counts") +
  theme(
    axis.text.x = element_text(size = 8, angle = 45),
    axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 10),
    axis.title.y = element_text(size = 10),
    plot.title = element_text(size = 14),
    strip.text.x = element_text(size = 10))
plot(p4)

# Count function (adapted from the "Beads and buffer combinations" analysis)
count.species <- function(x, species=c()){

  if (species == "phospho_proteins") {
    x <- melt(x, id =c("protein", "Engine"))
    x$variable <- 1
    x <- dcast(x, value ~ protein + Engine, value.var = "variable")
  } else if (species == "peptide"){
    ## is a rather complex fragment of code ...
    ## in fact, there are some peptides, which have different modifications so therefore we need to combine the once with modifcation,
    ## and the once which do not have any modifaction in one row, to really count the number of detected peptides, because it can also be,
    ## so that we account for peptides with the phosphorylation on different sites.

    x <- subset(x, select = c(Sample_ID, peptide, Engine, protein, modified_peptide))
    index <- x$modified_peptide == is.na(TRUE)
    index[is.na(index)] <- TRUE
    x$modified_peptide[index] <- (x$peptide[index])
    x <- subset(x, select = c(Sample_ID, modified_peptide, Engine, protein))
    x <- melt(subset(x, select = c(Sample_ID, modified_peptide, Engine)), id =c("modified_peptide", "Engine"))
    x <- unique(x)
    x$variable <- 1
    x <- dcast(x, value ~ modified_peptide + Engine, value.var = "variable")
  } else if (species == "phospho_peptide") {

    ## produces the same result as in the earlier analysis. I am not so sure about the unique. The issue is,
    ## the number of phosphopeptide do not alter, if you take the Proteins into account and therefore I would rather suggest not to unique.
    ## In the protein list in the peptides.tsv list are only proteotpyic peptides (checked it in the excel)
    x <- subset(x, select = c(Sample_ID, modified_peptide, Engine, protein))
    x <- melt(subset(x, select = c(Sample_ID, modified_peptide, Engine)), id =c("modified_peptide", "Engine"))
    x <- unique(x)
    x$variable <- 1
    x <- dcast(x, value ~ modified_peptide + Engine, value.var = "variable")

  } else if (species == "delocalized") {
    ## same about the delocalized as for the peptides, if the protein is took into account, there is not one entrey removed. Therefore we should stick
    ## here also to the non reduced one.
    x <- melt(subset(x, select = c(Sample_ID, Delocalized, Engine)), id =c("Delocalized", "Engine"))
    x <- unique(x)
    x$variable <- 1
    x <- dcast(x, value ~ Delocalized + Engine, value.var = "variable")
  } else if (species == "decoy"){
    x <- sub_DECOY
    x <- melt(subset(x, select = c(Sample_ID, protein, Engine)), id =c("protein", "Engine"))
    x <- unique(x)
    x$variable <- 1
    x <- dcast(x, value ~ protein + Engine, value.var = "variable")
  }


  data <- x
  colname <- c()
  colname <- species
  data$CoOmXT <- apply(data[,grep("*_CoOmXT", colnames(data))], 1, function(x)sum(x, na.rm = TRUE))
  data$MaxQuant <- apply(data[,grep("*_MaxQuant", colnames(data))], 1, function(x)sum(x, na.rm = TRUE))
  data <- melt(subset(data, select =c(value, CoOmXT, MaxQuant)), id="value")
  colnames(data)[3] <- colname[1]
  return(data)
}

# count the different species
count_sub_PHOSPHO_Protein <- count.species(sub_PHOSPHO_Protein, species = "phospho_proteins")
count_sub_PEPIDE <- count.species(sub_PEPIDE, species = "peptide")
count_sub_PHOSPHO <- count.species(sub_PHOSPHO, species = "phospho_peptide")
count_sub_DELOCALIZED <- count.species(sub_DELOCALIZED, species = "delocalized")
count_sub_DECOY <- count.species(sub_DECOY, species = "decoy")

## merge function with reduce from http://stackoverflow.com/questions/14096814/r-merging-a-lot-of-data-frames
count_merge <- Reduce(function(x,y) merge(x,y, all=TRUE), list(count_sub_PHOSPHO_Protein, count_sub_PEPIDE, count_sub_PHOSPHO, count_sub_DELOCALIZED,
count_sub_DECOY))
count_merge <- count_merge[order(count_merge$variable),]
count_merge$enrichment <- round(count_merge$phospho_peptide/count_merge$peptide, digits = 2)
colnames(count_merge)[colnames(count_merge) == "value"] <- "Sample_ID"
colnames(count_merge)[colnames(count_merge) == "variable"] <- "Engine"
```

```
annotation.file <- "Study_design_plotting.txt"
Study_design <- read.delim2(file.path(getwd(), annotation.file), dec=".", sep ="\t", header=TRUE)
count_merge <- merge(count_merge, Study_design, by = "Sample_ID")
levels(count_merge$Engine) <- c("iPortal", "MaxQuant")

# Represent the data with various plots
p5 <-ggplot(count_merge, aes(x = factor(Sample_ID), y = (enrichment)*100)) +
  geom_bar(stat = "identity", aes(fill =Sample_ID)) +
  facet_wrap( ~Engine, ncol = 2) +
  ggtitle("Phosphoenrichment in percentages per sample \n for the iPortal and MaxQuant TPP") +
  ylim(0, 100) +
  geom_text(aes(label=(enrichment)*100), position=position_dodge(width=0.9), vjust=-0.25, size =2.5) +
  labs(x = "", y = "counts")+
  guides(fill=guide_legend(ncol = 2 )) +
  theme(
    axis.text.x = element_text(size = 12,  angle = 45, hjust = 1, color="black"),
    axis.text.y = element_text(size = 12),
    axis.title = element_text(size = 12),
    axis.title.y = element_text(size = 12),
    plot.title = element_text(size = 16),
    legend.text = element_text(size = 14),
    legend.title = element_text(size=15, face="bold"),
    strip.text.x = element_text(size = 12))
plot(p5)

# use subsets for plotting
df_plot <- subset(count_merge, subset = Statring_material == 1.0 & Volume == 800)

p6 <-ggplot(df_plot, aes(factor(x = Beads_amount), y = phospho_peptide)) +
  geom_point(aes(colour = factor(TechReplicate)),size = 5) +
  geom_text(aes(label=Sample_ID),hjust=0.9, vjust=2.0, size=5) +
  facet_wrap( ~Engine, ncol = 4) +
  ggtitle("Variation of the beads to starting material ratio \n with constant starting material of 1 mg ") +
  scale_colour_discrete(name="Technical Replicate",
                        breaks=c("1", "2"),
                        labels=c("Replicate 1", "Replicate 2")) +
  labs(x = "Beads ratio", y = "Total number of identified phosphopeptides")+
  scale_x_discrete(breaks=c("3", "5", "10", "20"),
                   labels=c("3:1", "5:1", "10:1", "20:1")) +
  theme(
    axis.text.x = element_text(size = 14, color = "black"),
    axis.text.y = element_text(size = 14),
    axis.title = element_text(size = 14),
    axis.title.y = element_text(size = 14),
    plot.title = element_text(size = 18),
    legend.text = element_text(size = 14),
    legend.title = element_text(size= 14, face="bold"),
    strip.text.x = element_text(size = 14))
plot(p6)

df_plot <- subset(count_merge, subset = Sample_ID != ("FF161"))
df_plot <- subset(df_plot, subset = Sample_ID != ("FF150"))


p7 <-ggplot(subset(df_plot, Beads_ratio == "3:1" ), aes(factor(x = Statring_material), y = phospho_peptide)) +
  geom_point(aes(colour = factor(TechReplicate), shape =factor(Engine)),size = 5) +
  ggtitle("Variation of the amount of starting material at a \n constant beads to starting material ratio of 3:1 ") +
  scale_colour_discrete(name="Technical Replicate",
                        breaks=c("1", "2"),
                        labels=c("Replicate 1", "Replicate 2")) +
  scale_shape_discrete(name="Engine",
                       breaks=c("iPortal", "MaxQuant"),
                       labels=c("iPortal", "MaxQuant")) +
  labs(x = "protein starting material [mg]", y = "Total number of identified phosphopeptides")+
  theme(
    axis.text.x = element_text(size = 14, color = "black"),
    axis.text.y = element_text(size = 14),
    axis.title = element_text(size = 14),
    axis.title.y = element_text(size = 14),
    plot.title = element_text(size = 18),
    legend.text = element_text(size = 14),
    legend.title = element_text(size= 14, face="bold"),
    strip.text.x = element_text(size = 14))
plot(p7)


df_plot <- subset(df_plot, subset = Statring_material <= 1.0)
df_plot <- subset(df_plot, subset = Beads_ratio == "3:1")

p8 <-ggplot(df_plot, aes((x = Volume), y = phospho_peptide)) +
  geom_point(aes(colour = factor(TechReplicate), shape =factor(Statring_material)),size = 3) +
  ggtitle("Influence of starting amount concentration to the phospho enrichment \n for the 0.5 mg and 1 mg starting material samples with a beads ratio of
3:1") +
  scale_colour_discrete(name="Technical Replicate",
                        breaks=c("1", "2"),
                        labels=c("Replicate 1", "Replicate 2")) +
  scale_shape_discrete(name="Starting material",
                       breaks=c("0.5", "1"),
                       labels=c("0.5 mg", "1 mg")) +
  scale_x_continuous(limit = c(200,800)) +
  scale_y_continuous(limits = c(2000, 3500)) +
  labs(x = "Volume [mL]", y = "Total number of phosphopeptides")+
  theme(
    axis.text.x = element_text(size = 12, color="black"),
    axis.text.y = element_text(size = 12),
    axis.title = element_text(size = 12),
    axis.title.y = element_text(size = 12),
    plot.title = element_text(size = 18),
    legend.text = element_text(size = 14),
    legend.title = element_text(size= 15, face="bold"),
    strip.text.x = element_text(size = 10))
plot(p8)

# plot intensities of the MaxQuant LFQ result
data_sum_MQ_phospho <- as.data.frame(colSums(Filter(is.numeric, data_mq_phospho)))
data_sum_MQ_phospho_c <- as.data.frame(colSums(Filter(is.numeric, data_mq_phospho_Hela)))
colnames(data_sum_MQ_phospho)[1] <- "Intensity_sum"
colnames(data_sum_MQ_phospho_c)[1] <- "Intensity_sum"
data_sum_MQ_phospho_all <- rbind(data_sum_MQ_phospho_c, data_sum_MQ_phospho)
data_sum_MQ_phospho_all$Sample_ID <- rownames(data_sum_MQ_phospho_all)

annotation.file <- "Study_design_phospho_intensity.txt"
Study_design <- read.delim2(file.path(getwd(), annotation.file), dec=".", sep ="\t", header=TRUE)
data_sum_MQ_phospho <- merge(data_sum_MQ_phospho_all, Study_design, by = "Sample_ID")
```

```
p9 <-ggplot(data_sum_MQ_phospho, aes((x = x_annotation), y = Intensity_sum)) +
  geom_point(aes(colour = factor(TechReplicate), shape =factor(Statring_material)),size = 3) +
  ggtitle("Sum of intensities of all tested conditions for the \n label free quantifcation result of MaxQuant") +
  scale_colour_discrete(name="Technical Replicate",
                        breaks=c("1", "2"),
                        labels=c("Replicate 1", "Replicate 2")) +
  scale_shape_discrete(name="Statring_material",
                       breaks=c("0.5", "1", "2","4"),
                       labels=c("0.5 mg protein", "1 mg protein", "2 mg protein", "4 mg protein")) +
  scale_x_discrete(limits=c("3:1_0.5mg_buffer_200mL","3:1_0.5mg_buffer_400mL","3:1_0.5mg_buffer_400mL_C",
                            "3:1_0.5mg_buffer_800mL", "3:1_1.0mg_buffer_400mL", "3:1_2.0mg_buffer_800mL",
                            "3:1_4.0mg_buffer_800mL", "3:1_1.0mg_buffer_800mL","5:1_1.0mg_buffer_800mL",
                            "10:1_1.0mg_buffer_800mL", "20:1_1.0mg_buffer_800mL")) +
  scale_y_log10() +
  labs(x = "", y = "log10(sum(intensities))")+
  theme(
    axis.text.x = element_text(size = 8, angle = 45, hjust= 1),
    axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 8),
    axis.title.y = element_text(size = 8),
    plot.title = element_text(size = 10),
    legend.text = element_text(size = 6),
    legend.title = element_text(size= 6, face="bold"),
    strip.text.x = element_text(size = 8))
plot(p9)

write.table(count_merge, file="beads_ratio_amount_starting.txt", quote = FALSE, row.names=FALSE, sep="\t")


#########################################################################################################################################################
#                                                                                                                                                       #
#                 "SWATH2stats for OpenSWATH data - one example"                                                                                        #
#                                                                                                                                                       #
# Author: Peter Blattmann, Moritz Heusel (2015 - bioconductor; needs the newest R-version for fully function)                                           #
# Date: various days for each OpenSWATH dataset                                                                                                         #
# Summary: The SWATH2stats package was used to i) annotate the OpenSWATH output datasets ii) to filter them iii) to write out the data         #
#          in a format, such they can be used to for the subsequent analysis in mapDIA                                                                  #
#########################################################################################################################################################

# Is an adappted script, which can aslo be used for further processing of other libraries
setwd("Y:\\20160324_PTM-itestportal_George_comparison/")

# Clear the workspace before starting
rm(list=ls())

# load the libraries
library(SWATH2stats)
library(reshape2)
library(ggplot2)

# Impurt the result from the openSWATH analysis from iPortal
file.name <- "E1603101059_feature_alignment.tsv"
data <- data.frame(read.table(file.path("Y:\\html\\openBIS\\20160310135139810-1152224\\", file.name), sep = '\t', header= TRUE))

# annotating the data
nlevels(factor(data$align_origfilename))
levels(factor(data$align_origfilename))
annotation.file <- "Study_design.txt"
Study_design <- read.delim2(file.path(getwd(), annotation.file), dec=".", sep ="\t", header=TRUE)

# reduce the amount of rows which are necessary for MStats & mapDIA
# delete the iRT peptides
data <-data[grep("iRT", data$ProteinName, invert=TRUE),]

# Annotation of the data
data.annotated <-sample_annotation(data, Study_design)
head(unique(data.annotated$ProteinName))

data.FDR <- data.frame(read.table(file.path("Y:\\html\\openBIS\\20160310135139810-1152224\\", file.name), sep = '\t', header= TRUE))
data.FDR <- sample_annotation(data.FDR, Study_design)

# FDR overview and visualization
# For each the dataset anohter FFT is used (described in the mehtods chapter)

data.annotated$ProteinName <- gsub("Subgroup_[0-9]_[0-9];", "", data.annotated$ProteinName)
assess_decoy_rate(data.annotated)
overall_fdr_table <-assess_fdr_overall(data.annotated, FFT=0.29, output = "Rconsole")
byrun_fdr_cube <- assess_fdr_byrun(data.annotated, FFT = 0.29, output ="Rconsole")

