Auswertung experimenteller Proteomics-Daten unter Verwendung der Analysesoftware SAS – Etablierung einer generellen Herangehensweise

Heiko Krause
Universitätsmedizin Greifswald KdöR
Klinik und Poliklinik für Psychiatrie und Psychotherapie
Ellernholzstr. 1/2
17475 Greifswald
Heiko.Krause@uni-greifswald.de

Zusammenfassung

Dieser Beitrag stammt aus meiner Abschlussarbeit im Rahmen der Ausbildung zum Medizinischen Dokumentar. Zur Verfügung standen gemessenen Daten von einem Massenspektrometer, welche bereits mittels der Software „Proteome Discoverer 1.4“ zusammengestellt wurden. Eine der Hauptaufgaben war, die gemessenen Daten zur weiteren Bearbeitung aufzubereiten. Die Herausforderung bestand darin, Daten unterschiedlicher Messungen zu ordnen, zu formatieren und so zu strukturieren, dass eine eindeutige Vergleichbarkeit innerhalb des Datensatzes als auch mit zukünftigen oder bereits vorhandenen Forschungsdaten gewährleistet werden kann.

Schlüsselwörter: Proteomics-Daten, Massenspektrometer, Odds ratio, Variablenberechnung, Normalisierung, Overlap, Venn Diagramm

1 Einleitung

1.1 Hintergrund

In der 3-jährigen Ausbildung zum Medizinischen Dokumentar muss das vierte und abschließende Praktikum mit einer schriftlichen Arbeit beendet werden. Dieses Hausarbeitsthema wurde am Leibniz-Institut für Plasmaforschung und Technologie e.V. (INP Greifswald) im Forschungsschwerpunkt Plasmamedizin erstellt. Die Messergebnisse eines Massenspektrometers, welches eine große Anzahl an Datensätzen von plasmabehandelten humanen, pflanzlichen, oder tierischen Zellen bzw. den daraus gewonnenen Proteinen liefert, wurden zunächst vereinfacht in MS Excel-Tabellen zusammengefasst und exportiert. Meine Aufgabe bestand darin, die im xls-Format vorliegenden Dateien in SAS einzulesen und in automatisierten Schritten in eine Struktur zu überführen, die die Lesbarkeit von Werten und Zeichenketten erhöht sowie eine grafische Darstellung der Daten implementiert. Zuzüglich sollte geprüft werden, inwieweit eine Integration von Transkriptom-Daten in die Tabellenstruktur möglich und handhabbar ist. Die Betreuung dieser Hausarbeit übernahm Herr Dr. Kristian Wende.

