Supporting Information

Profiling the Serum Protein Corona of Fibrillar Human Islet Amyloid Polypeptide

Emily H. Pilkington,1,2 Ove J. R. Gustafsson,3 Yanting Xing,4 Juan Hernandez-Fernaud,5 Cleidi Zampronio,5 Aleksandr Kakinon,1 Ava Faridi,1 Feng Ding,4 Paul Wilson,1,2 Pu Chun Ke1* and Thomas P. Davis1,2*

1ARC Centre of Excellence in Convergent Bio-Nano Science and Technology, Monash Institute of Pharmaceutical Sciences, 381 Royal Parade, Parkville, VIC 3052, Australia
2Department of Chemistry, University of Warwick, Library Road, CV4 4AL, Coventry, UK
3ARC Centre of Excellence in Convergent Bio-Nano Science and Technology, Future Industries Institute, University of South Australia, University Boulevard, Mawson Lakes, SA 5095, Australia
4Department of Physics and Astronomy, Clemson University, Clemson, SC 29634, USA
5Warwick Proteomics Research Technology Platform, School of Life Sciences, Gibbet Hill Road, University of Warwick, CV4 7AL, Coventry, UK

Blue-Native polyacrylamide gel electrophoresis (PAGE) and analysis

Blue-Native PAGE allowed the examination of protein complex formation between IAPP amyloid fibrils (0.9 mg/mL) and FBS, which were pre-incubated (2% and 50%, in water) for 2 h. A 15 µL aliquot of each sample was mixed 1:1 with chilled Native PAGE sample buffer and then transferred to 4-15% gel (Mini-Protean TGX). Blue-Native PAGE was performed at 4 °C, pH = 8.3, using sequential buffer steps: briefly, Tris/Glycine buffer was supplemented with Coomassie Brilliant Blue G-250 at either 0.02%, 0.002%, or omitted, with each buffer solution utilized in this order during the assay. For protein binding capacity a densitometry of lane and well band profiles, normalized against background intensity, was performed using ImageJ. The experiment was performed and analyzed in duplicate. All materials listed were sourced from BioRad.
Circular dichroism spectroscopy

Structural changes in bovine serum albumin (BSA), the most abundant protein species in FBS medium, were examined upon its interaction with IAPP fibrils. BSA (0.1 mg/mL) was exposed to IAPP amyloid fibrils (>1 week old, in water; 0.1 mg/mL) for 24 h, and CD spectra of BSA were obtained at different timepoints using an Aviv Model 410 CD spectrophotometer (Biomedical, Inc.). The spectra were recorded over a wavelength range of 190~260 nm, with a 1 nm step size and a scanning speed of 15 nm/min at room temperature. The final spectra were baseline-corrected and IAPP amyloid spectrum subtracted where applicable. The data were measured in mean residue ellipticity (θ) and converted to the standard units deg·cm$^2$ dmol$^{-1}$.

nLC-MS/MS queries, analysis and informatics

Analysis of nLC-MS/MS data utilizing custom R scripts are as follows. Briefly, the combined summary files were used to plot peptide sequences identified, ratio of MS/MS identified: MS/MS submitted, the number of peaks and the mass standard deviation (ppm). Protein ID, protein name, gene name and sequence were extracted from the same uniprot reference proteome (Bos taurus) using seqinr and primary sequence dependent characteristics (GRAVY, pI, MW) were calculated using alakazam. Using protein ID as a key, the calculated values were extracted from the processed reference proteome for those proteins identified in the combined proteinGroup file to create a final data frame for plotting of identification overlaps and trends in GRAVY/pI/MW as well as amino acid composition of sequences for selected protein subsets (seqinr, ggplot2, gridExtra, VennDiagram).
**Figure S1.** Protein deposition on IAPP-functionalized QCM sensors (n = 4), as illustrated by frequency and dissipation shift after sequential introduction of IAPP amyloid and FBS.