# we used an target FDR of 0.01 as this is the standard in Proteomics analysis
#
# We get a mscore value and it rights down the m-score cutoff uand teh FDR for the assay
mscore4assayfdr(data = data.annotated, FFT = 0.29, fdr_target = 0.01)
mscore4protfdr(data = data.FDR, FFT = 0.29, fdr_target =  0.01)
mscore4pepfdr(data = data.FDR, FFT = 0.29, fdr_target =  0.01)

# For Filtering the data, we used m-score cutoff, due to the fact, that we have a second criteria the caluculated FDR
# can be loared to achive a 0.01 peptide FDR, which we aimed for;
# this was in every data analysis, depending on the experiment and the library used for the extraction different
# for the Mouse reference population a phosphopeptide transition had to be deteced in at least 60% of the samples
# to be not removed; in the aging dataset a phosphopeptide hat to be quantified in at least 2 replicates
# We allways used different m-score cutoffs, to achive the 0.01 peptide FDR

data.filtered.mscore <- filter_mscore_condition(data= data.annotated, 0.0035, n.replica = 2, rm.decoy = FALSE)
assess_decoy_rate(data.filtered.mscore)

overall_fdr_table <-assess_fdr_overall(data.filtered.mscore, FFT=0.29, output = "Rconsole")

# after controlling how the filterd dataset looks like (FDR and m-scroe); we can than also
# deleting the DECOYs from the dataset and estimating the m-score againg
data.filtered.mscore <- filter_mscore_condition(data= data.annotated, 0.0035, n.replica = 2, rm.decoy = TRUE)

# export the Data for further processing;
# The data are filtered with the function data.filtered.mscore; first it is controlled
# for which amounts, with the reduction of Decoys the number of true hits are reduced.
# The third paramter filters only the peaks which are found in at least two replicates;

data.transition <- disaggregate(data.filtered.mscore)
write.csv(data.transition, file ="transition_level_output_PTM_SWATH.csv", row.names=FALSE, quote=FALSE)

data.transition <-data.frame(read.csv("transition_level_output_PTM_SWATH.csv"))

# Convert to Msstats
```

```
MSstats.input <- convert4MSstats(data.transition)
quantData <- dataProcess(MSstats.input)


# convert to MapDIa
# to have unique protein name, the sequenze is coppied to the protein
# because in the end, we want to see if a single peptide is different phosphorylated comapred to
# another (e.g. regulation through phosphorylation)

mapDIA.input <-convert4mapDIA(data = data.transition, RT=TRUE)

# only Protein Name merging if the data set is a phosphopeptide dataset

mapDIA.input$ProteinName <- paste(mapDIA.input$ProteinName, mapDIA.input$PeptideSequence, sep="_")
head(mapDIA.input)


write.table(mapDIA.input, file="mapDIA_unfiltered_PTM_SWATH.txt", quote = FALSE, row.names=FALSE, sep="\t")


#########################################################################################################################################################
#                                                                                                                                                       #
#                 "Refine a phospho-SWATH-MS library"                                                                                                   #
#                                                                                                                                                       #
# Author: Peter Blattmann & Fabian Frommelt                                                                                                             #
# Date: 01.07.2016                                                                                                                                      #
# Summary: Used to refine a SWATH assay library which was constructed with the OpenSWATH library construction workflow (Schubert, O.T., et al.,#
#          Nature Protolocs. 2015) and site localization scoring with LuciPHOr2 (Fermin, D., et al., Bioinformatics, 2015)                              #
#########################################################################################################################################################

# Analyse TSV library of the "Spectrast"-workflow (.tsv file format)
# converting to a TraML file

setwd("Y:\\20160125_phospho_library_mouse_liver/")
getwd()

library(gtools)

# load the saved functions which are storred in the folder "Functions"
sapply(list.files(pattern="[.]R$", path="Functions/R/", full.names=TRUE), source);

SpecLib_table <- read.table(file.path("Y:","20160125_phospho_library_mouse_liver","SpecLib_cons_openswath.tsv"),header=TRUE,quote = "")

reshape.SpecLib_cons_openswath <-
  function (x){
    x <- x[grep("DECOY", x$ProteinName, invert = TRUE),]
    k  <- x[grep("\\UniMod\\:21\\)", x$transition_group_id), ] #phosphorylation
    k1 <- x[grep("\\/iRT_protein", x$ProteinName),]
    #k1 <-x[grep("\\UniMod\\:4\\)", x$transition_group_id, invert = TRUE),] #after C
    #k <- x[grep("\\UniMod\\:35\\)", x$transition_group_id, invert = TRUE),] #after M
    y <-rbind(k,k1)
    SpecLib_table <- y
    return(SpecLib_table)
  }

SpecLib_table <- reshape.SpecLib_cons_openswath(SpecLib_table)

unique(subset(SpecLib_table, ProteinName == "Subgroup_0_1/iRT_protein")[,c("Tr_recalibrated", "PeptideSequence", "ProteinName")])

# Assess and change retention times
# read in iRT retention times
iRT <- read.table(file.path("Y:","analysis","irtkit_minus_LFL.txt"))
colnames(iRT)<- c("Peptide", "RT")

# observe difference of retention time in library versus consensus table
merge(iRT, unique(SpecLib_table[, c("PeptideSequence", "Tr_recalibrated")]), by.x="Peptide", by.y="PeptideSequence", all.x=TRUE)

# change retention time to the times from consensus table
for(i in iRT$Peptide){
  SpecLib_table[SpecLib_table$PeptideSequence == i, "Tr_recalibrated"] <- iRT[iRT$Peptide == i, "RT"]
}

# control to see that retention times match now
unique(SpecLib_table[grep("iRT_protein", SpecLib_table$ProteinName), c("PeptideSequence", "Tr_recalibrated")])



# remove Subgroup_0_1/
SpecLib_table$ProteinName <- gsub("Subgroup_0_1/", "", SpecLib_table$ProteinName)
SpecLib_table$UniprotID <-gsub("Subgroup_0_1/", "", SpecLib_table$UniprotID)

write.table(x = SpecLib_table, file="SpecLib_cons_openswath_control.tsv", row.names = FALSE, quote = FALSE, sep="\t")



#########################################################################################################################################################
#                                                                                                                                                       #
#                 "Comparison of the three SWATH-MS librariers"                                                                                         #
#                                                                                                                                                       #
# Author: Fabian Frommelt                                                                                                                               #
# Date: 20.04.2016 (last adaption)
# Summary: Statistical comparison of the OpenSWATH/PTM libraries with the phospho-SWATH MS library. The data represented by various plots      #
#                                                                                                                                                       #
#########################################################################################################################################################


# Analysis of the three constructed SWATH assay librariers for phosphopeptide enriched mouse livert tissue samples.
# 1) 31 DDA measurements, openSWATH/ PTM analysis without filtering
# 2) 31 DDA measurements, openSWATH/ PTM analysis from the filtered data as the 3rd analysis is also constructed
#    so the only different to the third library is the library construction
# 3) Phospho-SWATH Library from 31 DDA measurements; filtered with LuciPHOr, extracted without MaxQuant
# Analysis of phospho enriched samples 4 biological samples, 3 replicates / in total 12 samples
# Aim: To comapre the different library constrcution methods and try to find out if one of the methods is doing
#      a better shop. Also if  there are significant differences, look if this differences are
#      due to the filtering at the LuciPHOr2 threshhold, where peptides are sorted out.


# set the working directory to the Elite LFQ folder
setwd("Y:\\20160324_PTM-itestportal_George_comparison/")

# Clear the workspace before starting
rm(list=ls())

# load libraries
```

```
library(data.table) # for renaming
library(gtools)
library(ggplot2)
library(Peptides)
library(VennDiagram)
library(stringr)

# load the three data sets
file.name <- "peptide_level.txt"
data_phospho_SWATH <- read.table(file.path("Y:\\20160324_PTM-itestportal_George_comparison\\filtered_phospho_SWATH_standard_without_requant\\", file.name),
                                 header=TRUE, sep="\t", fill=TRUE, stringsAsFactors = FALSE)
data_openSWATH_PTM_fil <- read.table(file.path("Y:\\20160324_PTM-itestportal_George_comparison\\filtered_SWATH_PTM\\", file.name), header=TRUE, sep="\t",
                                     fill=TRUE, stringsAsFactors = FALSE)
data_openSWATH_PTM_unfil <- read.table(file.path("Y:\\20160324_PTM-itestportal_George_comparison\\unfiltered_SWATH_PTM\\", file.name), header=TRUE, sep="\t",
                                       fill=TRUE, stringsAsFactors = FALSE)

# 1) Data reshaping and calculating of statistical data for each of the datasets

# Change the phosphosite annotation
phosphorylation.annotation <- function(data){
  # Function changes annotation and creates a column in which all Proteins are written only with
  # their identifier
  x <- data
  x[,grep("BXD\\.RHO\\.[[:digit:]]{4}\\_*", colnames(x))] <- sapply(x[,grep("BXD\\.RHO\\.[[:digit:]]{4}\\_*", colnames(x))],as.numeric)
  x <- as.data.frame(sapply(x,gsub,pattern="*\\(UniMod_21\\)*",replacement="\\(Phospho\\)"), stringsAsFactors = FALSE)
  x <- as.data.frame(sapply(x,gsub,pattern="*\\(UniMod_35\\)*",replacement="\\(Oxidation\\)"), stringsAsFactors = FALSE)
  x <- as.data.frame(sapply(x,gsub,pattern="*\\(UniMod_4\\)*",replacement="\\(Carbamidomethyl\\)"), stringsAsFactors = FALSE)
  x$Protein_alone <- x$Protein
  x$Protein_alone <- sapply(x$Protein_alone,gsub,pattern="Subgroup\\_[[:digit:]]{1}\\_[[:digit:]]{1}\\/",replacement="")
  x$Protein_alone <- gsub("\\(Phospho\\)", replacement = "", x = x$Protein_alone)
  x$Protein_alone <- gsub("\\(Oxidation\\)", replacement = "", x = x$Protein_alone)
  x$Protein_alone <- gsub("\\(Carbamidomethyl\\)", replacement = "", x = x$Protein_alone)
  x$Protein_alone <- sub("\\_([[:alnum:]])+","", x$Protein_alone)
  x$Count_Phospho <- sapply("(Phospho)", str_count, string =x$Peptide)
  x$Deloc <- x$Peptide
  x$Deloc <- sapply(x$Deloc,gsub,pattern="*\\(Phospho\\)*",replacement="")
  x$Deloc <- sapply(x$Deloc,gsub,pattern="*\\(Oxidation\\)*",replacement="")
  x$Delocalized <- paste(x$Deloc, x$Count_Phospho, sep="_P")
  x <- subset(x, select = -c(Deloc))
  x$Delocalized <- gsub(x$Delocalized, pattern = "NA_PNA", replacement = NA )
  x$Delocalized <- gsub(x$Delocalized, pattern = "\\_P0", replacement = "" )
  return(x)
}

data_openSWATH_PTM_fil <- phosphorylation.annotation(data = data_openSWATH_PTM_fil)
data_openSWATH_PTM_unfil <- phosphorylation.annotation(data =data_openSWATH_PTM_unfil)
data_phospho_SWATH <- phosphorylation.annotation(data = data_phospho_SWATH)

# calculate statistical data for the Replicates
Statistics_mean_sd_CV_Replciate <-
  function (data = dataframe()){
    m.data <- melt(data, id=c("Protein", "Peptide", "nFragment", "Delocalized", "Count_Phospho", "Protein_alone"))
    m.data$Replicate <- paste("Rep", gsub(".*_([[:digit:]])$", "\\1", m.data$variable))
    m.data$Mouse <- gsub("_[[:digit:]]$", "", m.data$variable)
    m.data$value <- as.numeric(m.data$value)
    data <- dcast(m.data, Protein + Peptide + Mouse  ~ Replicate)
    data$mean.rep <- apply(data[,grep("Rep", colnames(data))], 1, function(x)mean(x, na.rm = TRUE))
    data$sd.rep <- apply(data[,grep("Rep", colnames(data))], 1, function(x)sd(x, na.rm = TRUE))
    data$CV.rep <- apply(data[, grep("Rep", colnames(data))], 1, function(x) sd(x,na.rm = TRUE)/mean(x, na.rm = TRUE))
    data$Var.rep <- apply(data[,grep("Rep", colnames(data))], 1, function(x)var(x, y=NULL, na.rm = TRUE))
    return(data)
  }

data_phospho_SWATH.pep_Rep <- Statistics_mean_sd_CV_Replciate(data_phospho_SWATH)
data_openSWATH_PTM_fil.pep_Rep <- Statistics_mean_sd_CV_Replciate(data_openSWATH_PTM_fil)
data_openSWATH_PTM_unfil.pep_Rep <- Statistics_mean_sd_CV_Replciate(data_openSWATH_PTM_unfil)

# calculate statistical data for each data set over all samples
Statistics_mean_sd_CV_total <-
  function (data = dataframe()){
    data[,grep("BXD.RHO*", colnames(data))] <- sapply(data[,grep("BXD.RHO*", colnames(data))],as.numeric)
    data$mean.total <- apply(data[,grep("BXD.RHO*", colnames(data))], 1, function(x)mean(x, na.rm = TRUE))
    data$sd.total <- apply(data[,grep("BXD.RHO*", colnames(data))], 1, function(x)sd(x, na.rm = TRUE))
    data$CV.total <- apply(data[, grep("BXD.RHO*", colnames(data))], 1, function(x) sd(x, na.rm = TRUE)/mean(x, na.rm = TRUE))
    data$Var.total <- apply(data[,grep("BXD.RHO*", colnames(data))], 1, function(x)var(x, y=NULL, na.rm = TRUE))
    return(data)
  }

data_phospho_SWATH.pep_Total <- Statistics_mean_sd_CV_total(data = data_phospho_SWATH)
data_openSWATH_PTM_fil.pep_Total <- Statistics_mean_sd_CV_total(data = data_openSWATH_PTM_fil)
data_openSWATH_PTM_unfil.pep_total <- Statistics_mean_sd_CV_total(data = data_openSWATH_PTM_unfil)