1.2 Datenentstehung

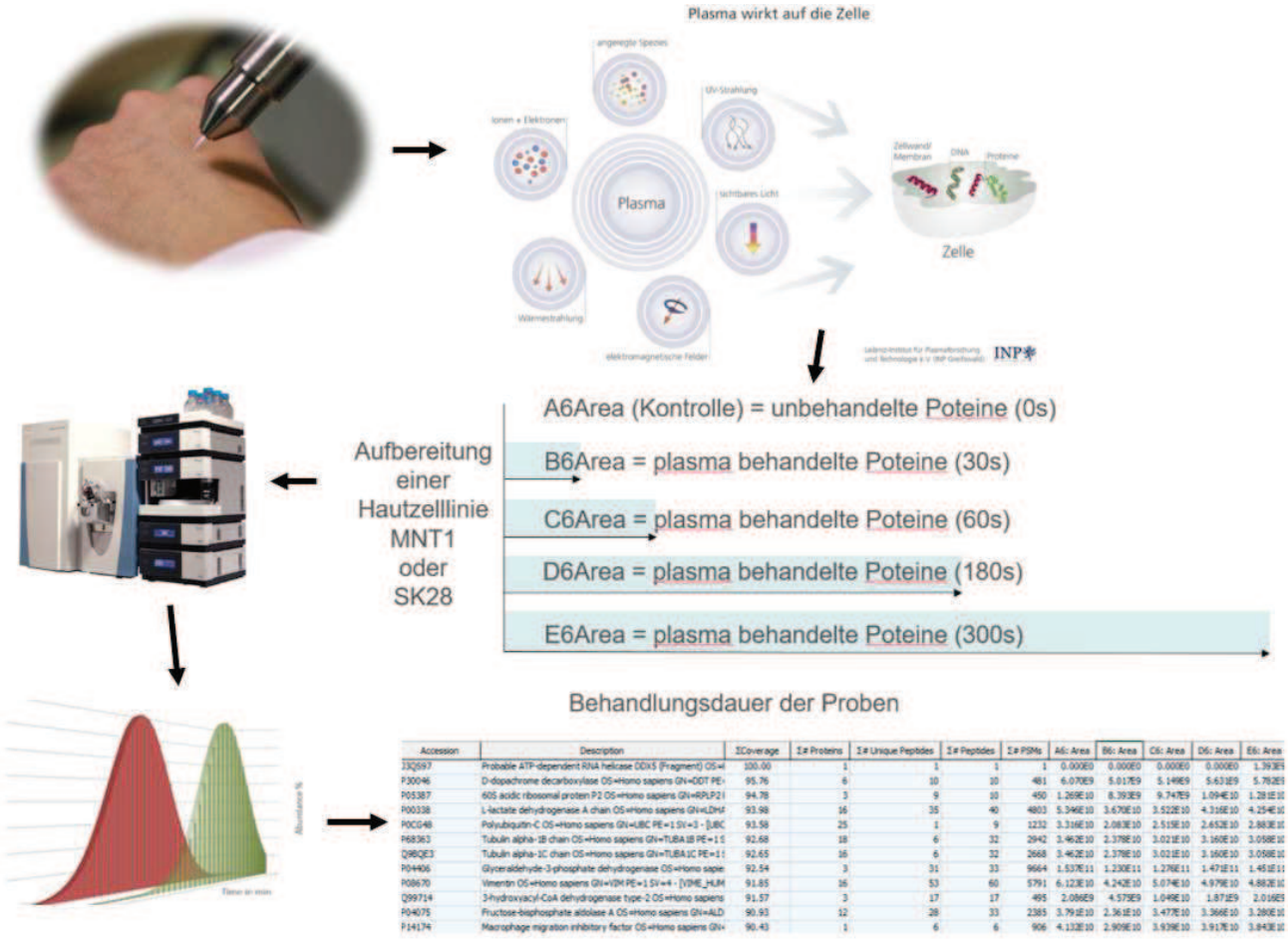


Abbildung 1: Skizzierung des Experiments und der Datenentstehung

Die mit aufgereinigten und mit Trypsin verdauten Proteine (=Peptide) der mit Plasma behandelten Zellen bzw. unbehandelten Kontrollen wurden in einen Hochleistungsflüssigkeitschromatographen (HPLC) über Pumpen in einen geschlossenen, luftdichten Kreislauf gebracht. Hier lösten sich die verschiedenen Peptidketten unterschiedlich schnell und wanderten als geladene Peptidionen in ein hochauflösendes Massenspektrometer (MS), wo sie anhand ihres spezifischen Masse-zu-Ladungsverhältnisses im Massenanalysator erfasst wurden (Abb. 2). Anschließend erfolgte eine Datenbankrecherche, bei der eine Zuordnung der gemessenen Massen zur Aminosäuresequenz und final den Peptiden bzw. den zugrundeliegenden Proteinen erfolgte. Über eine längere Zeit betrachtet entstand ein 3D Diagramm (Abb. 3), welches die Häufigkeit der gemessenen Peptidionen enthielt, sogenannte Integrale. Alle erwähnten und viele weitere Daten wurden mit anderen, in dieser Form gemessenen Proben, mit dem Proteome Disoverer 1.4 in einer Tabelle vereinigt und nach Excel exportiert (andere Exportoptionen waren nicht gegeben). Im Folgenden wurden die gemessenen Flächen der Proteine als „A6Area“, „B6Area“, „C6Area“, „D6Area“ und „E6Area“ bezeichnet (Abb. 1). Hierbei ist zu beachten, dass „A6Area“ die unbehandelten Proteine bezeichnet und nachstehend unter anderem „Kontrolle“ genannt werden.

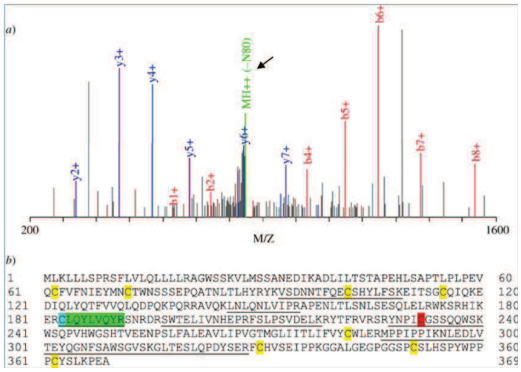


Abbildung 2: Sequenzierung

An Hand der mit dem Pfeil markierten eindeutigen Sequenz, die sich in Zeile 4 am Buchstabe 3-11 widerspiegelt, wird dieses Protein identifiziert, welches aus 369 Peptiden besteht. Die Höhe des Peaks gibt die Intensität des Proteinfragmentes an. (Biotin-Maleimid-Modifikation der Maus)

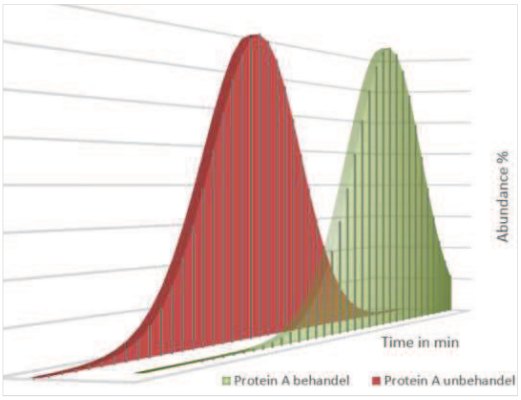


Abbildung 3: Zwei Integrale

Darstellung zweier Kurven (Integrale)
Das Protein A (der behandelten und unbehandelten Proben) wurde über die Zeit in unterschiedlicher Intensität (Abundance) nachweislich gefunden. Die Integrale der behandelten Proben werden folgend ins Verhältnis zur Kontrolle gesetzt und verglichen (Odds ratio (OR)).

1.3 Darstellung des Problems

Die humanen Hautzelllinien SK28 und MNT1 sollten auf Veränderungen der Proteinexpression durch den Einfluss von Plasma untersucht werden. Die vorliegenden Daten sind in tabellarischer Form z.T. sehr unübersichtlich (siehe Spalte „Description in Abb. 4). Die bisher etablierte Auswertepipline mit Excel war zeitintensiv und fehleranfällig. Eine grafische Darstellung (z.B. Mengenanalyse) war nur mittels verschiedener Online-Tools möglich, welche nicht stabil funktionierten.