**Figure S2.** Optimization of centrifugal capture (CC) methodology, demonstrating retention of IAPP amyloid on high molecular weight filter surface (n = 6) with only low nonspecific binding of FBS proteins (n = 8) after four spin-wash cycles. Error is SEM.
Figure S3a. LC-MS/MS chromatogram showing correlation between independent experiments of A (n = 6, overlaid).
Figure S3b. LC-MS/MS chromatogram showing correlation between independent experiments of AF (n = 6, overlaid).
Figure S3c. LC-MS/MS chromatogram showing correlation between independent experiments of F (n = 12, overlaid).
Figure S3d. LC-MS/MS chromatogram showing correlation between independent experiments of AE (n = 4, overlaid).
**Figure S3e.** LC-MS/MS chromatogram showing correlation between independent experiments of EF (n = 4, overlaid).
Figure S4. MS/MS spectrum identified by MaxQuant/Andromeda as a tryptic peptide of Amyloid Precursor Protein (APP, uniprot IDs Q28053, A4_BOVIN). Full sequence with identified peptide emphasized in red is overlaid onto the MaxQuant annotated MS/MS visualization, including matched fragments for the b and y ion series.
Figure S5a. The GRAVY/molecular weight (MW) relationships for unique amyloid-corona proteins (yellow markers, unique AF) and unique FBS-only proteins (blue markers, F) for CC experiments. All plots are overlaid onto the *Bos taurus* proteome background (gray). Top: points labelled are GRAVY ≤ -1.5 and ≥ 0.5, MW ≥ 6E+5. Bottom: points labelled are GRAVY (same as top) and MW ≥ 2E+5.
Figure S5b. The GRAVY/molecular weight (MW) relationships for unique amyloid-corona proteins (yellow markers, AE) and unique control proteins (blue markers, EF) for QCM experiments (AE/EF) are shown. Plots are overlaid onto the *Bos taurus* proteome background (gray). Top: points labelled are outside ranges GRAVY $\geq 0.5$ and GRAVY $\leq -1.5$, MW $\geq 3E+5$. Bottom: points labelled are outside ranges GRAVY $\geq 0.5$ and GRAVY $\leq -1.5$, MW $\geq 3E+5$. \[\]
Figure S6: Blue-Native PAGE demonstrates sequestering of FBS proteins through corona formation on IAPP amyloids. A: Representative gel of n = 2 runs. Lanes are as follows: (1) IAPP amyloid (0.9 mg/mL); (2) IAPP amyloid + 2% FBS; (3) 2% FBS control; (4) IAPP amyloid + 10% FBS; (5) 10% FBS control. B: Intensity profile of sample wells. C: Comparison of maximum intensity measured within sample lanes (excluding well); error = SEM.
Figure S7. Centrifugal capture (CC) method amyloid and serum corona (AF) STRING (string-db.org)\textsuperscript{7,8} protein network (molecular action) produced using database and experimental interactors, with a minimum interaction score of 0.400 and no additional interactors, against a whole \textit{Bos taurus} genome background. Enrichment analysis and molecular action legends are included, in addition to predicted action effects – positive (arrowhead), negative (endpoint line), unspecified (endpoint circle).
Figure S8a. Enriched gene ontology (GO) molecular function terms for proteins identified in AE (red) and AF (blue) experiments. Enrichment was performed using the STRING resource. Plot was generated in R using ggplot (see scatter plot code below). Point size is scaled to log(false.discovery.rate).

Figure S8b. Enriched gene ontology (GO) KEGG pathway terms for proteins identified in AE (red) and AF (blue) experiments. Enrichment was performed using the STRING resource. Plot was generated in R using ggplot (see scatter plot code below). Point size is scaled to log(false.discovery.rate).
Figure S8c. Enriched gene ontology (GO) molecular functions for *Catalytic activity* (green) and *hydrolase activity* (blue), with the KEGG pathway for *Complement and coagulation cascades* also included (red). Enrichment was performed using the STRING resource. Enrichment analysis and molecular action legends are included, in addition to predicted action effects – positive (arrowhead), negative (endpoint line), unspecified (endpoint circle).
**Figure S9. Structures of top AF proteins.** Alpha-actinin-4 (a), protein AMBP (b), neuropilin in either close (c) or open state (d), laminin subunit alpha 2 (orange) (e), SPARC (f), and IQ motif containing GTPase activating protein 1 (g) are shown in cartoon representations. The missing structures are represented by dashed lines.

**Figure S10. Structures of top AE proteins.** Serum albumin (a), thrombospondin-1 (b), and cartilage oligomeric matrix protein (c) are shown in cartoon representations. The missing structures are represented by dashed lines.
Figure S11. **Structures of top FBS proteins.** Serum albumin (a), catalytic domain of the ConC cGMP-specific 3′,5′-cyclic phosphodiesterase alpha-subunit (b), alpha-1-antiproteinase (c), Lactoperoxidase (d), NADH-ubiquinone oxido-reductase 75 from CroyEM (e), Hemiferrin (f), and thrombin domain of Prothrombin (g) are shown in cartoon representations.

**Fig S12.** (A) CD spectra of BSA (0.1 mg/mL) upon its interaction with IAPP amyloid fibrils (0.1 mg/mL) after 15 min of incubation. (B) Changing intensity of CD spectra of BSA interacting with IAPP fibrils at 208 nm (one of two signature peaks of alpha helices) up to 24 h of interaction.
Table S1. Top 10 unique amyloid serum coronae (AF) protein from the CC method. For a given protein, the PDB ID of its whole sequence or sub-domains (sequence indices are given in the parentheses), the number of total residues, and net charges at physiological conditions are listed.