# 2) Correlation for the intensities in each data set

cor.function <- function (data =dataframe()) {
  pearson.cor <- cor(data[,3:14], use="pairwise.complete.obs", method="pearson")
  pearson.cor[lower.tri(pearson.cor)] <- NA

  spearman.cor <- cor(data[,3:14], use="pairwise.complete.obs", method="spearman")
  spearman.cor[upper.tri(spearman.cor, diag = TRUE)] <- NA

  data.plot <- rbind(melt(pearson.cor), melt(spearman.cor))
  data.plot <- data.plot[!is.na(data.plot$value),]
  p <- (ggplot(data.plot, aes(x=Var2, y=Var1, fill=value)) + geom_tile()
        + scale_fill_gradient(low = "white", high="red", name="Correlation\n[R or rho]")
        + xlab("") + ylab("")
        + labs(title="Correlation between samples: Pearson  (upper triangle) and Spearman correlation (lower triangle)")
        + geom_text(aes(fill = data.plot$value, label = round(data.plot$value, digits= 2)))
        + theme(plot.title = element_text(hjust = 0, vjust = 1),
                axis.text.x = element_text(size = 10, angle = 45, hjust = 1)))

  print(p)
}

cor.function(data_phospho_SWATH.pep_Total)
cor.function(data_openSWATH_PTM_fil.pep_Total)
cor.function(data_openSWATH_PTM_unfil.pep_total)

cor.fc.total.all <- function(data_1 =dataframe(), data_2 = dataframe(), data_3 = dataframe()) {

  data_1 <- data_1[, c(2,19)]

  data_2 <- data_2[, c(2,19)]
```

```
  data_3 <- data_3[, c(2,19)]

  data_merge <- merge(data_1, data_2, by = "Peptide", all = TRUE)
  data_merge <- merge(data_merge, data_3, by = "Peptide", all = TRUE)
  colnames(data_merge)[colnames(data_merge) == "mean.total.x"] <- "mean.total.openSWATH_PTM_unfil"
  colnames(data_merge)[colnames(data_merge) == "mean.total.y"] <- "mean.total.openSWATH_PTM_fil"
  colnames(data_merge)[colnames(data_merge) == "mean.total"] <-   "mean.total.phospho-SWATH"

  pearson.cor <- cor(data_merge[,2:4], use="pairwise.complete.obs", method="pearson")
  pearson.cor[lower.tri(pearson.cor)] <- NA

  spearman.cor <- cor(data_merge[,2:4], use="pairwise.complete.obs", method="spearman")
  spearman.cor[upper.tri(spearman.cor, diag = TRUE)] <- NA

  data.plot <- rbind(melt(pearson.cor), melt(spearman.cor))
  data.plot <- data.plot[!is.na(data.plot$value),]
  p <- (ggplot(data.plot, aes(x=Var2, y=Var1, fill=value)) + geom_tile()
        + scale_fill_gradient(low = "white", high="red", name="Correlation\n[R or rho]")
        + xlab("") + ylab("")
        + labs(title="Correlation between samples: Pearson  (upper triangle) and Spearman correlation (lower triangle)")
        + geom_text(aes(fill = data.plot$value, label = round(data.plot$value, digits= 2)))
        + theme(plot.title = element_text(hjust = 0, vjust = 1),
                axis.text.x = element_text(size = 10, angle = 45, hjust = 1)))

  print(p)
}

cor.fc.total.all(data_1 = data_openSWATH_PTM_unfil.pep_total, data_2 = data_openSWATH_PTM_fil.pep_Total, data_3 = data_phospho_SWATH.pep_Total)

# 3) Plot the Coefficient of Variation as violin plot for the three phospho-SWATH librariers

# construct a Dataset for plotting the Variance values in a violin plot
sub.data_phospho_SWATH.pep_Total <- subset(data_phospho_SWATH.pep_Total, select = c("Peptide", "CV.total"))
sub.data_phospho_SWATH.pep_Total <- melt(sub.data_phospho_SWATH.pep_Total, id = "Peptide")
sub.data_phospho_SWATH.pep_Rep <- subset(data_phospho_SWATH.pep_Rep, select = c("Peptide", "CV.rep"))
sub.data_phospho_SWATH.pep_Rep <- melt(sub.data_phospho_SWATH.pep_Rep, id ="Peptide")

sub.data_openSWATH_PTM_fil.pep_Total <- subset(data_openSWATH_PTM_fil.pep_Total, select = c("Peptide", "CV.total"))
sub.data_openSWATH_PTM_fil.pep_Total <- melt(sub.data_openSWATH_PTM_fil.pep_Total, id = "Peptide")
sub.data_openSWATH_PTM_fil.pep_Rep <- subset(data_openSWATH_PTM_fil.pep_Rep, select = c("Peptide", "CV.rep"))
sub.data_openSWATH_PTM_fil.pep_Rep <- melt(sub.data_openSWATH_PTM_fil.pep_Rep, id ="Peptide")

sub.data_openSWATH_PTM_unfil.pep_Total <- subset(data_openSWATH_PTM_unfil.pep_total, select = c("Peptide", "CV.total"))
sub.data_openSWATH_PTM_unfil.pep_Total <- melt(sub.data_openSWATH_PTM_unfil.pep_Total, id = "Peptide")
sub.data_openSWATH_PTM_unfil.pep_Rep <- subset(data_openSWATH_PTM_unfil.pep_Rep, select = c("Peptide", "CV.rep"))
sub.data_openSWATH_PTM_unfil.pep_Rep <- melt(sub.data_openSWATH_PTM_unfil.pep_Rep, id ="Peptide")


# annotation of the data
sub.data_phospho_SWATH.pep_Total$variable <- gsub("CV.total", "CV.all.phospho-SWATH", sub.data_phospho_SWATH.pep_Total$variable)
sub.data_phospho_SWATH.pep_Rep$variable <- gsub("CV.rep", "CV.rep.phospho-SWATH", sub.data_phospho_SWATH.pep_Rep$variable)

sub.data_openSWATH_PTM_fil.pep_Total$variable <- gsub("CV.total", "CV.all.openSWATH_fil", sub.data_openSWATH_PTM_fil.pep_Total$variable)
sub.data_openSWATH_PTM_fil.pep_Rep$variable <- gsub("CV.rep", "CV.rep.openSWATH_fil", sub.data_openSWATH_PTM_fil.pep_Rep$variable)

sub.data_openSWATH_PTM_unfil.pep_Total$variable <- gsub("CV.total", "CV.all.openSWATH_unfil", sub.data_openSWATH_PTM_unfil.pep_Total$variable)
sub.data_openSWATH_PTM_unfil.pep_Rep$variable <- gsub("CV.rep", "CV.rep.openSWATH_unfil", sub.data_openSWATH_PTM_unfil.pep_Rep$variable)


table.SWATH.CV <- rbind(sub.data_phospho_SWATH.pep_Total, sub.data_phospho_SWATH.pep_Rep,
                        sub.data_openSWATH_PTM_fil.pep_Total, sub.data_openSWATH_PTM_fil.pep_Rep,
                        sub.data_openSWATH_PTM_unfil.pep_Total, sub.data_openSWATH_PTM_unfil.pep_Rep)


table.SWATH.CV$label <- factor(table.SWATH.CV$variable, c("CV.all.phospho-SWATH", "CV.rep.phospho-SWATH", "CV.all.openSWATH_fil", "CV.rep.openSWATH_fil",
                                                          "CV.all.openSWATH_unfil", "CV.rep.openSWATH_unfil"))

p <- ggplot(table.SWATH.CV, aes(factor(label), value)) +
  geom_violin(scale="area") +
  stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median,
               geom = "crossbar", width = 0.5) +
  #scale_y_continuous(trans="log10") +
  #geom_boxplot(width=0.1) +
  labs(title=" Violin plots of the CV within the replicates and over all samples,\n analyzed for the three SWATH assay librarires" ,
       x="", y="[sd]/[mean of all or replicate Intensity]") +
  #scale_x_discrete(labels=c("CV.phospho.total", "CV.phospho.rep", "CV.total.cell.lysate")) +
  theme(axis.text = element_text(size = 10, colour = "black"), axis.title = element_text(size = 12),
        plot.title = element_text(size = 15),
        axis.text.x = element_text(hjust = 1, angle = 45))
print(p)

#### calculate media nand mode for the data
estimate_mode <- function(x) {
  d <- density(x)
  d$x[which.max(d$y)]
}

aggregate(table.SWATH.CV[,"value"], by=list(table.SWATH.CV$variable), FUN = function(x) median(x, na.rm=TRUE))
aggregate(table.SWATH.CV[,"value"], by=list(table.SWATH.CV$variable), FUN = "estimate_mode")

# 4) Correlation of all samples
# Try to correlate the Intensities of the technical replicates with each other

renaming_Rep_column_names <-
  function(data =dataframe()) {
    colnames(data)[colnames(data)=="Rep 1"] <- "Rep_1"
    colnames(data)[colnames(data)=="Rep 2"] <- "Rep_2"
    colnames(data)[colnames(data)=="Rep 3"] <- "Rep_3"
    data <- subset(data, select =c("Peptide", "Rep_1", "Rep_2", "Rep_3"))

    return(data)
  }

data_phospho_SWATH_1 <- renaming_Rep_column_names(data_phospho_SWATH.pep_Rep)
data_openSWATH_PTT_fil_1 <- renaming_Rep_column_names(data_openSWATH_PTM_fil.pep_Rep)
data_openSWATH_PTT_unfil_1 <- renaming_Rep_column_names(data_openSWATH_PTM_unfil.pep_Rep)

colnames(data_phospho_SWATH_1)[colnames(data_phospho_SWATH_1)=="Rep_1"] <- "Phospho_SWATH_Rep_1"
colnames(data_phospho_SWATH_1)[colnames(data_phospho_SWATH_1)=="Rep_2"] <- "Phospho_SWATH_Rep_2"
colnames(data_phospho_SWATH_1)[colnames(data_phospho_SWATH_1)=="Rep_3"] <- "Phospho_SWATH_Rep_3"

colnames(data_openSWATH_PTT_fil_1)[colnames(data_openSWATH_PTT_fil_1)=="Rep_1"] <- "openSWATH_PTM_fil_Rep_1"
colnames(data_openSWATH_PTT_fil_1)[colnames(data_openSWATH_PTT_fil_1)=="Rep_2"] <- "openSWATH_PTM_fil_Rep_2"
colnames(data_openSWATH_PTT_fil_1)[colnames(data_openSWATH_PTT_fil_1)=="Rep_3"] <- "openSWATH_PTM_fil_Rep_3"
```

```
colnames(data_openSWATH_PTT_unfil_1)[colnames(data_openSWATH_PTT_unfil_1)=="Rep_1"] <- "openSWATH_PTM_unfil_Rep_1"
colnames(data_openSWATH_PTT_unfil_1)[colnames(data_openSWATH_PTT_unfil_1)=="Rep_2"] <- "openSWATH_PTM_unfil_Rep_2"
colnames(data_openSWATH_PTT_unfil_1)[colnames(data_openSWATH_PTT_unfil_1)=="Rep_3"] <- "openSWATH_PTM_unfil_Rep_3"

data_cor <- merge(x = data_phospho_SWATH_1, data_openSWATH_PTT_fil_1, by = "Peptide", all = TRUE)
data_cor <- merge(x= data_cor, data_openSWATH_PTT_unfil_1, by = "Peptide", all = TRUE)

cor.function.rep(data_cor)

cor.function.rep <- function (data =dataframe()) {
  pearson.cor <- cor(data[,2:9], use="pairwise.complete.obs", method="pearson")
  pearson.cor[lower.tri(pearson.cor)] <- NA

  spearman.cor <- cor(data[,2:9], use="pairwise.complete.obs", method="spearman")
  spearman.cor[upper.tri(spearman.cor, diag = TRUE)] <- NA

  data.plot <- rbind(melt(pearson.cor), melt(spearman.cor))
  data.plot <- data.plot[!is.na(data.plot$value),]
  p <- (ggplot(data.plot, aes(x=Var2, y=Var1, fill=value)) + geom_tile()
        + scale_fill_gradient(low = "white", high="red", name="Correlation\n[R or rho]")
        + xlab("") + ylab("")
        + labs(title="Correlation between samples: Pearson  (upper triangle) and Spearman correlation (lower triangle)")
        + geom_text(aes(fill = data.plot$value, label = round(data.plot$value, digits= 2)))
        + theme(plot.title = element_text(hjust = 0, vjust = 1),
                axis.text.x = element_text(size = 10, angle = 45, hjust = 1)))

  print(p)
}


# 5) correlation of all data

data_1 <- (data_phospho_SWATH.pep_Total[2:14])
data_2 <- (data_openSWATH_PTM_fil.pep_Total[2:14])
data_3 <- (data_openSWATH_PTM_unfil.pep_total[2:14])

data_cor <- merge(data_1, data_2, by = "Peptide", all = TRUE)
data_cor <- merge(data_cor, data_3, by = "Peptide", all = TRUE)

pearson.cor <- cor(data_cor[,2:37], use="pairwise.complete.obs", method="pearson")
pearson.cor[lower.tri(pearson.cor)] <- NA

spearman.cor <- cor(data_cor[,2:37], use="pairwise.complete.obs", method="spearman")
spearman.cor[upper.tri(spearman.cor, diag = TRUE)] <- NA

data.plot <- rbind(melt(pearson.cor), melt(spearman.cor))
data.plot <- data.plot[!is.na(data.plot$value),]
p <- (ggplot(data.plot, aes(x=Var2, y=Var1, fill=value)) + geom_tile()
      + scale_fill_gradient(low = "white", high="red", name="Correlation\n[R or rho]")
      + xlab("") + ylab("")
      + labs(title="Correlation between samples: Pearson  (upper triangle) and Spearman correlation (lower triangle)")
      + geom_text(aes(fill = data.plot$value, label = round(data.plot$value, digits= 1)))
      + theme(plot.title = element_text(hjust = 0, vjust = 1),
              axis.text.x = element_text(size = 10, angle = 45, hjust = 1)))

print(p)


# Analysis t-test and multiple testing for all three samples
# The lists are sorted and the TOP 20 and TOP 50 were comapred to the match
# thresholds wee used to sort the lists.
# Lists were exported and analyzed in excel.

data_pS <- data_phospho_SWATH
data_OSPTM_f <- data_openSWATH_PTM_fil
data_OSPTM_u <- data_openSWATH_PTM_unfil

calc.regulated <- function(data = dataframe()){
  data[,grep("BXD\\.RHO\\.[[:digit:]]{4}\\_*", colnames(data))] <- sapply(data[,grep("BXD\\.RHO\\.[[:digit:]]{4}\\_*", colnames(data))],as.numeric)

  data$mean.young <- apply(data[,grep("BXD.RHO.286*", colnames(data))], 1, function(x)mean(x, na.rm = TRUE))
  data[,grep("BXD.RHO.*", colnames(data))] <- (data[,grep("BXD.RHO.*", colnames(data))] /data$mean.young)
  data[,grep("BXD.RHO.*", colnames(data))] <- sapply(data[,grep("BXD.RHO.*", colnames(data))], function(x)log2(x))
  t_test <- apply(data[,3:14],1,function (x) t.test(x[1:6], x[7:12], paired = FALSE, var.equal = TRUE))
  data$p_value <- unlist(lapply(t_test, function(x) x$p.value))
  data$p_adjusted <- p.adjust(p = data$p_value,method = "BH")
  data$mean.young <- apply(data[,grep("BXD.RHO.286*", colnames(data))], 1, function(x)mean(x, na.rm = TRUE))
  data$mean.old <- apply(data[,grep("BXD.RHO.291*", colnames(data))], 1, function(x)mean(x, na.rm = TRUE))
  data$Protein <- gsub(pattern = "1/", replacement = "", data$Protein)
  return(data)
}

data_pS <-calc.regulated(data_pS)
data_OSPTM_f <- calc.regulated(data_OSPTM_f)
data_OSPTM_u <- calc.regulated(data_OSPTM_u)

write.table(data_pS, file ="data_ps_all.tsv", sep = "\t", quote = FALSE, row.names = FALSE)
write.table(data_OSPTM_f, file ="data_OSPTM_f_all.tsv", sep = "\t", quote = FALSE, row.names = FALSE)
write.table(data_OSPTM_u, file ="data_OSPTM_u_all.tsv", sep = "\t", quote = FALSE, row.names = FALSE)


data_down_old_pS <- subset(data_pS, p_adjusted <= 0.1 & mean.old <= -0.5)
data_up_old_pS <- subset(data_pS, p_adjusted <= 0.1 & mean.old >= 0.5 )

data_down_old_OSPTM_f <- subset(data_OSPTM_f, p_adjusted <= 0.1 & mean.old <= -0.5)
data_up_old_OSPTM_f  <- subset(data_OSPTM_f, p_adjusted <= 0.1 & mean.old >= 0.5 )

data_down_old_OSPTM_u  <- subset(data_OSPTM_u , p_adjusted <= 0.1 & mean.old <= -0.5)
data_up_old_OSPTM_u  <- subset(data_OSPTM_u , p_adjusted <= 0.1 & mean.old >= 0.5 )


write.table(data_down_old_pS, file= "data_down_old_pS.tsv", sep = "\t", quote = FALSE, row.names = FALSE)
write.table(data_up_old_pS , file= "data_up_old_pS.tsv", sep = "\t", quote = FALSE, row.names = FALSE)

write.table(data_down_old_OSPTM_f, file= "data_down_old_OSPTM_f.tsv", sep = "\t", quote = FALSE, row.names = FALSE)
write.table(data_up_old_OSPTM_f, file= "data_up_old_OSPTM_f.tsv", sep = "\t", quote = FALSE, row.names = FALSE)

write.table(data_down_old_OSPTM_u, file= "data_down_old_OSPTM_u.tsv", sep = "\t", quote = FALSE, row.names = FALSE)
write.table(data_up_old_OSPTM_u, file= "data_up_old_OSPTM_u.tsv", sep = "\t", quote = FALSE, row.names = FALSE)

data$threshold = as.factor(abs(data$mean.old) > 0.5 & data$p_adjusted < 0.1)