Accession	Description
J3QS97	Probable ATP-dependent RNA helicase DDX5 (Fragment) OS=H
P30046	D-dopachrome decarboxylase OS=Homo sapiens GN=DDT PE=
P05387	60S acidic ribosomal protein P2 OS=Homo sapiens GN=RPLP2 I
P00338	L-lactate dehydrogenase A chain OS=Homo sapiens GN=LDHA
P0CG48	Polyubiquitin-C OS=Homo sapiens GN=UBC PE=1 SV=3 - [UBC

A6: Area	B6: Area	C6: Area	D6: Area	E6: Area
0.000E0	0.000E0	0.000E0	0.000E0	1.393E9
6.070E9	5.017E9	5.149E9	5.631E9	5.782E9
1.269E10	8.393E9	9.747E9	1.094E10	1.281E10
5.346E10	3.670E10	3.522E10	4.316E10	4.254E10
3.316E10	2.083E10	2.515E10	2.652E10	2.883E10

Abbildung 4: Teilausschnitte der unbearbeiteten Ursprungsdatei

Der Wunsch lag sehr nahe, alle einzelnen Schritte in einem Programm unterzubringen. Damit sollte gewährleistet werden, dass alle so gewonnenen Daten immer die gleichen Berechnungen durchlaufen und sich so keine manuellen Fehler einschleichen. Ebenso sollten alle Ausgabe- und Vergleichsmöglichkeiten abgedeckt werden, um entscheidende signifikante Ergebnisse schneller zu identifizieren und damit die Arbeit enorm zu vereinfachen.

Mit der SAS-Software, die zur Datenanalyse entwickelt wurde und in verschiedenen Kliniken, Unternehmen und der Forschung zur Betrachtung von komplexen Datensätzen zum Einsatz kommt, sollten die Messdaten analysiert und ein allgemein verwendbares Analyseprotokoll entwickelt werden. Begonnen wurde mit der Onlineversion „SAS OnDemand for Academics“, die im Verlauf der Arbeit von der „SAS University Edition“ abgelöst wurde. Das „SAS Studio“ ist eine Web-basierte Programmierumgebung und steht als Oberfläche für beide Versionen zur Verfügung.

2 Lösung des Problems

2.1 Einlesen und Bearbeitung der Daten

Vorab mussten in Excel, alle Sonderzeichen sowie Leerzeichen der ersten Zeile (Labels) entfernt werden, da diese sonst zu Bearbeitungsproblemen in SAS führen. Im Anschluss wurde die Datei in „SAS Studio“ hochgeladen und mit FILENAME der Dateipfad (in diesem Fall automatisch) zugeordnet, ebenso wurde mit der globalen LIBNAME Anweisung der Pfad zugewiesen, wo alle künftigen Dateien permanent abgelegt wurden.

2.1.1 Zerlegung der Variablen

Mit dem KEEP Statement wurden nur die Variablen eingelesen, die bearbeitet oder beibehalten werden sollten. Bei der Variable „Description“ wurde an bestimmten Stellen mit der Funktion TRANWRD ein Sonderzeichen eingesetzt, um den String an diesen Positionen mit der SCAN Funktion zu trennen. Dies wurde in den neu erstellten Variablen „SV_“, „PE_“, „GENENAME_“ und „SPECIES_“ ebenfalls nötig, um in den folgenden Abschnitten die Variablen mit der CATT Funktion zu verbinden und die Werte hinter dem „=“ zu Filtern sowie den richtigen Variablen zuzuordnen. In diesem Fall war die Reihenfolge (Tabelle von hinten begonnen) zur Variablenzerlegung und –zusammensetzung entscheidend, damit sich am Ende alle Werte in der richtigen Spalte wiederfanden. Dazu kam die Formatänderung zweier Spalten von Text zu Zahl mit der Funktion INPUT.

Das folgende Beispiel ist ein Teilausschnitt des gesamten DATA Step.

```
Data p.Multi_Reports;
Set p.import(keep=Accession Description Coverage Proteins...);
...
SV1=tranwrd(SV_,"SV=","sv*SV=");
SV2=tranwrd(PE_,"SV=","sv*SV=");
SV_3=scan(SV1,2,"*");
SV_4=scan(SV2,2,"*");
SV_n=catt(SV_3,SV_4);
SV5=scan(SV_n,2,"=");
...
SV=INPUT(SV5, best.);
PE=INPUT(PE5, best.);
run;
```

2.1.2 Variablenberechnung und -anordnung

Um die Proteine der Wertigkeit nach zu ordnen, wurde das Produkt der Variablen „PSMs“ und „Unique Peptides“ zum „PSMsScores“ gebildet und darauffolgend absteigend (descending) sortiert. Zudem wurden alle Variablen mit „RETAIN“ nach Relevanz für das Forscherteam angeordnet.

```
data P.Multi_Reports2 (keep=...);
 retain Accession Proteinname Species Genename ...;
 set P.Multi_Reports;
 PSMsScore=UniquePeptides*PSMs;
run;
proc sort data=P.Multi_Reports2;
 by descending PSMsScore;
run;
```

Accession	Description	ΣCovera
J3QS97	Probable ATP-dependent RNA helicase DDX5 (Fragment) OS=I	100.00
P30046	D-dopachrome decarboxylase OS=Homo sapiens GN=DDT PE	95.76
P05387	60S acidic ribosomal protein P2 OS=Homo sapiens GN=RPLP2	94.78

Accession	Proteinname	Species	Genename	PE	SV	Coverage	Proteins	UniquePeptides	Peptides	PSMs	AAs	MWkDa	calcpI	PSMsScore
Q09666	Neuroblast differentiation-associated protein AHNAK	Homo sapiens	AHNAK	1	2	75.3	6	243	254	3156	371	43.7	9.9204101563	766908
Q14204	Cytoplasmic dynein 1 heavy chain 1	Homo sapiens	DYNC1H1	1	5	54.11	3	204	209	2771	240	26.7	5.1987304688	565284
P49327	Fatty acid synthase	Homo sapiens	FASN	1	3	63.52	2	120	122	3445	298	33.0	9.7592773438	413400
P10809	60 kDa heat shock protein, mitochondrial	Homo sapiens	HSPD1	1	2	82.72	8	54	55	6301	978	106.7	8.0600585938	340254
Q9Y490	Talin-1	Homo sapiens	TLN1	1	3	72.92	7	116	137	2751	621	68.7	7.9575195313	319116

Abbildung 5: Tabelle nach der Zerlegung der Spalte Description sowie der Berechnung des PMSsScore

2.1.3 Erstellung der Normalisierungswerte

Um die behandelten Proben mit der Kontrolle vergleichen zu können, mussten die Flächen normalisiert werden. Hierzu wurden zwei Normalisierungsmethoden miteinander verglichen. Es wurde mit der PROC MEANS die Summen aller sowie der ersten 50 Observationen, welche nach dem „PSMsScore“ sortiert sind, der Variablen „A6Area“ bis „E6Area“ berechnet. Daraus ließen sich die Verhältnisse („VerhB“ bis „VerhE“) der behandelten Proben zur Kontrollfläche „A6Area“ beider Varianten ermittelt. Je näher die berechneten Verhältniswerte der 1 liegen, desto genauer wird die Normalisierung und weitere Berechnung. Wie zum Beispiel:

$$VerhB = \frac{\sum Fläche\ B}{\sum Fläche\ A}$$

Beide erstellten Tabellen wurden mit dem „SET“ Statement zusammengeführt. Der Mittelwert „Mean“ sollte zur Entscheidung der Normalisierungsmethode beitragen, aber

dieser konnte < 1 und > 1 sein, was die automatisierte Erkennung des kleinsten Abstandes zu 1 erschwerte (Abb. 6). Die Vergleichsvariable „Restzueins“ addiert den Absolutbetrag der Differenz zu 1 eines jeden Verhältniswertes und mittelt diese Summe.

$$\text{Restzueins} = \frac{\text{abs}(1 - \text{VerhB}) + \text{abs}(1 - \text{VerhC}) + \text{abs}(1 - \text{VerhD}) + \text{abs}(1 - \text{VerhE})}{4}$$

Anschließend wurde „Restzueins“ absteigend sortiert, um die zweite entscheidende Observation im Folgenden zu verwenden.

FREQ	VerhB	VerhC	VerhD	VerhE	Mean	Restzueins
6748	0.73525	1.17755	1.07914	1.06548	1.01436	0.14673
50	0.73866	0.84526	0.93726	0.92052	0.86043	0.13958

Abbildung 6: Erstellte Beispieltabelle. Je kleiner Restzueins (absteigend sortiert), desto genauer die Werte zur Normalisierung. An dieser Stelle wäre die Normalisierung anhand der 50 Top Proteine zu bevorzugen.

Das folgende Beispiel ist ein Teilausschnitt des beschriebenen Vorgehens:

```
proc means Data=P.Multi_Reports2 n sum noprint;
 Var A6Area B6Area C6Area D6Area E6Area;
 output out=P.verhges sum=A6Area B6Area C6Area D6Area E6Area;
run;

data P.MR_Verhges (keep=VerhB VerhC VerhD VerhE _Freq_);
 set P.verhges;
 VerhB=B6Area/A6Area;
 VerhC=C6Area/A6Area;
 VerhD=D6Area/A6Area;
 VerhE=E6Area/A6Area;
run;

...

data P.MR_Uebersicht_berg;
 set P.MR_Verhges P.MR_Verh50;
 Mean=(VerhB+VerhC+VerhD+VerhE)/4;
 Restzueins=(abs(1-VerhB)+abs(1-VerhC)+abs(1-VerhD)+abs(1-VerhE))/4;
run;

proc sort data=P.MR_Uebersicht_berg;
 by descending Restzueins;
run;
```

2.1.4 Normalisierung und Odds Ratio der Integrale

Es wurde die Variable „Accession“, die den Proteinnamen enthält, der bereinigten Datei mit PROC SQL gezählt und in der Makrovariable „N“ abgelegt. Aus der Tabelle mit

den Normalisierungswerten wurde nur die zweite Beobachtung eingelesen und in einer DO Schleife von 1 bis zum gezählten „N“ vervielfältigt. Die Vereinigung der Tabellen geschieht mit MERGE. Dabei werden beide Tabellen nebeneinandergelegt und somit für die weitere Berechnung verbunden. Die Normalisierung ergibt sich, indem man die Flächen durch die zugehörige Verhältniszahl dividiert, wie das folgende Beispiel zeigt.

$$Normierung = \frac{gemessene\ Proteine\ B}{VerhB}$$

Odds ratios (OR) sind die Schätzer für das relative Risiko und treffen Aussagen über die Chance, dass etwas zutrifft, um nach der sogenannten „Nadel im Heuhaufen“ zu suchen. Es wurden die vergleichbaren Integrale der behandelten Proteine durch das der Kontrolle dividiert. Die entstehenden OR geben das Vielfache der Veränderung an. Weicht das behandelte Protein in Größe seiner Fläche stark ab, so erhält man eine signifikante OR – unsere mögliche „Nadel“ – egal ob sie grösser 2.0 oder kleiner 0.5 ist.