##Construct the plot object
g = ggplot(data=data, aes(x=mean.old, y=-log10(p_adjusted), colour=threshold)) +
```

```
  geom_point(alpha=0.6, size=4) +
  theme(legend.position = "none") +
  xlim(c(-2.5, 2.5)) + ylim(c(0, 2.8)) +
  xlab("log2(Fold Change)") + ylab("-log10(p_adjusted)") +
  labs(title="Volcano plot of the phospho-SWATH analysis")+
  theme(axis.text = element_text(size = 28, colour = "black"), axis.title = element_text(size = 28),
        plot.title = element_text(size = 30))
g

## extra: Correlation of all intensities in all samples of the three libraries

data_1 <- (data_phospho_SWATH.pep_Total[2:14])
m.data_1 <- melt(data_1, id=c("Peptide"))
m.data_1$Peptide <- paste(m.data_1$Peptide, m.data_1$variable, sep="_")
m.data_1 <- m.data_1[,-2]
colnames(m.data_1)[colnames(m.data_1)=="value"] <- "Phospho_SWATH_Intensity"

data_2 <- (data_openSWATH_PTM_fil.pep_Total[2:14])
m.data_2 <- melt(data_2, id=c("Peptide"))
m.data_2$Peptide <- paste(m.data_2$Peptide, m.data_2$variable, sep="_")
m.data_2 <- m.data_2[,-2]
colnames(m.data_2)[colnames(m.data_2)=="value"] <- "fil_OpenSWATH_PTM_Intensity"

data_3 <- (data_openSWATH_PTM_unfil.pep_total[2:14])
m.data_3 <- melt(data_3, id=c("Peptide"))
m.data_3$Peptide <- paste(m.data_3$Peptide, m.data_3$variable, sep="_")
m.data_3 <- m.data_3[,-2]
colnames(m.data_3)[colnames(m.data_3)=="value"] <- "unfil_OpenSWATH_PTM_Intensity"

data_cor <- merge(m.data_1, m.data_2, by = "Peptide", all = TRUE)
data_cor <- merge(data_cor, m.data_3, by = "Peptide", all = TRUE)

p <- ggplot()+
  geom_point(data = data_cor, mapping = aes(x=data_cor$Phospho_SWATH_Intensity, y=data_cor$fil_OpenSWATH_PTM_Intensity), na.rm = TRUE) +
  scale_x_continuous(trans="log10") +
  scale_y_continuous(trans="log10") +
  #geom_boxplot(width=0.1) +
  labs(title="Correlation of the intensities of the aging phosphopeptide measurements extracted with
       phospho-SWATH vs. filtered OpenSWATH/PTM library",
       x="log10(mean intensity of peptides for Elite measurements)", y="log10(mean intensity of peptides for SWATH measurements)")
print(p)

p <- ggplot()+
  geom_point(data = data_cor, mapping = aes(x=data_cor$Phospho_SWATH_Intensity, y=data_cor$unfil_OpenSWATH_PTM_Intensity), na.rm = TRUE) +
  scale_x_continuous(trans="log10") +
  scale_y_continuous(trans="log10") +
  #geom_boxplot(width=0.1) +
  labs(title="Correlation of the intensities of the aging phosphopeptide measurements extracted with
       phospho-SWATH vs. unfiltered OpenSWATH/PTM library",
       x="log10(mean CV of peptides for Elite measurements)", y="log10(mean CV of peptides for SWATH measurements)")
print(p)

p <- ggplot()+
  geom_point(data = data_cor, mapping = aes(x=data_cor$fil_OpenSWATH_PTM_Intensity, y=data_cor$unfil_OpenSWATH_PTM_Intensity), na.rm = TRUE) +
  scale_x_continuous(trans="log10") +
  scale_y_continuous(trans="log10") +
  #geom_boxplot(width=0.1) +
  labs(title="Correlation of the intensities of the aging phosphopeptide measurements extracted with
       filtered OpenSWATH/PTM library vs. unfiltered OpenSWATH/PTM library",
       x="log10(mean CV of peptides for Elite measurements)", y="log10(mean CV of peptides for SWATH measurements)")
print(p)



cor.test(x =data_cor$Phospho_SWATH_Intensity, y=data_cor$fil_OpenSWATH_PTM_Intensity, na.rm =  TRUE )
cor.test(x =data_cor$Phospho_SWATH_Intensity, y=data_cor$unfil_OpenSWATH_PTM_Intensity, na.rm =  TRUE )
cor.test(x =data_cor$fil_OpenSWATH_PTM_Intensity, y=data_cor$unfil_OpenSWATH_PTM_Intensity, na.rm =  TRUE )



#########################################################################################################################################################
#                                                                                                                                                       #
#                 "LFQ & phospho-SWATH-MS comparison in the aging dataset"                                                                              #
#                                                                                                                                                       #
# Author: Fabian Frommelt                                                                                                                               #
# Date: 05.05.2016 (last update)                                                                                                                        #
# Summary: Aim is to do several comparision between the SWATH result and the Elite result the two LFQ results and two SWATH output                     #
#          one phospho SWATH output and a total SWATH output result. Some of the plots are further used for the thesis and are parts                    #
#          of the thesis. B) it calculates an adjusted p-value and effect size for the regulatd phosphopeptides and also the adjusted                   #
#          p-value and effecte size for the regulated peptides.                                                                                         #
#########################################################################################################################################################


# set the working directory to the Elite LFQ folder
setwd("Y:\\20160418_Aging_dataset_analysis_LFQ_and_SWATH_aging\\")

# load libraries
library(data.table) # for renaming
library(gtools)
library(ggplot2)
library(Peptides)
library(VennDiagram)
library(stringr)


# load the different data into R,
# 2 SWATH outputs and the LFQ output of the iPortal workflow

file.name <- "peptide_level.txt"
data_phospho_SWATH <- read.table(file.path("Y:\\20160129_phospho_SWATH_aging_new_library\\", file.name), header=TRUE, sep="\t", fill=TRUE, stringsAsFactors =
FALSE)
data_total_SWATH <- read.table(file.path("Y:\\20160122_SWATH_analysis_total_cell_lysis_Aging\\", file.name), header = TRUE, sep="\t", fill =TRUE,
stringsAsFactors = FALSE)
data_OpenMS_LFQ <-read.table("peptides.csv", header=TRUE, sep=",", fill=TRUE, stringsAsFactors = FALSE)

# import the results of MaxQuant
file.name <- "Phospho (STY)Sites.txt"
data_MaxQuant_LFQ <- read.table(file.path("Y:\\Max_Quant\\MaxQuant_analysis_160310\\combined\\txt\\", file.name), header=TRUE, sep="\t", fill=TRUE,
stringsAsFactors = FALSE,quote = "")

data_MaxQuant_LFQ <- phospho.mq.annotate(data=data_MaxQuant_LFQ, threshhold = 0.0)

data_MaxQuant_LFQ <- melt(data_MaxQuant_LFQ, id=c("Protein","Phospho..STY..Probabilities"))
colnames(data_MaxQuant_LFQ)[colnames(data_MaxQuant_LFQ) == "Phospho..STY..Probabilities"] <- "Peptide"
colnames(data_MaxQuant_LFQ)[colnames(data_MaxQuant_LFQ) == "Protein"] <- "Protein"
```

```
colnames(data_MaxQuant_LFQ)[colnames(data_MaxQuant_LFQ) == "variable"] <- "Sample_ID"
colnames(data_MaxQuant_LFQ)[colnames(data_MaxQuant_LFQ) == "value"] <- "Intensity"
data_MaxQuant_LFQ <- data_MaxQuant_LFQ[!grepl("CON__", data_MaxQuant_LFQ$Protein),]
data_MaxQuant_LFQ <- data_MaxQuant_LFQ[!grepl("REV__", data_MaxQuant_LFQ$Protein),]
data_MaxQuant_LFQ$Protein <- sub("(sp\\|)([[:alnum:]]+)(\\|[[:alnum:]]+_MOUSE)","\\2", data_MaxQuant_LFQ$Protein)

annotation.file <- "Study_design_MaxQuant.txt"
Study_design <- read.delim2(file.path(getwd(), annotation.file), dec=".", sep ="\t", header=TRUE)
data_MaxQuant_LFQ <- merge(data_MaxQuant_LFQ, Study_design, by = "Sample_ID")
data_mq_LFQ <- dcast(data_MaxQuant_LFQ, Protein + Peptide ~ Mouse, value.var = "Intensity", function(x) max(x, na.rm = TRUE))
data_mq_LFQ[data_mq_LFQ ==0] <- NA

# Renaming the LFQ result of the OpenMS analysis
# renaming the columns of the Elite measurements
setnames(data_OpenMS_LFQ, old=colnames(data_OpenMS_LFQ),
         new=c("Peptide", "Protein", "n_proteins","charge",
               "BXD.RHO.2919_3", "BXD.RHO.2918_2",
               "BXD.RHO.2919_2", "BXD.RHO.2868_1",
               "BXD.RHO.2864_2", "BXD.RHO.2919_1",
               "BXD.RHO.2868_3", "BXD.RHO.2864_3",
               "BXD.RHO.2918_1", "BXD.RHO.2868_2",
               "BXD.RHO.2864_1", "BXD.RHO.2918_3"))
data_OpenMS_LFQ <- data_OpenMS_LFQ[grep(".*(Phospho).*", data_OpenMS_LFQ$Peptide),]
data_OpenMS_LFQ <- data_OpenMS_LFQ[ , !names(data_OpenMS_LFQ) %in% c("n_proteins", "charge")]
data_OpenMS_LFQ <- data_OpenMS_LFQ[!grepl("DECOY_", data_OpenMS_LFQ$Protein),]
data_OpenMS_LFQ[data_OpenMS_LFQ ==0] <- NA

# get rid of the n-Fragment row of teh SWATH data
data_phospho_SWATH <- data_phospho_SWATH[ , !names(data_phospho_SWATH) %in% c("nFragment")]
data_total_SWATH <- data_total_SWATH[ , !names(data_total_SWATH) %in% c("nFragment")]
setnames(data_total_SWATH, old=colnames(data_total_SWATH),
         new=c("Protein", "Peptide","BXD.RHO.2864_1","BXD.RHO.2864_2",
               "BXD.RHO.2864_3","BXD.RHO.2868_1","BXD.RHO.2868_2",
               "BXD.RHO.2868_3","BXD.RHO.2918_1","BXD.RHO.2918_2","BXD.RHO.2918_3",
               "BXD.RHO.2919_1","BXD.RHO.2919_2","BXD.RHO.2919_3"))

# annotate phosphorylation for all und delocalization
phosphorylation.annotation <- function(data){
  # Function changes annotation and creates a column in which all Proteins are written only with
  # their identifier
  x <- data
  x[,grep("BXD\\.RHO\\.[[:digit:]]{4}\\_*", colnames(x))] <- sapply(x[,grep("BXD\\.RHO\\.[[:digit:]]{4}\\_*", colnames(x))],as.numeric)
  x <- as.data.frame(sapply(x,gsub,pattern="*\\(UniMod_21\\)*",replacement="\\(Phospho\\)"), stringsAsFactors = FALSE)
  x <- as.data.frame(sapply(x,gsub,pattern="*\\(UniMod_35\\)*",replacement="\\(Oxidation\\)"), stringsAsFactors = FALSE)
  x <- as.data.frame(sapply(x,gsub,pattern="*\\(UniMod_4\\)*",replacement="\\(Carbamidomethyl\\)"), stringsAsFactors = FALSE)
  x$Protein_alone <- x$Protein
  x$Protein_alone <- sapply(x$Protein_alone,gsub,pattern="Subgroup\\_[[:digit:]]{1}\\_[[:digit:]]{1}\\/",replacement="")
  x$Protein_alone <- gsub("\\(Phospho\\)", replacement = "", x = x$Protein_alone)
  x$Protein_alone <- gsub("\\(Oxidation\\)", replacement = "", x = x$Protein_alone)
  x$Protein_alone <- gsub("\\(Carbamidomethyl\\)", replacement = "", x = x$Protein_alone)
  x$Protein_alone <- sub("\\_([[:alnum:]])+","", x$Protein_alone)
  x$Count_Phospho <- sapply("(Phospho)", str_count, string =x$Peptide)
  x$Deloc <- x$Peptide
  x$Deloc <- sapply(x$Deloc,gsub,pattern="*\\(Phospho\\)*",replacement="")
  x$Deloc <- sapply(x$Deloc,gsub,pattern="*\\(Oxidation\\)*",replacement="")
  x$Delocalized <- paste(x$Deloc, x$Count_Phospho, sep="_P")
  x <- subset(x, select = -c(Deloc))
  x$Delocalized <- gsub(x$Delocalized, pattern = "NA_PNA", replacement = NA )
  x$Delocalized <- gsub(x$Delocalized, pattern = "\\_P0", replacement = "" )
  return(x)
}

data_phospho_SWATH <-phosphorylation.annotation(data = data_phospho_SWATH)
data_total_SWATH <- phosphorylation.annotation(data = data_total_SWATH)
data_mq_LFQ <- phosphorylation.annotation(data = data_mq_LFQ)
data_OpenMS_LFQ <- phosphorylation.annotation(data = data_OpenMS_LFQ)