$$OR = \frac{(normierte)\ Flächen\ B}{Fläche\ A}$$

Das folgende Beispiel ist ein Teilausschnitt des beschriebenen Vorgehens:

```
proc sql noprint;
 SELECT count (accession) into: N
 FROM P.Multi_Reports2;
quit;

data P.Multi_Reports4;
 set P.MR_Uebersicht_berg (FIRSTOBS=2);
 do var=1 to &N;
 output;
 end;
run;

...

data P.multi_reports3;
 set P.multi_reports3;
 RgB6Area=B6Area/VerhB;
 RgC6Area=C6Area/VerhC;
 RgD6Area=D6Area/VerhD;
 RgE6Area=E6Area/VerhE;
 OR_B=RgB6Area/A6Area;   OR_B= Round(OR_B,0.0001);
 OR_C=RgC6Area/A6Area;   OR_C= Round(OR_C,0.0001);
 OR_D=RgD6Area/A6Area;   OR_D= Round(OR_D,0.0001);
 OR_E=RgE6Area/A6Area;   OR_E= Round(OR_E,0.0001);
run;
```

2.1.5 Datenbereinigung der Isoformen mit ARRAY und SQL

Eine Vielzahl der Proteine besitzen sogenannte Isoformen, also strukturell abweichende, aber sehr nah verwandte Moleküle, die mit einem Bindestrich und eine zusätzliche Zahl am Proteinnamen versehen sind. Die Werte dieser Proteine und ihre Isoform(en) wiesen Missings, gleiche oder sich unterscheidende Werte auf (Abb. 7). Ziel war es, die fehlenden Messwerte dem Protein zuzuordnen, das schon die meisten Messwerte besaß und die Proteine sowie deren Isoformen zu behalten, welche in nur einer Variable einen unterscheidenden Messwert besaßen (Abb. 8).

Name	D1	E1	B2	C2	B3	C3
O00139	.	.	.	.	1.23	1.58
O00139-2	1.69	2.49	0.94	0.80	.	.
O00429	0.84	0.94	0.99	1.02	1.20	0.85
O00429-4	.	.	0.99	1.02	.	.
O00429-2	0.84	0.94	.	.	1.20	0.85
O94925-3	1.64	1.76	0.78	0.81	.	.
O94925	1.52	1.41	0.92	0.87	10.00	10.00

Abbildung 7: (erstelltes Beispiel) Namen mit Bindestrich sind die Isoformen mit den auffällig gleichen ORs oder, wie die letzten beiden Obs zeigen, ungleiche Werte

Name	_D1	_E1	_B2	_C2	_B3	_C3
O00139-2	1.69	2.49	0.94	0.80	1.23	1.58
O00429	0.84	0.94	0.99	1.02	1.20	0.85
O94925	1.52	1.41	0.92	0.87	10.00	10.00
O94925-3	1.64	1.76	0.78	0.81	.	.

Abbildung 8: Die ORs wurden dem richtigen Namen zugeordnet oder, wie die letzten beiden Observationen zeigen, sind erhalten geblieben

Diesbezüglich wurde eine Tabelle erstellt mit allen relevanten Angaben und der neuen Variable „Name_corpus“, die nur die Proteinhauptnamen ohne die Isoform enthält (SCAN Funktion). Mit RENAME wurden die Variablennamen zu B1 bis E1 vereinfacht.

```
 Name_corpus=scan(accession,1,'-');
 rename OR_B=B1;
```

In einem Array wurde die Anzahl der fehlenden Werte einer jeden Beobachtung ermittelt und einer Spalte „anzMissing“ zugewiesen und sehr wichtig, absteigend sortiert.

```
data missing;
 set mr_nodup;
 array vars(*) B1--E1;
 anzMissing = cmiss(of vars[*]);
run;
```

Es wurde ein ARRAY „newvars“ für die neuen und „old“ für die bekannten Variablen erstellt. In einer IF-THEN-DO Schleife wurden die neuen Variablen mit den alten aufgefüllt bzw. überschrieben und damit die richtigen Missings eines jeden Proteins besetzt.

```
data want;
 set missing;
 by Name_corpus;
 retain _B1 _C1 _D1 _E1;
 array newvars {*} _B1 _C1 _D1 _E1;
 array old {*} B1 C1 D1 E1;
 if first.name_corpus
 then do i = 1 to dim(newvars);
   newvars{i} = old{i};
 end;
 else do i = 1 to dim(newvars);
   if missing(newvars{i}) then newvars{i} = old{i};
 end;
 if last.name_corpus then output;
run;
```

Dieser Vorgang hatte zur Folge, dass die Proteine und Isoformen mit den unterschiedlichen Werten ebenfalls überschrieben bzw. ignoriert wurden. Dafür wurden folgend alle alten und neuen Variablen sowie die „anzMissing“ < 1 verglichen und gefiltert.

```
data wantnew;
 set want (where=(B1--E1 ne _B1--_E1 and anzmissing<1));
run;
```

Damit erhielt man alle entsprechend betroffenen Proteinnamen, die wiederum in einer PROC SQL Anweisung verwendet wurden, um alle gleichnamigen Proteine in der Ausgangdatei zu finden. Die Variablennamen dieser Datei „Wichtig“ wurden mit RENAME angepasst. Anschließend wurden beide Dateien nach der Variable „Accession“ sortiert, bevor sie anhand dieser Schlüsselvariable, im MERGE Statement vereinigt wurden. Die vereinfachten Spaltennamen wurden wieder zurückformatiert, womit der Bereinigungsvorgang der Isoformen abgeschlossen war.

```
proc Sql;
 create table Wichtig as
 select missing.*
 from missing, wantnew
 where missing.name_corpus=wantnew.name_corpus;
quit;
...

proc sort data=wantku;  by Accession; run;
proc sort data=wichtig; by Accession; run;

data ziel (drop=name_corpus);
 merge wantku wichtig;
 by Accession;
run;
...
```

2.1.6 Signifikante Zuweisungen per IF-THEN-DO

Wenn einzelne Proteine in der Kontrolle nicht identifiziert wurden (Konzentrationsproblem), zeigten sie keine Messwerte auf und wurden von der Software auf „Null“ gesetzt. Wird der Messwert einer Probe durch „Null“ dividiert, erhält man *per definitionem* keine (bzw. nicht erlaubte) Ergebnisse. Dennoch musste davon ausgegangen werden, dass einzelne Flächen der behandelten Proben möglicherweise Werte beinhalteten, die dergestalt vernachlässigt würden. Selbiges ergibt die inverse Betrachtung, wenn keine Messwerte durch die Kontrollfläche mit Messwert geteilt wurden, die OR ist Null. Diese Überlegung sorgte dafür, den zugehörigen ORs einen signifikanten Wert zuzuweisen, damit diese im Verlauf möglicherweise aktiv werdenden Proteine, entsprechend bedacht werden können.

Mit dem IF-THEN-DO Statement wurden diejenigen ORs, bei denen die Kontrollfläche eine Null aufzeigt und die gemessenen Proben einen Wert besaßen, mit dem frei gewählten Wert 10 versehen, aber nur unter der Bedingung, dass an mind. zwei aufeinanderfolgenden Zeitpunkten das Protein gemessen wurde. Die einzige Ausnahme bildete die Kombination der Flächen „C“ und „D“. In einem weiteren DATA Step wurde den beiden ORs, die der längsten Behandlung unterlagen und eine Null besaßen („D“ und „E“), während die Kontrolle einen positiven Wert enthält, im selbigen Vorgehen der Wert 0.1 zugewiesen.

```
data P.MR_OR10;
 set P.ziel;
 if A6Area<=0 and B6Area ne 0 and C6Area ne 0 and D6Area ne 0 and
E6Area ne 0
 then do OR_B=10; OR_C=10;OR_D=10;OR_E=10; end;
...
 if A6Area<=0 and C6Area ne 0 and D6Area ne 0 and E6Area ne 0
 then do OR_C=10;OR_D=10;OR_E=10; end;
run;

data P.MR_OR01;
 set P.MR_OR10;
 if A6Area > 0 and B6Area > 0 and C6Area > 0 and D6Area=0 and
E6Area=0
 then do OR_D=0.1; OR_E=0.1; end;
run;
```

2.1.7 Filtern mit WHERE

Damit diese Proteine sofort zu finden sind, wurden mit dem WHERE Statement alle ORs mit dem Wert 10 und 0.1 gefiltert und in je einer Tabelle zusammengestellt. Ebenso wurden den signifikanten Proteinen separate Tabellen zugewiesen sowie alle Proteine in auf- und absteigender Form aufgelistet. Dabei wurden Beobachtungen die keine Werte besaßen mit DELETE gelöscht.

Das folgende Beispiel ist ein Teilausschnitt des beschriebenen Vorgehens:

```
data P.MR_ORF;
 set P.mr_or01;
 where (OR_B>=2.0 or OR_B<=0.5) or (OR_C>=2.0 or OR_C<=0.5) or
(OR_D>=2.0 or OR_D<=0.5) or (OR_E>=2.0 or OR_E<=0.5);
 if OR_B=. and OR_C=. and OR_D=. and OR_E=. then delete;
run;
...

data P.MR_OR_desc4;
 set P.mr_orf_w (where=(OR_E<OR_D and OR_D<OR_C and OR_C<OR_B));
run;
```

Accession	OR_B	OR_C	OR_D	OR_E
H7C233	0.7183	0.6639	0.3499	0.3486
I6L894	3.0552	0.9295	0.2577	0.1100
O00422	6.3578	6.1132	4.9589	1.0588
O00425	1.5980	0.9829	0.6068	0.4753
O00566	2.0513	1.9557	1.0996	0.9185

Abbildung. 9: Teilausschnitt der Tabelle MR_OR_desc4 mit der absteigenden Anordnung der OR zu den am längsten behandelten Proteinen

2.1.8 Zählung der Overlaps (OL)

Die Overlaps der signifikanten Proteine wurden gezählt und in Form einer Tabelle sowie grafisch ausgegeben. Dafür wurden zwei Dateien erstellt, die nur den Proteinnamen und je ihre signifikanten ORs enthielten. Die nicht signifikanten ORs bekamen mit IF-THEN einen Punkt (.) zugewiesen und leere Zeilen wurden gelöscht, damit nur betroffene Proteine in die Zählung fallen. Mit RENAME wurden zudem die Variablennamen vereinfacht (A bis D). Verschiedene Makrovariablen mit Anzahl der Proteine wurden zur Berechnung erstellt. Die Zahlzuweisung erfolgte mit IF-THEN-ELSE, während die Mengenermittlung mit PROC UNIVARIATE geschah und zur weiteren Berechnung wiederum in einer Tabelle abgelegt wurde. Es wurde eine Variable „Obs“ zur genauen Beschreibung eingearbeitet und mit RETAIN an die passende Stelle delegiert. Später wurden die Variablen der OL wieder mit RENAME umbenannt.