Statistics.mean.sd.CV.Replciate <-
  function (data = dataframe()){
    m.data <- melt(data, id=c("Protein", "Peptide", "Delocalized", "Count_Phospho", "Protein_alone"))
    m.data$Replicate <- paste("Rep", gsub(".*_([[:digit:]])$", "\\1", m.data$variable))
    m.data$Mouse <- gsub("_[[:digit:]]$", "", m.data$variable)
    m.data$value <- as.numeric(m.data$value)
    data <- dcast(m.data, Protein + Peptide + Mouse  ~ Replicate)
    data$mean.rep <- apply(data[,grep("Rep", colnames(data))], 1, function(x)mean(x, na.rm = TRUE))
    data$sd.rep <- apply(data[,grep("Rep", colnames(data))], 1, function(x)sd(x, na.rm = TRUE))
    data$CV.rep <- apply(data[, grep("Rep", colnames(data))], 1, function(x) sd(x,na.rm = TRUE)/mean(x, na.rm = TRUE))
    data$Var.rep <- apply(data[,grep("Rep", colnames(data))], 1, function(x)var(x, y=NULL, na.rm = TRUE))
    return(data)
  }

data_phospho_SWATH.rep <- Statistics.mean.sd.CV.Replciate(data_phospho_SWATH)
data_total_SWATH.rep <- Statistics.mean.sd.CV.Replciate(data = data_total_SWATH)
data_mq_LFQ.rep <- Statistics.mean.sd.CV.Replciate(data = data_mq_LFQ)
data_OpenMS_LFQ.rep <- Statistics.mean.sd.CV.Replciate(data = data_OpenMS_LFQ)

Statistics.mean.sd.CV.total <-
  function (data = dataframe()){
    data[,grep("BXD.RHO*", colnames(data))] <- sapply(data[,grep("BXD.RHO*", colnames(data))],as.numeric)
    data$mean.total <- apply(data[,grep("BXD.RHO*", colnames(data))], 1, function(x)mean(x, na.rm = TRUE))
    data$sd.total <- apply(data[,grep("BXD.RHO*", colnames(data))], 1, function(x)sd(x, na.rm = TRUE))
    data$CV.total <- apply(data[, grep("BXD.RHO*", colnames(data))], 1, function(x) sd(x, na.rm = TRUE)/mean(x, na.rm = TRUE))
    data$Var.total <- apply(data[,grep("BXD.RHO*", colnames(data))], 1, function(x)var(x, y=NULL, na.rm = TRUE))
    return(data)
  }

data_phospho_SWATH.total <- Statistics.mean.sd.CV.total(data = data_phospho_SWATH)
data_total_SWATH.total <- Statistics.mean.sd.CV.total(data = data_total_SWATH)
data_mq_LFQ.total <- Statistics.mean.sd.CV.total(data = data_mq_LFQ)
data_OpenMS_LFQ.total <- Statistics.mean.sd.CV.total(data = data_OpenMS_LFQ)

write.table(data_mq_LFQ, file= "data_mq_LFQ.tsv", sep = "\t", quote = FALSE, row.names = FALSE)
write.table(data_OpenMS_LFQ , file= "data_OpenMS_LFQ.tsv", sep = "\t", quote = FALSE, row.names = FALSE)

### correleate MQ with OpenMS LFQ
m_mq_LFQ <- melt(data_mq_LFQ[,c(1,3:14,17)], id = c("Protein", "Delocalized"))
m_OpenMS_LFQ <- melt(data_OpenMS_LFQ[,c(2,3:14,17)], id = c("Protein", "Delocalized"))

colnames(m_mq_LFQ)[colnames(m_mq_LFQ)=="value"] <- "MQ_Intensity"
colnames(m_OpenMS_LFQ)[colnames(m_OpenMS_LFQ)=="value"] <- "OpenMS_Intensity"

data_plot <- merge(x = m_mq_LFQ, y= m_OpenMS_LFQ, by = c("Protein", "Delocalized", "variable"), all = TRUE)
data_plot$MQ_Intensity <- as.numeric(data_plot$MQ_Intensity)
data_plot$OpenMS_Intensity <- as.numeric(data_plot$OpenMS_Intensity)
```

```
p <- ggplot()+
  geom_point(data = data_plot, mapping = aes(x=MQ_Intensity, y=OpenMS_Intensity), na.rm = TRUE) +
  scale_x_continuous(trans="log10") +
  scale_y_continuous(trans="log10") +
  #geom_boxplot(width=0.1) +
  labs(title="Correlation of the MaxQuant LFQ with the OpenMS LFQ",
       x="log10(MaxQuant LFQ Intensity)", y="log10(OpenMS LFQ Intensity)")
print(p)

cor.test(x =data_plot$MQ_Intensity , y=data_plot$OpenMS_Intensity, na.rm = TRUE)

# 3) Plot the Coefficient of Variation as violin plot samples
# construct a Dataset for plotting the Variance values in a violin plot

sub.data_phospho_SWATH.total <- subset(data_phospho_SWATH.total, select = c("Peptide", "CV.total"))
sub.data_phospho_SWATH.total <- melt(sub.data_phospho_SWATH.total, id = "Peptide")
sub.data_phospho_SWATH.rep <- subset(data_phospho_SWATH.rep, select = c("Peptide", "CV.rep"))
sub.data_phospho_SWATH.rep <- melt(sub.data_phospho_SWATH.rep, id ="Peptide")

sub.data_total_SWATH.total <- subset(data_total_SWATH.total, select = c("Peptide", "CV.total"))
sub.data_total_SWATH.total <- melt(sub.data_total_SWATH.total, id = "Peptide")
sub.data_total_SWATH.rep <- subset(data_total_SWATH.rep, select = c("Peptide", "CV.rep"))
sub.data_total_SWATH.rep <- melt(sub.data_total_SWATH.rep, id ="Peptide")

sub.data_mq_LFQ.total <- subset(data_mq_LFQ.total, select = c("Peptide", "CV.total"))
sub.data_mq_LFQ.total <- melt(sub.data_mq_LFQ.total, id = "Peptide")
sub.data_mq_LFQ.rep <- subset(data_mq_LFQ.rep, select = c("Peptide", "CV.rep"))
sub.data_mq_LFQ.rep <- melt(sub.data_mq_LFQ.rep, id ="Peptide")

sub.data_OpenMS_LFQ.total <- subset(data_OpenMS_LFQ.total, select = c("Peptide", "CV.total"))
sub.data_OpenMS_LFQ.total <- melt(sub.data_OpenMS_LFQ.total, id = "Peptide")
sub.data_OpenMS_LFQ.rep <- subset(data_OpenMS_LFQ.rep, select = c("Peptide", "CV.rep"))
sub.data_OpenMS_LFQ.rep <- melt(sub.data_OpenMS_LFQ.rep, id ="Peptide")

# annotation of the data
sub.data_phospho_SWATH.total$variable <- gsub("CV.total", "CV.all.phospho-SWATH", sub.data_phospho_SWATH.total$variable)
sub.data_phospho_SWATH.rep$variable <- gsub("CV.rep", "CV.rep.phospho-SWATH", sub.data_phospho_SWATH.rep$variable)

sub.data_total_SWATH.total$variable <- gsub("CV.total", "CV.all.total.lysate-SWATH", sub.data_total_SWATH.total$variable)
sub.data_total_SWATH.rep$variable <- gsub("CV.rep", "CV.rep.total.lysate-SWATH", sub.data_total_SWATH.rep$variable)

sub.data_mq_LFQ.total$variable <- gsub("CV.total", "CV.all.MQ_LFQ", sub.data_mq_LFQ.total$variable)
sub.data_mq_LFQ.rep$variable <- gsub("CV.rep", "CV.rep.MQ_LFQ", sub.data_mq_LFQ.rep$variable)

sub.data_OpenMS_LFQ.total$variable <- gsub("CV.total", "CV.all.OpenMS_LFQ", sub.data_OpenMS_LFQ.total$variable)
sub.data_OpenMS_LFQ.rep$variable <- gsub("CV.rep", "CV.rep.OpenMS_LFQ", sub.data_OpenMS_LFQ.rep$variable)

table.SWATH.CV <- rbind(sub.data_phospho_SWATH.total, sub.data_phospho_SWATH.rep,
                        sub.data_total_SWATH.total, sub.data_total_SWATH.rep,
                        sub.data_mq_LFQ.total, sub.data_mq_LFQ.rep,
                        sub.data_OpenMS_LFQ.total, sub.data_OpenMS_LFQ.rep)

table.SWATH.CV$label <- factor(table.SWATH.CV$variable, c("CV.all.phospho-SWATH", "CV.rep.phospho-SWATH", "CV.all.total", "CV.rep.total",
                                                          "CV.all.MQ_LFQ", "CV.rep.MQ_LFQ", "CV.all.OpenMS_LFQ", "CV.rep.OpenMS_LFQ"))


table.SWATH.CV <- rbind(sub.data_phospho_SWATH.total, sub.data_phospho_SWATH.rep,
                        sub.data_mq_LFQ.total, sub.data_mq_LFQ.rep,
                        sub.data_OpenMS_LFQ.total, sub.data_OpenMS_LFQ.rep)

table.SWATH.CV$label <- factor(table.SWATH.CV$variable, c("CV.all.phospho-SWATH", "CV.rep.phospho-SWATH",
                                                          "CV.all.MQ_LFQ", "CV.rep.MQ_LFQ", "CV.all.OpenMS_LFQ", "CV.rep.OpenMS_LFQ"))


table.SWATH.CV <- rbind(sub.data_phospho_SWATH.total, sub.data_phospho_SWATH.rep,
                        sub.data_total_SWATH.total, sub.data_total_SWATH.rep)

table.SWATH.CV$label <- factor(table.SWATH.CV$variable, c("CV.all.phospho-SWATH", "CV.rep.phospho-SWATH",
                                                          "CV.all.total.lysate-SWATH", "CV.rep.total.lysate-SWATH"))


p <- ggplot(table.SWATH.CV, aes(factor(label), value)) +
  geom_violin(scale="area") +
  stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median,
               geom = "crossbar", width = 0.3) +
  #scale_y_continuous(trans="log10") +
  #geom_boxplot(width=0.1) +
  labs(title=" Violin plots of the CV within the replicates and over all samples,\n analyzed for all phosphopeptides of the LFQ and the phospho-SWATH analysis"
,
       x="", y="[sd]/[mean of all or replicate Intensity]") +
  #scale_x_discrete(labels=c("CV.phospho.total", "CV.phospho.rep", "CV.total.cell.lysate")) +
  theme(axis.text = element_text(size = 12, colour = "black"), axis.title = element_text(size = 12),
        plot.title = element_text(size = 15),
        axis.text.x = element_text(hjust = 1, angle = 45))
print(p)


p <- ggplot(table.SWATH.CV, aes(factor(label), value)) +
  geom_violin(scale="area") +
  stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median,
               geom = "crossbar", width = 0.3) +
  #scale_y_continuous(trans="log10") +
  #geom_boxplot(width=0.1) +
  labs(title=" Violin plots of the CV within the replicates and over all samples,\n for the phospho-SWATH and the total tissue lysate SWATH analysis" ,
       x="", y="[sd]/[mean of all or replicate Intensity]") +
  #scale_x_discrete(labels=c("CV.phospho.total", "CV.phospho.rep", "CV.total.cell.lysate")) +
  theme(axis.text = element_text(size = 12, colour = "black"), axis.title = element_text(size = 12),
        plot.title = element_text(size = 15),
        axis.text.x = element_text(hjust = 1, angle = 45))
print(p)

# calculate media nand mode for the data

estimate_mode <- function(x) {
  d <- density(x)
  d$x[which.max(d$y)]
}


aggregate(table.SWATH.CV[,"value"], by=list(table.SWATH.CV$variable), FUN = function(x) median(x, na.rm=TRUE))


## Venn Plot of the matching in both LFQ resutls

peptide_list_OpenMS <- list(data_OpenMS_LFQ$Delocalized)
```

```
peptide_list_MQ <- list(data_mq_LFQ$Delocalized)

# Venn plots
venn.plot <- venn.diagram(
  x=c(peptide_list_OpenMS,
      peptide_list_MQ),
  filename = NULL,
  sacaled = TRUE,
  main = "Delocalized Phosphopeptides found in MaxQuant and OpenMS LFQ result",
  col = "black",
  fill = c("blue", "green"),
  category = c("OpenMS",
               "MaxQuant"),
  alpha = 0.50,
  cat.col = c("Black"),
  cat.cex = 1.0,
  main.cex = 2.0,
  cat.fontface = "bold");
grid.draw(venn.plot);

dev.off()

# B) Regulated Phosphopeptides for the phosphopeptide enriched samples
#    and regulatede peptides for the total tissue lysate samples

file.name <- "protein_level.txt"
df_phospho_SWATH <- read.table(file.path("Y:\\20160129_phospho_SWATH_aging_new_library\\", file.name), header=TRUE, sep="\t", fill=TRUE, stringsAsFactors =
FALSE)
df_total_SWATH <- read.table(file.path("Y:\\20160122_SWATH_analysis_total_cell_lysis_Aging\\", file.name), header = TRUE, sep="\t", fill =TRUE,
stringsAsFactors = FALSE)

## rename the BXD_RHO* column name of the total lysate samples

setnames(df_total_SWATH, old=colnames(df_total_SWATH),
         new=c("Protein","BXD.RHO.2864_1","BXD.RHO.2864_2",
               "BXD.RHO.2864_3","BXD.RHO.2868_1","BXD.RHO.2868_2",
               "BXD.RHO.2868_3","BXD.RHO.2918_1","BXD.RHO.2918_2","BXD.RHO.2918_3",
               "BXD.RHO.2919_1","BXD.RHO.2919_2","BXD.RHO.2919_3", "nFragment", "nPeptide"))

calc.regulated <- function(data = dataframe()){
  data[,grep("BXD\\.RHO\\.[[:digit:]]{4}\\_*", colnames(data))] <- sapply(data[,grep("BXD\\.RHO\\.[[:digit:]]{4}\\_*", colnames(data))],as.numeric)

  data$mean.young <- apply(data[,grep("BXD.RHO.286*", colnames(data))], 1, function(x)mean(x, na.rm = TRUE))
  data[,grep("BXD.RHO.*", colnames(data))] <- (data[,grep("BXD.RHO.*", colnames(data))] /data$mean.young)
  data[,grep("BXD.RHO.*", colnames(data))] <- sapply(data[,grep("BXD.RHO.*", colnames(data))], function(x)log2(x))
  t_test <- apply(data[,2:13],1,function (x) t.test(x[1:6], x[7:12], paired = FALSE, var.equal = TRUE))
  data$p_value <- unlist(lapply(t_test, function(x) x$p.value))
  data$p_adjusted <- p.adjust(p = data$p_value,method = "BH")
  data$mean.young <- apply(data[,grep("BXD.RHO.286*", colnames(data))], 1, function(x)mean(x, na.rm = TRUE))
  data$mean.old <- apply(data[,grep("BXD.RHO.291*", colnames(data))], 1, function(x)mean(x, na.rm = TRUE))
  data$Protein <- gsub(pattern = "1/", replacement = "", data$Protein)
  return(data)
}

df_phospho_SWATH <-calc.regulated(df_phospho_SWATH)
df_total_SWATH <- calc.regulated(df_total_SWATH)


write.table(df_phospho_SWATH, file ="df_phospho_SWATH_all_FC_p_adjusted.tsv", sep = "\t", quote = FALSE, row.names = FALSE)
write.table(df_total_SWATH, file ="df_total_SWATH_all_FC_p_adjusted.tsv", sep = "\t", quote = FALSE, row.names = FALSE)

data_down_old_phospho_SWATH <- subset(df_phospho_SWATH, p_adjusted <= 0.1 & mean.old <= -0.5)
data_up_old_phospho_SWATH <- subset(df_phospho_SWATH, p_adjusted <= 0.1 & mean.old >= 0.5 )

data_regulated_old_total_SWATH <-
  data_down_old_total_SWATH <- subset(df_total_SWATH, p_adjusted <= 0.1 & mean.old <= -0.5)
data_up_old_total_SWATH  <- subset(df_total_SWATH, p_adjusted <= 0.1 & mean.old >= 0.5 )


write.table(data_down_old_phospho_SWATH, file= "data_down_old_phospho_SWATH.tsv", sep = "\t", quote = FALSE, row.names = FALSE)
write.table(data_up_old_phospho_SWATH , file= "data_up_old_phospho_SWATH.tsv", sep = "\t", quote = FALSE, row.names = FALSE)

write.table(data_down_old_total_SWATH, file= "data_down_old_total_SWATH.tsv", sep = "\t", quote = FALSE, row.names = FALSE)
write.table(data_up_old_total_SWATH, file= "data_up_old_total_SWATH.tsv", sep = "\t", quote = FALSE, row.names = FALSE)