Beispiel zur Zählung anhand des zweifachen Overlap

```
data P.AreapercBC;
 set P.Areaperc;
 if A>0 and B>0     then AB=1; else AB=0;
 if A>0 and AB ne 1 then A1=1; else A1=0;
 if B>0 and AB ne 1 then B1=1; else B1=0;
run;

proc univariate data =P.AreapercBC noprint;
 var AB A1 B1;
 output out = P.apBC_sum sum = sum_BC sum = sum_B sum = sum_C;
```

```
run;

data P.apBC_sum;
 retain obs;
 set P.apBC_sum;
 obs='Overlap_Proteine';
 Inside = sum(sum_BC, sum_B, sum_C);
 Outside= &NOR-Inside;
run;
```

Dieses Prozedere wurde bis zum 4-fachen Overlap durchgeführt. Das Ergebnis wurde dann unter Zuhilfenahme von ARRAYS in eine prozentuale Angabe umgerechnet und entsprechend gerundet. Dabei wurde zusätzlich die Variable „Obs“ mit dem Hinweis „same_in_%“ versehen, um die Werte besser interpretieren zu können.

Beispiel für die Berechnung mit ARRAY

```
data ...(keep=Obs BCDE BCD BCE BDE CDE BC BD ..._Outside);
 set ...;
  Obs='same_in_%';
 array var{*} BCDE BCD BCE BDE CDE BC BD ..._Outside;
 array Zaehler{*} sum_BCDE--Outside;
 do i=1 to dim (var);
      var{i}=Zaehler{i}/(Inside+Outside)*100;
      var{i}=Round(var{i},0.01);
 end;
run;
```

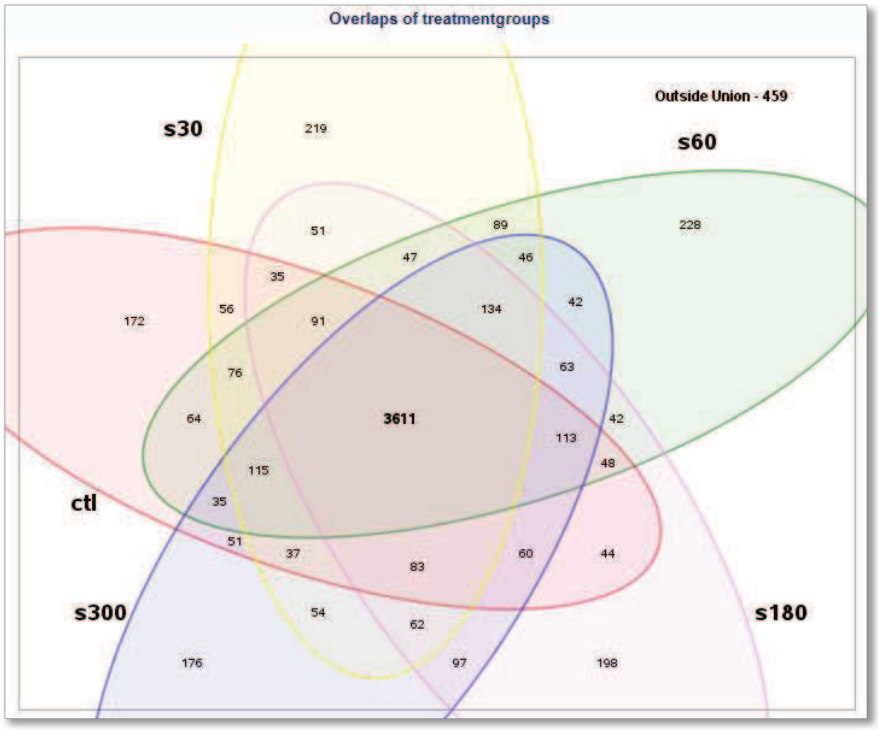
obs	BCDE	BCD	BCE	BDE	CDE	BC	BD	BE	CD	CE	DE	B	C	D	E	_Inside
Overlap_Proteine	3745.00	138.00	161.00	146.00	176.00	165.00	86.00	91.00	90.00	77.00	157.00	275.00	292.00	242.00	227.00	6067.00
same_in_%	61.73	2.27	2.65	2.39	2.90	2.72	1.42	1.50	1.48	1.27	2.59	4.53	4.81	3.99	3.74	100.00
Overlap_ORlt_0.5	68.00	12.00	16.00	44.00	33.00	47.00	16.00	49.00	27.00	39.00	121.00	160.00	112.00	99.00	249.00	1092.00
same_in_%	1.82	8.70	9.94	30.34	18.75	28.48	18.60	53.85	30.00	50.65	77.07	58.18	38.36	40.91	109.69	18.00
Overlap_ORgt_2	213.00	81.00	31.00	39.00	103.00	228.00	34.00	27.00	45.00	19.00	242.00	143.00	168.00	158.00	138.00	1669.00
same_in_%	5.69	58.70	19.25	26.90	58.52	138.18	39.53	29.67	50.00	24.68	154.14	52.00	57.53	65.29	60.79	27.51

Abbildung 10: Zeigt die Zählung und Berechnung der signifikanten OR inkl. der Overlaps aller OR. Dieser Überblick entstand mit dem PROC PRINT Statement inkl. des BLANKLINE=2, welche die Leerzeilen einfügt.

2.1.9 Graphische Darstellung im Venn Diagramm

Die Abbildung 11 zeigt ein Beispiel und illustriert, inwieweit sich Kontrolle und behandelte Proben unterscheiden. Sie umfasst das Ergebnis der Zählung für den 5-fachen Overlap und stellt die Anzahl der Proteine dar, die in gemeinsamen Behandlungen sowie der Kontrolle vorkommen. Die Grundsyntax dieses Makros, die aus nur 4 Overlaps bestand, ist aus der Quelle des „SAS Global Forum 2013“ und wurde von Kriss Harris vorgestellt [1]. In diesem Makro, welches um einen 5. OL und eine Überschrift TITLE erweitert wurde, besteht die Möglichkeit, vor jedem Programmdurchlauf die Anzahl der Overlaps, die Gruppenbezeichnung, Größe des CUTOFF sowie den Pfad und den Da-

teinamen zu bestimmen. Die Mengendiagramme wurden mit vier Makros in das Programm eingebettet. Es wurde die Zählung aller Integrale sowie ihre Angaben in Prozent und jeweils die Zählung der signifikanten OR erstellt (< 0.5 und > 2).



Die Anzahl an gemeinsamen Proteinen, die in bestimmten Behandlungsdauern auftauchen, werden in diesem Venn Diagramm grafisch dargestellt und sorgen damit für einen großen Wertgewinn. Es sind die Kontrolle (ctl) und die Behandlungen mit ihrer jeweiligen Dauer in Sekunden (s30 bis s300) dargestellt.

Abbildung 11: Venn Diagramm

Hiermit ist das Hauptprogramm abgeschlossen. Es analysiert das Vorkommen von Proteinen in unterschiedlich behandelten Proben aus sechs verschiedenen Datensätzen, welche alle einen separaten Speicherort erhielten.

2.2 Übersichtstabelle aller Datensätze

Es folgte ein neues SAS Skript in dem die Dateien der ungefilterten ORs extrahiert und verknüpft wurden. Dies diente dem Zweck des Datenvergleichs und Herausfilterns von besonders auffälligen Proteinen von mehreren Messungen. Dies geschieht über mehrere LIBNAME Statements, welche auf die verschiedenen Speicherorte der Proteinbehandlungen zugreifen und den gewünschten Datensatz einlesen. Zudem wurden die Variablennamen umbenannt, damit beim Vereinigen die Daten eindeutig blieben. Mit KEEP wurde nur die „Accession“ und „OR“ eingelesen. Es folgte die Sortierung jeder Datei nach der BY Variable „Accession“ und das Verschmelzen der Daten mit dem MERGE Statement (anhand der BY Variable). Proteine und Isoformen mit gleichen Werten, ebenso wie Observationen, die kleiner Null waren, wurden entfernt. Die Variablennamen wurden wieder in den Ursprung zurückgeführt. Somit sind die Variablen für potentielle Nutzer wieder eindeutig zu verstehen und die Dateien permanent unter neu angegebenem Pfad gespeichert.

Accession	MNT160608_30s	MNT160608_60s	MNT160608_180s	MNT160608_300s	MNT160720_30s	MNT160720_60s	MNT160720_180s	MNT160720_300s
A0AV96	.	.	.	.	.	.	.	.
A0AVI2	.	.	.	.	.	.	.	.
A0AVT1	1.4652	0.9476	2.1661	0.0000	0.8145	0.0000	1.2164	5.9630
A0FGR8	0.3978	0.0000	0.2174	0.0212	1.4262	0.6339	0.5562	0.0201

MNT160723_AR	MNT160723_19s	MNT160723_48s	MNT160723_89s	Ns_SK160715_30s	Ns_SK160715_60s	Ns_SK160715_180s	Ns_SK160715_300s
.	.	.	.	2.3204	2.3922	1.1590	0.3983
.	.	.	.	.	.	.	.
1.7345	0.0000	0.0000	0.0000	0.7673	0.7327	1.0906	0.7824
.	.	.	.	0.0000	0.0260	0.0165	0.0000

Ns_SK160723_AR	Ns_SK160723_19s	Ns_SK160723_48s	Ns_SK160723_89s	Ns_SK160724_30s	Ns_SK160724_60s	Ns_SK160724_180s	Ns_SK160724_300s
0.0952	0.0799	0.0691	0.0000	1.7138	1.1335	0.0625	1.3101
.	.	.	.	0.8602	0.9648	0.0000	0.6870
0.8260	0.8319	0.8087	0.6497	1.5784	0.6674	0.8877	1.2376
1.4271	1.2045	0.1000	0.1000	0.6154	0.0000	24.4294	0.0000

Abbildung 12: Ausschnitt der in drei Teile zerlegten Übersichtstabelle aller ORs von plasmabehandelten Proteinen der humanen Hautzelllinien SK28 und MNT1.

3 Fazit

Mit dieser Arbeit konnte gezeigt werden, dass jeder nach diesem Prinzip zusammengestellte Datensatz des Proteome Discoverers 1.4 nur durch Änderung der Globalanweisung in dieses SAS-Programm eingelesen und auf beschriebenen Weg analysiert werden kann. Jedes plasmabehandelte und unbehandelte Protein der humanen Hautzelllinien SK28 und MNT1, das von Interesse ist, kann in jedem möglichen Zusammenhang und durch zusätzliche Filtermöglichkeiten in SAS gefunden werden. Ebenfalls konnte das Ziel erreicht werden, auch Daten aus nicht-humanen Proben, die im MS entstehen und vom Proteome Discoverer 1.4 zusammengestellt wurden, wie zum Beispiel unterschiedlich behandelte Zellen der Alge *Chlorella vulgares*, den selben Programmdurchlauf zu ermöglichen. Zukünftige Programmschritte zur Integration von Transkriptom-Daten in die Tabellenstruktur der Proteom-Daten konnten bereits erstellt werden, sodass weitere Analysemöglichkeiten von Genen und Proteinen bestehen.
Die SAS Software ist somit für die Analyse von großen Datenmengen aus der Proteomforschung erfolgreich und zielführend einsetzbar. Eine generelle Herangehensweise konnte etabliert werden.

Literatur

[1] http://support.sas.com/resources/papers/proceedings13/243-2013.pdf (14.02.2017)