# plot the data you want to plot

data <- df_total_SWATH
data$threshold = as.factor(abs(data$mean.old) > 0.5 & data$p_adjusted < 0.1)

# Construct the plot object
g = ggplot(data=data, aes(x=mean.old, y=-log10(p_adjusted), colour=threshold)) +
  geom_point(alpha=0.6, size=4) +
  theme(legend.position = "none") +
  xlim(c(-2.5, 2.5)) + ylim(c(0, 4)) +
  xlab("log2(Fold Change)") + ylab("-log10(p_adjusted)") +
  labs(title="Volcano plot of the total cell lysate SWATH analysis")+
  theme(axis.text = element_text(size = 10, colour = "black"), axis.title = element_text(size = 12),
        plot.title = element_text(size = 14))
g

data <- df_phospho_SWATH
data$threshold = as.factor(abs(data$mean.old) > 0.5 & data$p_adjusted < 0.1)

##Construct the plot object
plot2 = ggplot(data=data, aes(x=mean.old, y=-log10(p_adjusted), colour=threshold)) +
  geom_point(alpha=0.6, size=4) +
  theme(legend.position = "none") +
  xlim(c(-2.5, 2.5)) + ylim(c(0, 3)) +
  xlab("log2(Fold Change)") + ylab("-log10(p_adjusted)") +
  labs(title="Volcano plot of the total cell lysate SWATH analysis")+
  theme(axis.text = element_text(size = 10, colour = "black"), axis.title = element_text(size = 12),
        plot.title = element_text(size = 14))
plot2
```

```
#########################################################################################################################################################
#                                                                                                                                                       #
#                "Analysis of peptide identification results in DDA results of the aging experiment"                                                    #
#                                                                                                                                                       #
# Author: Fabian Frommelt                                                                                                                               #
# Date: 05.05.2016                                                                                                                                      #
# Summary: R-script based analysis of the DDA Orbitrap results, analayzed via the iPortal platform and MaxQuant peptide identfication                   #
#          searches. Further the DDA measurments on the TripleToF are analyzed.                                                                        #
#                                                                                                                                                       #
#########################################################################################################################################################


## Quick analysis of the aging samples measured with the OrbitrapElite

setwd("Y:\\20160418_Aging_dataset_analysis_LFQ_and_SWATH_aging\\")

# load required R packages
library(gplots)
library(ggplot2)
library(stringr)
library(gridExtra)
library(reshape2)
library(VennDiagram)

# import the peptide search results of iportal
file.name <- "peptides.tsv"
data_search_Elite <- read.table(file.path("Y:\\html\\openBIS\\20151104193832473-1115722\\", file.name), header=TRUE,
                                sep="\t", fill=TRUE, stringsAsFactors = FALSE)
data_search_SWATH <- read.table(file.path("Y:\\html\\openBIS\\20160105132435883-1133292\\", file.name), header=TRUE,
                                sep="\t", fill=TRUE, stringsAsFactors = FALSE)


# import the results from MaxQuant
file.name <- "Phospho (STY)Sites.txt"
data_MaxQuant_LFQ <- read.table(file.path("Y:\\Max_Quant\\MaxQuant_analysis_160310\\combined\\txt\\", file.name),
                                header=TRUE, sep="\t", fill=TRUE, stringsAsFactors = FALSE,quote = "")
file.name <- "peptides.txt"
data_mq_peptide_1 <- read.table(file.path("Y:\\Max_Quant\\MaxQuant_analysis_160310\\combined\\txt\\", file.name),
                                header=TRUE, sep="\t", fill=TRUE, stringsAsFactors = FALSE,quote = "")

# generic function from the beads to buffer combination experiment
pat <- "^[[:space:]]*$"
data_mq_peptide_1 <- data_mq_peptide_1[grepl(pat, data_mq_peptide_1$Phospho..STY..site.IDs),]
data_MaxQuant_LFQ <- phospho.mq.annotate(data=data_MaxQuant_LFQ, threshhold = 0.0)

data_MaxQuant_LFQ <- melt(data_MaxQuant_LFQ, id=c("Protein","Phospho..STY..Probabilities"))
colnames(data_MaxQuant_LFQ)[colnames(data_MaxQuant_LFQ) == "Phospho..STY..Probabilities"] <- "modified_peptide"
colnames(data_MaxQuant_LFQ)[colnames(data_MaxQuant_LFQ) == "Protein"] <- "protein"
colnames(data_MaxQuant_LFQ)[colnames(data_MaxQuant_LFQ) == "variable"] <- "Sample_ID"
colnames(data_MaxQuant_LFQ)[colnames(data_MaxQuant_LFQ) == "value"] <- "Intensity_phospho"

data_mq_peptide_1 <- (data_mq_peptide_1[,c("Leading.razor.protein","Sequence",colnames(data_mq_peptide_1)
                                           [grep("Intensity.FF[[:digit:]]{3}$",colnames(data_mq_peptide_1))])])
colnames(data_mq_peptide_1) <- gsub("Intensity.",replacement = "", x= colnames(data_mq_peptide_1))
data_mq_peptide_1 <- melt(data_mq_peptide_1, id=c("Leading.razor.protein","Sequence"))
colnames(data_mq_peptide_1)[colnames(data_mq_peptide_1) == "Leading.razor.protein"] <- "protein"
colnames(data_mq_peptide_1)[colnames(data_mq_peptide_1) == "Sequence"] <- "peptide"
colnames(data_mq_peptide_1)[colnames(data_mq_peptide_1) == "variable"] <- "Sample_ID"
colnames(data_mq_peptide_1)[colnames(data_mq_peptide_1) == "value"] <- "Intensity_peptide"

data_MaxQuant_LFQ <- data_MaxQuant_LFQ[data_MaxQuant_LFQ[,4] > 0, ]
data_mq_peptide_1 <- data_mq_peptide_1[data_mq_peptide_1[,4] > 0, ]

data_MaxQuant_LFQ$peptide <- data_MaxQuant_LFQ$modified_peptide
data_MaxQuant_LFQ$peptide <- sapply(data_MaxQuant_LFQ$peptide,gsub,pattern="*\\(Phospho\\)*",replacement="")

data_mq_all <- merge(data_mq_peptide_1, data_MaxQuant_LFQ, by=c("protein", "Sample_ID", "peptide"), all.x = TRUE, all.y = TRUE)
data_mq_all <- data_mq_all[!grepl("CON__", data_mq_all$protein),]
data_mq_all <- data_mq_all[!grepl("REV__", data_mq_all$protein),]
data_mq_all$protein <- sub("(sp\\|)([[:alnum:]]+)(\\|[[:alnum:]]+_MOUSE)","\\2", data_mq_all$protein)

# generic functions from the beads to buffer combination experiment
data_mq_all <- annotate.phospho(data_mq_all)
data_mq_all <- delocalize.phospho(data_mq_all)

data_search_Elite <- annotate.phospho(data_search_Elite)
data_search_Elite <- delocalize.phospho(data_search_Elite)

data_search_SWATH <- annotate.phospho(data_search_SWATH)
data_search_SWATH <- delocalize.phospho(data_search_SWATH)

annotation.file <- "Study_design_MaxQuant2.txt"
Study_design <- read.delim2(file.path(getwd(), annotation.file), dec=".", sep ="\t", header=TRUE)
data_mq_all <- merge(data_mq_all, Study_design, by = "Sample_ID")

annotation.iportal <-

  function (data, sample.annotation, data.type = "iportal", column.file = "spectrum",
            change.run.id = TRUE, verbose = FALSE)
  {

    if (!(column.file %in% colnames(data))) {
      warning("Warning: column for spectrum is not present in data file")
    }


    if (nlevels(factor(paste(sample.annotation$spectrum))) !=
        nlevels(factor(data[, column.file]))) {
      stop("Warning: the number of sample annotation condition and spectrum in data are not balanced.",
           "\n", "Different filenames in sample annotation file: ",
           nlevels(factor(sample.annotation$Condition)), "\n",
           "Different filenames in data file: ", nlevels(factor(data[,
                                                                     column.file])))
    }

    if (data.type == "iportal") {
      colnames(data) <- gsub("Run", column.file, colnames(data))
      for (i in levels(sample.annotation$spectrum)) {

        coord <- grep(i, data[, column.file])
        if (length(coord) == 0) {
          warning("No measurement value found for this sample in the data file: ",
                  print(i))
        }
```

```
        data.subset <- sample.annotation[which(i == sample.annotation$spectrum),
                                         ]
        data[coord, "Sample_ID"] <- data.subset[, "Sample_ID"]
        data[coord, "Replicate"] <- data.subset[, "Replicate"]
        data[coord, "Mouse"] <- data.subset[, "Mouse"]
        data[coord, "Engine"] <- data.subset[, "Engine"]

      }
      add.colnames <- colnames(data)[!(colnames(data) %in%
                                       c("Sample_ID", "peptide", "modified_peptide", "protein","S_167", "T_181", "Y_243",
                                         "DECOY", "PHOSPHO", "Count_Phospho", "Delocalized",
                                         "Replicate", "Mouse", "Engine"))]
      data <- data[, c("Sample_ID", "protein","peptide", "modified_peptide", "S_167", "T_181", "Y_243",
                       "DECOY", "PHOSPHO", "Count_Phospho", "Delocalized", "Engine",
                       "Replicate", "Mouse",
                       add.colnames)]
      return(data)
    }
  }

annotation.file <- "Study_design_Elite_TPP.txt"
Study_design <- read.delim2(file.path(getwd(), annotation.file), dec=".", sep ="\t", header=TRUE)
data_search_Elite$spectrum <- gsub(pattern="*~.*",replacement="", data_search_Elite$spectrum )
data_search_Elite <- annotation.iportal(data_search_Elite, Study_design)

annotation.file <- "Study_design_SWATH_TPP.txt"
Study_design <- read.delim2(file.path(getwd(), annotation.file), dec=".", sep ="\t", header=TRUE)
data_search_SWATH$spectrum <- gsub(pattern="*~.*",replacement="", data_search_SWATH$spectrum )
data_search_SWATH <- annotation.iportal(data_search_SWATH, Study_design)


data_iportal <- do.call("rbind", list(data_search_SWATH, data_search_Elite))
n_data_iportal <- subset(data_iportal, select = c(Sample_ID, protein, peptide, modified_peptide, PHOSPHO, DECOY, Delocalized, Engine, Mouse))
n_data_maxquant <- subset(data_mq_all, select = c(Sample_ID, protein, peptide, modified_peptide, PHOSPHO, DECOY, Delocalized, Engine, Mouse))
n_data_all <- do.call("rbind", list(n_data_iportal, n_data_maxquant))
n_data_all <- unique(n_data_all)

sub_PEPIDE <- subset(n_data_all, DECOY == FALSE)
sub_DECOY <- subset(n_data_all, DECOY == TRUE)
sub_PHOSPHO <-subset(n_data_all, DECOY == FALSE & PHOSPHO == TRUE)
sub_PHOSPHO_Protein <- subset(n_data_all, DECOY == FALSE & PHOSPHO == TRUE)
sub_PHOSPHO_Protein <- unique(subset(sub_PHOSPHO_Protein, select = c(Sample_ID, protein, Engine)))
sub_DELOCALIZED <- subset(n_data_all, DECOY == FALSE & PHOSPHO == TRUE)
sub_DELOCALIZED <- unique(subset(sub_DELOCALIZED, select = c(Sample_ID, protein, Engine, Delocalized)))


p1 <-ggplot(sub_PEPIDE, aes(factor(Mouse))) +
  geom_bar(aes(fill =Sample_ID), position = "dodge") +
  facet_wrap( ~Engine, ncol = 3) +
  ggtitle("Identified phopeptides for all iportal settings") +
  ylim(0, 11000) +
  labs(x = "", y = "counts")+
  theme(
    axis.text.x = element_text(size = 6, angle = 45, hjust = 1),
    axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 10),
    axis.title.y = element_text(size = 10),
    plot.title = element_text(size = 14),
    strip.text.x = element_text(size = 10))
plot(p1)

p2 <-ggplot(sub_PHOSPHO, aes(factor(Mouse))) +
  geom_bar( aes(fill = Sample_ID), position ="dodge") +
  facet_wrap( ~Engine, ncol = 3) +
  ggtitle("Identified unique phosphophopeptides for the three DDA measurements") +
  ylim(0, 4000) +
  labs(x = "", y = "counts")+
  theme(
    axis.text.x = element_text(size = 10, angle = 45, hjust =1, color = "black"),
    axis.text.y = element_text(size = 12),
    axis.title = element_text(size = 12),
    axis.title.y = element_text(size = 12),
    plot.title = element_text(size = 18),
    strip.text.x = element_text(size = 12))
plot(p2)

p3 <-ggplot(sub_DELOCALIZED, aes(factor(Sample_ID))) +
  geom_bar(aes(fill =Sample_ID)) +
  facet_wrap( ~Engine, ncol = 2) +
  ggtitle("Identified unique delocalized phosphopeptdies for all iportal settings") +
  ylim(0, 4000) +
  labs(x = "", y = "counts")+
  theme(
    axis.text.x = element_text(size = 6, angle = 45),
    axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 10),
    axis.title.y = element_text(size = 10),
    plot.title = element_text(size = 14),
    strip.text.x = element_text(size = 10))
plot(p3)


p4 <-ggplot(sub_PHOSPHO_Protein, aes(factor(Sample_ID))) +
  geom_bar(aes(fill =Sample_ID)) +
  facet_wrap( ~Engine, ncol = 2) +
  ggtitle("Identified unique phosphoproteins for all iportal settings") +
  ylim(0, 2090) +
  labs(x = "", y = "counts")+
  theme(
    axis.text.x = element_text(size = 6, angle = 45),
    axis.text.y = element_text(size = 8),
    axis.title = element_text(size = 10),
    axis.title.y = element_text(size = 10),
    plot.title = element_text(size = 14),
    strip.text.x = element_text(size = 10))
plot(p4)



count.species <- function(x, species=c()){

  if (species == "phospho_proteins") {
    x <- melt(x, id =c("protein", "Engine"))
    x$variable <- 1
```

```
    x <- dcast(x, value ~ protein + Engine, value.var = "variable")
  } else if (species == "peptide"){
    ## is a rather complex fragment of code ...
    ## in fact, there are some peptides, which have different modifications so therefore we need to combine the once with modifcation,
    ## and the once which do not have any modifaction in one row, to really count the number of detected peptides, because it can also be,
    ## so that we account for peptides with the phosphorylation on different sites.

    x <- subset(x, select = c(Sample_ID, peptide, Engine, protein, modified_peptide))
    index <- x$modified_peptide == is.na(TRUE)
    index[is.na(index)] <- TRUE
    x$modified_peptide[index] <- (x$peptide[index])
    x <- subset(x, select = c(Sample_ID, modified_peptide, Engine, protein))
    x <- melt(subset(x, select = c(Sample_ID, modified_peptide, Engine)), id =c("modified_peptide", "Engine"))
    x <- unique(x)
    x$variable <- 1
    x <- dcast(x, value ~ modified_peptide + Engine, value.var = "variable")
  } else if (species == "phospho_peptide") {

    ## produces the same result as in the earlier analysis. I am not so sure about the unique. The issue is,
    ## the number of phosphopeptide do not alter, if you take the Proteins into account and therefore I would rather suggest not to unique.
    ## In the protein list in the peptides.tsv list are only proteotpyic peptides (checked it in the excel)
    x <- subset(x, select = c(Sample_ID, modified_peptide, Engine, protein))
    x <- melt(subset(x, select = c(Sample_ID, modified_peptide, Engine)), id =c("modified_peptide", "Engine"))
    x <- unique(x)
    x$variable <- 1
    x <- dcast(x, value ~ modified_peptide + Engine, value.var = "variable")

  } else if (species == "delocalized") {
    ## same about the delocalized as for the peptides, if the protein is took into account, there is not one entrey removed. Therefore we should stick
    ## here also to the non reduced one.
    x <- melt(subset(x, select = c(Sample_ID, Delocalized, Engine)), id =c("Delocalized", "Engine"))
    x <- unique(x)
    x$variable <- 1
    x <- dcast(x, value ~ Delocalized + Engine, value.var = "variable")
  } else if (species == "decoy"){
    x <- sub_DECOY
    x <- melt(subset(x, select = c(Sample_ID, protein, Engine)), id =c("protein", "Engine"))
    x <- unique(x)
    x$variable <- 1
    x <- dcast(x, value ~ protein + Engine, value.var = "variable")
  }


  data <- x
  colname <- c()
  colname <- species
  data$CoOmXT_SWATH_DDA <- apply(data[,grep("*_CoOmXT_SWATH_DDA", colnames(data))], 1, function(x)sum(x, na.rm = TRUE))
  data$CoOmXT_Elite_DDA <- apply(data[,grep("*_CoOmXT_Elite_DDA", colnames(data))], 1, function(x)sum(x, na.rm = TRUE))
  data$MaxQuant <- apply(data[,grep("*_MaxQuant", colnames(data))], 1, function(x)sum(x, na.rm = TRUE))
  data <- melt(subset(data, select =c(value, CoOmXT_SWATH_DDA, CoOmXT_Elite_DDA, MaxQuant)), id="value")
  colnames(data)[3] <- colname[1]
  return(data)
}

count_sub_PHOSPHO_Protein <- count.species(sub_PHOSPHO_Protein, species = "phospho_proteins")
count_sub_PEPIDE <- count.species(sub_PEPIDE, species = "peptide")
count_sub_PHOSPHO <- count.species(sub_PHOSPHO, species = "phospho_peptide")
count_sub_DELOCALIZED <- count.species(sub_DELOCALIZED, species = "delocalized")
count_sub_DECOY <- count.species(sub_DECOY, species = "decoy")

count_merge <- Reduce(function(x,y) merge(x,y, all=TRUE), list(count_sub_PHOSPHO_Protein, count_sub_PEPIDE, count_sub_PHOSPHO, count_sub_DELOCALIZED,
count_sub_DECOY))
count_merge <- count_merge[order(count_merge$variable),]
count_merge$enrichment <- round(count_merge$phospho_peptide/count_merge$peptide, digits = 2)
colnames(count_merge)[colnames(count_merge) == "value"] <- "Sample_ID"
colnames(count_merge)[colnames(count_merge) == "variable"] <- "Engine"


annotation.file <- "Study_design_plotting.txt"
Study_design <- read.delim2(file.path(getwd(), annotation.file), dec=".", sep ="\t", header=TRUE)
count_merge <- merge(count_merge, Study_design, by = "Sample_ID")


write.table(count_merge, file ="count_merge_SWATH_DDA_and_Elite_DDA.tsv", sep = "\t", quote = FALSE, row.names = FALSE)


## Union of all deteced phosphoproteins
write.table(data_search_Elite, file ="data_Elite.tsv", sep = "\t", quote = FALSE, row.names = FALSE)

#########################################################################################################################################################
#                                                                                                                                                       #
#                 "Analysis mapDIA output of the BXD-mouse reference population samples"                                                                #
#                                                                                                                                                       #
# Author: Fabian Frommelt                                                                                                                               #
# Date: 29.04.2016 (last update)                                                                                                                        #
# Summary: Analysis of the phosphopeptide enriched samples of the BXD mouse reference population. We aimed to filter out phosphopeptides               #
#          which are regulated due to diet or genotype.                                                                                                 #
#########################################################################################################################################################


#1) Check for requantification values on the feature alignment level

library(reshape2)
library(ggplot2)
library(gridExtra)
library(stringr)

setwd("Y:\\20160202_First_Attempt_Analysis_of_BXD_SWATH/")

file.name <- "E1601291726_feature_alignment_requant.tsv"
data <- data.frame(read.table(file.name, sep = '\t', header= TRUE))

#########################################################################################################################
# Data are used to check for requant values within the frist and the second batch;
# As result, values are visualized in R, which have a low requant value in the first analysis batch,
# and a high requant value in the second MS-injections batch; (therefor the mean of the m-scores of the two batches is taken)

data.new <- data[,c("FullPeptideName", "ProteinName", "m_score", "transition_group_id", "filename")]

data_melt <- melt(data.new, id=c("transition_group_id","FullPeptideName", "ProteinName", "m_score"))
data_melt <- data_melt[,-5]

data_cast <- dcast(data_melt, transition_group_id + ProteinName + FullPeptideName ~ value, value.var = "m_score")

names(data_cast) <- gsub("_SW.mzXML.gz", "", names(data_cast))
```

```
names(data_cast) <- gsub("/scratch/[0-9]{7}.tmpdir/fabianf_L151223_", "", names(data_cast))


data_cast<-(data_cast[,order(colnames(data_cast),decreasing=FALSE)])
data_cast$mean_first <- apply(data_cast[,1:20], 1, function(x)mean(x, na.rm = TRUE))
data_cast$mean_second <- apply(data_cast[,56:76], 1, function(x)mean(x, na.rm = TRUE))

# a subset of all the peptides which have a mean m_score over 1 for the second batch of MS measurements
# further the data are sorted to the criteria if the mean m_score of the first 20 measurements of the
# first batch is below 0.01 (1%)

# 121 fit the given criteria; a few peptides are picked to load them into Skyline to check
# manually if there is a pick in the second acquistion batch or not; also some of the iRT-peptides
# are added to the transition list in Skyline to check them manually.

data_sec_requant <- subset(data_cast, mean_second > 1)
y <- (subset(data_sec_requant, mean_first > 1))
y <- (subset(data_sec_requant, mean_first < 0.01))

#############################################################################################################################
# 2) prepare the mapDIA output data for analysis
#

# inport the mapDIA output file to R

data_SWATH <-read.table("peptide_level.txt", header=TRUE, sep="\t", fill=TRUE, stringsAsFactors = FALSE)


# delocalize function which is also used in previous scripts.
delocalize.phospho <- function(data = dataframe())
{
  # http://stackoverflow.com/questions/19666965/count-pattern-matching-in-r
  # from this site I got the hint with the str_count command
  x <-data
  x$Count_Phospho <- sapply("(Phospho)", str_count, string =x$Peptide)
  x$Deloc <- x$Peptide
  x$Deloc <- sapply(x$Deloc,gsub,pattern="*\\(Phospho\\)*",replacement="")
  x$Deloc <- sapply(x$Deloc,gsub,pattern="*\\(Oxidation\\)*",replacement="")
  x$Delocalized <- paste(x$Deloc, x$Count_Phospho, sep="_P")
  x <- subset(x, select = -c(Deloc))
  x$Delocalized <- gsub(x$Delocalized, pattern = "NA_PNA", replacement = NA )
  x$Delocalized <- gsub(x$Delocalized, pattern = "\\_P0", replacement = "" )
  return(x)
}

# annotate the phospho sites correct for SWATH-MS output of mapDIA
mapDIA.PTM.annotation.to.iportal <- function(x=dataframe())
{
  x <- as.data.frame(sapply(x,gsub,pattern="*\\(UniMod_21\\)*",replacement="\\(Phospho\\)"), stringsAsFactors = FALSE)
  x <- as.data.frame(sapply(x,gsub,pattern="*\\(UniMod_35\\)*",replacement="\\(Oxidation\\)"), stringsAsFactors = FALSE)
  x <- as.data.frame(sapply(x,gsub,pattern="*\\(UniMod_4\\)*",replacement="\\(Carbamidomethyl\\)"), stringsAsFactors = FALSE)

}

data_SWATH <- mapDIA.PTM.annotation.to.iportal(data_SWATH)
data_SWATH <- delocalize.phospho(data_SWATH)
data_SWATH <- data_SWATH[,-79]

#############################################################################################################################
# 3) Calcualte HFD vs CD (exclude genetic background informtion; only take the diet into account)

Statistics_mean_FC <-
  function (data = dataframe()){
    data[,grep("([[:alnum:]]+)_([[:alpha:]]+)_([[:digit:]]+).*", colnames(data))] <- sapply(data[,grep("([[:alnum:]]+)_([[:alpha:]]+).*",
colnames(data))],as.numeric)
    data$mean.CD <- apply(data[,grep("_CD_.*", colnames(data))], 1, function(x)mean(x, na.rm = TRUE))
    data$mean.HFD <- apply(data[,grep("_HFD_.*", colnames(data))], 1, function(x)mean(x, na.rm = TRUE))
    data$mean.total <- apply(data[,grep("([[:alnum:]]+)_([[:alpha:]]+)_([[:digit:]]+).*", colnames(data))], 1, function(x)mean(x, na.rm =TRUE))
    data$sd.CD <- apply(data[,grep("_CD_.*", colnames(data))], 1, function(x)sd(x, na.rm = TRUE))
    data$sd.HFD <- apply(data[,grep("_HFD_.*", colnames(data))], 1, function(x)sd(x, na.rm = TRUE))
    data$sd.total <- apply(data[,grep("([[:alnum:]]+)_([[:alpha:]]+)_([[:digit:]]+).*", colnames(data))], 1, function(x)sd(x, na.rm =TRUE))
    data$CV.CD <- apply(data[,grep("_CD_.*", colnames(data))], 1, function(x) sd(x,na.rm = TRUE)/mean(x, na.rm = TRUE))
    data$CV.HFD <- apply(data[,grep("_HFD_.*", colnames(data))], 1, function(x) sd(x,na.rm = TRUE)/mean(x, na.rm = TRUE))
    data$CV.total <- apply(data[,grep("([[:alnum:]]+)_([[:alpha:]]+)_([[:digit:]]+).*", colnames(data))], 1, function(x) sd(x,na.rm = TRUE)/mean(x, na.rm =
TRUE))
    data$var.CD <- apply(data[,grep("_CD_.*", colnames(data))], 1, function(x)var(x, y=NULL, na.rm = TRUE))
    data$var.HFD <- apply(data[,grep("_HFD_.*", colnames(data))], 1, function(x)var(x, y=NULL, na.rm = TRUE))
    data$var.total <- apply(data[,grep("([[:alnum:]]+)_([[:alpha:]]+)_([[:digit:]]+).*", colnames(data))],  1, function(x)var(x, y=NULL, na.rm = TRUE))
    col_new <-colnames(data[,grep("([[:alnum:]]+)_([[:alpha:]]+)_([[:digit:]]+).*", colnames(data))])
    col_new <- paste(c("FC"), col_new ,sep = "_")
    data[col_new] <- NA
    data[,93:168] <- data[, 3:78]
    data[,grep("FC_*.", colnames(data))] <- data[,grep("FC_*.", colnames(data))] /data$mean.CD
    data[,grep("FC_*.", colnames(data))] <- sapply(data[,grep("FC_*.", colnames(data))], function(x)log2(x))
    return(data)
  }
data_SWATH_diet <- data_SWATH
data_SWATH_diet <- Statistics_mean_FC(data_SWATH_diet)

#############################################################################################################################
# 4) Data for CV plotting
# All data are collected within another script for plotting (all data of the different analysis methods)


sub.dSd<- subset(data_SWATH_diet, select = c("Protein", "CV.total", "CV.CD", "CV.HFD"))
sub.dSd <- melt(sub.dSd, id = "Protein")

#sub.dSd$ID <- factor(sub.dSd$variable, c("CV.total", "CV.HFD", "CV.CD"))

write.table(sub.dSd, file = "data_CV_BXD_reference.tsv", sep = "\t", quote = FALSE,col.names = TRUE, row.names = FALSE)


#############################################################################################################################
# 5) pairwise student test
#
#
# pairwise student test and adjusted p-value for the analysis of the influence of the
# factor diet and if we see diferently regulated phospho peptides due to diet

melt_dSd <- melt(data_SWATH_diet[,c(1, 2,grep("FC_.*", colnames(data_SWATH_diet)))], id=c("Protein", "Peptide"))
melt_dSd$Diet <- (gsub("([[:alnum:]])+_([[:alnum:]]+)_([[:alpha:]]+).*", "\\3", melt_dSd$variable))
melt_dSd$Mice <- (gsub("([[:alnum:]])+_([[:alnum:]]+)_([[:alpha:]]+).*", "\\2", melt_dSd$variable))

cast_dSd <- dcast(data = melt_dSd, Protein + Peptide + Mice ~ Diet, value.var = "value")
```

```
pvalue_table <- data.frame(Protein = unique(cast_dSd$Protein), effectsize = NA, pvalue = NA)
for(i in 1:nrow(pvalue_table)){
  print(i)
  protein <- pvalue_table[i, "Protein"]
  data.sel <- subset(cast_dSd, Protein == protein)
  pval <- t.test(data.sel$CD, data.sel$HFD, paired = TRUE)$p.value
  effect <- t.test(data.sel$CD, data.sel$HFD, paired = TRUE)$estimate
  pvalue_table[i, "pvalue"] <- pval
  pvalue_table[i, "effectsize"] <- effect

}
pvalue_table$effectsize <- pvalue_table$effectsize * -1
pvalue_table$pvalue.adj <- p.adjust(pvalue_table$pvalue, method = "BH")
head(pvalue_table[order(pvalue_table$pvalue),])
head(pvalue_table[order(pvalue_table$effectsize, decreasing=TRUE),])

pvalue_table_sort <- subset(pvalue_table, effectsize > 0.5 | effectsize < -0.5)
pvalue_table_sort <- subset(pvalue_table_sort, pvalue.adj < 0.01)



#########################################################################################################################################
# 5b) Plotting one of the phosphoproteins where Diet is the influence factor
#

pvalue_table[grep("Q01279", pvalue_table$Protein),]

data.sel <- melt_dSd[grep("Q01279_ELVEPLT\\(Phospho\\)PSGEAPNQAHLR", melt_dSd$Protein),]
data.sel <- subset(data.sel, Protein == "Q01279_ELVEPLT(Phospho)PSGEAPNQAHLR")



HFD_mean <- mean(subset(data.sel, Diet == "HFD")$value, na.rm = TRUE)
CD_mean <- mean(subset(data.sel, Diet == "CD")$value, na.rm = TRUE)
df2 <- data.frame(Diet =c("HFD", "CD"), m = c(HFD_mean, CD_mean))

g = ggplot(data=data.sel, aes(x=Mice, y=value)) +
  facet_wrap(facets = "Diet") +
  geom_point() +
  geom_hline(data=df2,aes(yintercept=m), color = "red" ) +
  geom_text(aes(label=Mice),hjust=0, vjust=0) +
  theme(legend.position="none",
        plot.title = element_text(size=25),
        axis.text.y=element_text(size=12),
        axis.title=element_text(size=14),
        axis.ticks = element_blank(), axis.text.x = element_blank(),
        axis.title.x = element_blank()) +
  labs( y = "log2FC") +
  labs(title = "Scatterplot Q01279 - ELVEPLT(Phospho)PSGEAPNQAHLR")
g


write.table(pvalue_table_sort, file = "statistics.tsv", sep = "\t", quote = FALSE,col.names = TRUE, row.names = FALSE)



#########################################################################################################################################
# 6) Calculate the correlation between HFD, CD



data_SWATH_melt<- melt(data_SWATH, id=c("Protein", "Peptide", "Count_Phospho", "Delocalized"))

data_SWATH_melt$Mice <- (gsub("([[:alnum:]]+)_([[:alpha:]]+).*", "\\1", data_SWATH_melt$variable))
data_SWATH_melt$Diet <- (gsub("([[:alnum:]]+)_([[:alpha:]]+).*", "\\2", data_SWATH_melt$variable))

data_SWATH_cor <- data_SWATH_melt[,-c(2:5)]
data_SWATH_cor$value <- sapply(data_SWATH_cor$value, as.numeric)

data_SWATH_cor <- dcast(data_SWATH_cor, Protein + Mice ~ Diet, value.var = "value")


# Finally, if use has the value "pairwise.complete.obs"
# then the correlation or covariance between each pair of variables is computed using
# all complete pairs of observations on those variables. This can result in covariance or
# correlation matrices which are not positive semi-definite, as well as NA entries if there
# are no complete pairs for that pair of variables.

cor <- data.frame(Protein = unique(data_SWATH_cor$Protein), Spearman = NA, Pearson = NA)
for(i in 1:length((cor$Protein))){
  print(i)
  protein <- cor[i, "Protein"]
  data.sel <- subset(data_SWATH_cor, Protein == protein)
  Sp_man <- cor(data.sel$CD, data.sel$HFD, method = "spearman", use = "pairwise.complete")
  Pear <- cor(data.sel$CD, data.sel$HFD, method = "pearson", use = "pairwise.complete")
  cor[i, "Spearman"] <- Sp_man
  cor[i, "Pearson"] <-Pear
}

cor <- (cor[order(cor$Spearman, decreasing=TRUE),])
cor_1 <- subset(x = cor, subset = cor$Spearman > 0.5)

write.table(cor_1, file = "genetically_regulated.tsv", sep = "\t", quote = FALSE,col.names = TRUE, row.names = FALSE)



x <- data_SWATH_cor[grep("^Q8VI47_KQS\\(Phospho\\)QSQDVLVLEDSK$", data_SWATH_cor$Protein),]

g = ggplot(data=x, aes(x=CD, y=HFD, label = Mice)) +
  geom_point(na.rm = TRUE) +
  #facet_wrap(facets = "Protein") +
  scale_y_continuous(trans="log10") +
  scale_x_continuous(trans="log10") +
  geom_text(aes(label=Mice),hjust=0, vjust=0) +
  theme(legend.position="none",
        plot.title = element_text(size=25)) +
  labs(x = "Intensity CD", y = "Intensity HFD") +
  labs(title = "Correlation of Q8VI47 - KQS(p)QSQDVLVLEDSK") +

  geom_smooth(method = "lm", se = FALSE, color = "red", formula = y ~ x)

g
```

```
#####################################################################################################################################################
#                                                                                                                                                   #
#                 "QTL analysis with the R/qtl package"                                                                                             #
#                                                                                                                                                   #
# Author: Evan Williams                                                                                                                             #
#                                                                                                                                                   #
# Summary: Mapping of phospho-pQTLs with a R-script written by Evan Williams and which had been used for the analysis of QTLs for other           #
#          other publications.                                                                                                                      #
#####################################################################################################################################################

setwd("/Users/wevan/Dropbox/Evan/R_FINAL_ALL/RQTL_v2/")
library(qtl)

sug <- read.cross("csv", , "QTL_Phosphoprot_HFD_INPUT.csv", genotypes=c("0", "1", "2", "9"), alleles=c("0", "1", "2", "9"))
phenonames = names(sug[[2]]) # Pulls out the names of all the phenotypes
Output_File = "PhosphoprotQTL_HFD_OUTPUT.xls" # The name of the output file for the QTL information


### NOTE: FOR NON-NORMAL-FORCED DATA, TEST FOR NORMALITY BEFORE DOING QTL MAPPING OR JUST USE MODEL=NP

########

sink(Output_File, append = TRUE)

for (i in 1:(length(phenonames)-1)) {
  sink()
  print(c(i, phenonames[i])) # For monitoring progress, can comment out.

  # Skips traits that have fewer than 8 strains' worth of data (minimum for QTL mapping)
  if(length(which(!is.na(sug[[2]][ , i]))) < 8) {
    chr=c("Insufficient data for QTL calculations")
    y=cbind(phenonames[i], chr)
    write.table(y,file = Output_File, append = TRUE, quote = TRUE, sep = "\t", eol = "\n", na = "NA", dec = ".", row.names = TRUE, col.names = NA, qmethod =
c("escape", "double"), fileEncoding = "")
    next
  }

  #out.i <- scanone(sug, pheno.col=i, method="hk")      # Calculates the QTL with NORMAL ASSUMPTION
  out.i <- scanone(sug, pheno.col=i, method="hk", model="np")   # Calculates the QTL with NO ASSUMPTIONS

  if(any(out.i$lod==Inf)) {     # This fixes a rare bug when some QTLs have infinite values
    chr=c("Skipped due to infinite LOD score")
    y=cbind(phenonames[i], chr)
    write.table(y,file = Output_File, append = TRUE, quote = TRUE, sep = "\t", eol = "\n", na = "NA", dec = ".", row.names = TRUE, col.names = NA, qmethod =
c("escape", "double"), fileEncoding = "")
    next
  }

  else {
    operm <- scanone(sug, pheno.col=i, method="hk", model="np", n.perm=1000) # Calculates the significance threshold for the QTL
    x=summary(out.i, perms=operm, alpha=.95, pvalues=TRUE) # Pulls out all markers with p values < alpha
  }

  if(identical(x[[2]],numeric(0))) { # If no results with alpha < the number selected above, initiate blank results; REQUIRED
    chr=c("No suggestive or significant results")
    pos=c("")
    names(chr)=c(" ")
    x=cbind(chr,pos)
  }

  y=cbind(phenonames[i], x[])   # Puts the phenotype name in the structure with the significant markers (for all)
  write.table(y,file = Output_File, append = TRUE, quote = TRUE, sep = "\t", eol = "\n", na = "NA", dec = ".", row.names = TRUE, col.names = NA, qmethod =
c("escape", "double"), fileEncoding = "")
}


#####################################################################################################################################################
#                                                                                                                                                   #
#                 "Summary plots of the CV in different MS-measurements and experiments"                                                            #
#                                                                                                                                                   #
# Author: Fabian Frommelt                                                                                                                           #
# Date: 04.12.2016                                                                                                                                  #
# Summary: The scirpt is vor comparision of the Coefficient of Variation of the SWATH and LFQ "Aging" results. Therefore the results were           #
#          saved in each of the scripts and were loaded again into this script.                                                                     #
#####################################################################################################################################################



setwd(dir = "Y:\\20160412_Coefficient_of_Variation_of_SWATH_and_Elite\\")

library(reshape2)
library(ggplot2)
library(gridExtra)


#############################################################################
# 1) Loading the SWATH outputs from the "BXD-mouse reference population" experiment, the
# total tissue lysis of the "aging" experiment and the phospho-SWATH MS analysis of the
# aging  experiment.

file.name <- "data_CV_BXD_reference.tsv"
data_BXD_ref <- read.table(file.path("Y:\\20160202_First_Attempt_Analysis_of_BXD_SWATH\\", file.name), header=TRUE, sep="\t", fill=TRUE, stringsAsFactors =
FALSE)

file.name <- "data_CV_aging_whole_tissue.tsv"
data_aging_whole <- read.table(file.path("Y:\\20160122_SWATH_analysis_total_cell_lysis_Aging\\", file.name), header=TRUE, sep="\t", fill=TRUE, stringsAsFactors
= FALSE)

file.name <- "data_CV_aging_phospho.tsv"
data_aging_phospho <- read.table(file.path("Y:\\20160129_phospho_SWATH_aging_new_library\\", file.name), header=TRUE, sep="\t", fill=TRUE, stringsAsFactors =
FALSE)

file.name <- "data_CV_aging_phospho_Elite.tsv"
data_aging_phospho_Elite <- read.table(file.path("Y:\\20160418_Aging_dataset_analysis_LFQ_and_SWATH_aging\\", file.name), header=TRUE, sep="\t", fill=TRUE,
stringsAsFactors = FALSE)



#############################################################################
# 2) Combining the data
```

```
data_aging_whole$variable <- gsub(x = data_aging_whole$variable, pattern = "CV.rep.SWATH", replacement = "CV.rep.SWATH.whole.lysate")
data_aging_whole$variable <- gsub(x = data_aging_whole$variable, pattern = "CV.total.SWATH", replacement = "CV.all.SWATH.whole.lysate")
data_aging_phospho$variable <- gsub(x = data_aging_phospho$variable, pattern = "CV.rep.SWATH", replacement = "CV.rep.phospho.SWATH")
data_aging_phospho$variable <- gsub(x = data_aging_phospho$variable, pattern = "CV.total.SWATH", replacement = "CV.all.phospho.SWATH")
data_aging_phospho_Elite$variable <- gsub(x = data_aging_phospho_Elite$variable, pattern = "CV.rep.Elite", replacement = "CV.rep.phospho.Elite")
data_aging_phospho_Elite$variable <- gsub(x = data_aging_phospho_Elite$variable, pattern = "CV.total.Elite", replacement = "CV.all.phospho.Elite")


names(data_BXD_ref)[names(data_BXD_ref) == "Protein"] <- "Peptide"


table.plot <- rbind(data_aging_whole, data_aging_phospho, data_aging_phospho_Elite, data_BXD_ref)
table.plot$ID <- factor(table.plot$variable, c("CV.total", "CV.HFD", "CV.CD", "CV.all.phospho.SWATH", "CV.rep.phospho.SWATH",
"CV.all.SWATH.whole.lysate","CV.rep.SWATH.whole.lysate",
                                               "CV.all.phospho.Elite", "CV.rep.phospho.Elite" ))
estimate_mode <- function(x) {

  d <- density(x, na.rm = TRUE)
  d$x[which.max(d$y)]
}



################################################################################################
# 3) Plotting the data

p <- ggplot(table.plot, aes(factor(ID), value)) +
  geom_violin(scale="area") +
  stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median,
               geom = "crossbar", width = 0.3) +
  #scale_y_continuous(trans="log10") +
  #geom_boxplot(width=0.1) +
  labs(title="Coefficient of Variation for peptide signal in BXD mouse reference population and Aging dataset ",
       x="", y="CV") +
  scale_x_discrete(labels=c("all.BXD", "HFD.BXD", "CD.BXD", "all.phospho- \n aging","rep.phospho- \n aging", "all.whole.lysate - \n aging", "rep.whole.lysate-
\n aging", "all.Elite.phospho \n aging", "rep.Elite.phospho \n aging")) +
  theme(axis.text = element_text(size = 15, angle = 45, hjust= 1, colour = "black"), axis.title = element_text(size = 20),
        plot.title = element_text(size = 22))
print(p)


aggregate(table.plot[,"value"], by=list(table.plot$variable), FUN = function(x) median(x, na.rm=TRUE))
aggregate(table.plot[,"value"], by=list(table.plot$variable), FUN = "estimate_mode")


################################################################################################
# 4) Plotting the BXD SWATH data

table.plot <- rbind(data_aging_whole, data_aging_phospho, data_BXD_ref)
table.plot$ID <- factor(table.plot$variable, c("CV.total", "CV.HFD", "CV.CD", "CV.all.phospho.SWATH", "CV.rep.phospho.SWATH",
"CV.all.SWATH.whole.lysate","CV.rep.SWATH.whole.lysate" ))


p <- ggplot(table.plot, aes(factor(ID), value)) +
  geom_violin(scale="area") +
  stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median,
               geom = "crossbar", width = 0.3) +
  #scale_y_continuous(trans="log10") +
  #geom_boxplot(width=0.1) +
  labs(title="Coefficient of Variation for phosphopetide intensities in the  \n BXD mouse reference population and the aging dataset ",
       x="", y="CV") +
  scale_x_discrete(labels=c("all.BXD", "HFD.BXD", "CD.BXD", "all.phospho- \n aging","rep.phospho- \n aging", "all.whole.lysate - \n aging", "rep.whole.lysate-
\n aging", "all.Elite.phospho \n aging", "rep.Elite.phospho \n aging")) +
  theme(axis.text.x = element_text(size = 15, angle = 45, hjust= 1, colour = "black"),
        axis.title = element_text(size = 20),
        axis.text.y = element_text(size =15, colour = "black"),
        plot.title = element_text(size = 22))
print(p)


################################################################################################
# 5) Plotting the Aging SWATH

table.plot <- rbind(data_aging_whole, data_aging_phospho)
table.plot$ID <- factor(table.plot$variable, c("CV.all.phospho.SWATH", "CV.rep.phospho.SWATH", "CV.all.SWATH.whole.lysate","CV.rep.SWATH.whole.lysate" ))


p <- ggplot(table.plot, aes(factor(ID), value)) +
  geom_violin(scale="area") +
  stat_summary(fun.y = median, fun.ymin = median, fun.ymax = median,
               geom = "crossbar", width = 0.3) +
  #scale_y_continuous(trans="log10") +
  #geom_boxplot(width=0.1) +
  labs(title="CV within the replicates and among all samples for the phosphopeptide enriched \n   and whole tissue lysate aging samples analyzed with SWATH-MS
",
       x="", y="CV") +
  scale_x_discrete(labels=c( "all.phospho- \n aging","rep.phospho- \n aging", "all.whole.lysate - \n aging", "rep.whole.lysate- \n aging")) +
  theme(axis.text.x = element_text(size = 15, angle = 45, hjust= 1, colour = "black"),
        axis.title = element_text(size = 20),
        axis.text.y = element_text(size =15, colour = "black"),
        plot.title = element_text(size = 22))
print(p)


# @ Fabian Frommelt - 05.06.2016
#################################################################################################################################################################################
```