<table>
<thead>
<tr>
<th>#</th>
<th>AF Protein</th>
<th>Protein ID</th>
<th>PDB ID</th>
<th># Residues</th>
<th>Net Charge</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alpha-actinin-4</td>
<td>A5D7D1</td>
<td>4D1E</td>
<td>911</td>
<td>-41</td>
</tr>
<tr>
<td>2</td>
<td>Antithrombin-III</td>
<td>P41361</td>
<td>1ATT</td>
<td>465</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Protein AMBP</td>
<td>P00978</td>
<td>4ES7 (28-193), 1BIK (230-339)</td>
<td>352</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>Neuripilin</td>
<td>E1BMX5</td>
<td>4GZ9</td>
<td>926</td>
<td>-24</td>
</tr>
<tr>
<td>5</td>
<td>Neuripilin</td>
<td>E1BEL6</td>
<td>2QQL</td>
<td>926</td>
<td>-39</td>
</tr>
<tr>
<td>6</td>
<td>Laminin subunit alpha 2</td>
<td>F1MMT2</td>
<td>5MC9 (2083-2687)</td>
<td>3085</td>
<td>-55</td>
</tr>
<tr>
<td>7</td>
<td>SPARC</td>
<td>P13213</td>
<td>1BMO</td>
<td>303</td>
<td>-28</td>
</tr>
<tr>
<td>8</td>
<td>Uncharacterized protein</td>
<td>F1MSA6</td>
<td>N/A</td>
<td>391</td>
<td>-18</td>
</tr>
<tr>
<td>9</td>
<td>IQ motif containing GTPase activating protein 1</td>
<td>F1MC48</td>
<td>3I6X (28-191) 3FAY (962-1345) 1X0H (1559-1657)</td>
<td>1600</td>
<td>-15</td>
</tr>
<tr>
<td>10</td>
<td>Collagen type VI alpha 2 chain protein</td>
<td>F1MKG2</td>
<td>1KUN (3108-3165)</td>
<td>1027</td>
<td>-7</td>
</tr>
</tbody>
</table>

Table S2. Top 10 unique amyloid serum coronae (AE) protein from the QCM method.

<table>
<thead>
<tr>
<th>#</th>
<th>AE Protein</th>
<th>Protein ID</th>
<th>PDB</th>
<th># Residues</th>
<th>Net Charge</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Serum albumin</td>
<td>A0A140T897</td>
<td>5UJB</td>
<td>607</td>
<td>-13</td>
</tr>
<tr>
<td>2</td>
<td>Thrombospondin-1</td>
<td>F1N3A1</td>
<td>2OUJ(11-214), 3R6B(433-547), and 1UX6(813-1152)</td>
<td>1170</td>
<td>-66</td>
</tr>
<tr>
<td>3</td>
<td>LDL receptor related protein 1</td>
<td>E1BGJ0</td>
<td>N/A</td>
<td>4544</td>
<td>-183</td>
</tr>
<tr>
<td>4</td>
<td>Vitamin K-dependent protein S</td>
<td>P07224</td>
<td>1Z6C (200-286)</td>
<td>675</td>
<td>-16</td>
</tr>
<tr>
<td>5</td>
<td>Uncharacterized protein</td>
<td>F1MJK3</td>
<td>N/A</td>
<td>1480</td>
<td>-5</td>
</tr>
<tr>
<td>6</td>
<td>Inter-alpha-trypsin inhibitor heavy chain H1</td>
<td>F1MMP5</td>
<td>N/A</td>
<td>906</td>
<td>-4</td>
</tr>
<tr>
<td>7</td>
<td>Thyroglobulin</td>
<td>F1MPH3</td>
<td>N/A</td>
<td>2769</td>
<td>-42</td>
</tr>
<tr>
<td>8</td>
<td>Uncharacterized protein</td>
<td>E1BKL9</td>
<td>N/A</td>
<td>4587</td>
<td>-245</td>
</tr>
<tr>
<td>9</td>
<td>Nucleobindin-1</td>
<td>Q0P569</td>
<td>1SNL (225-323)</td>
<td>474</td>
<td>-30</td>
</tr>
<tr>
<td>10</td>
<td>Cartilage oligomeric matrix protein</td>
<td>P35445</td>
<td>3FBY (225-756)</td>
<td>756</td>
<td>-59</td>
</tr>
</tbody>
</table>
Table S3. Top 10 FBS proteins from a LC-MS method.

<table>
<thead>
<tr>
<th>#</th>
<th>FBS protein name</th>
<th>PDB ID</th>
<th># Residues</th>
<th>Net Charge</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Serum albumin</td>
<td>5UJB</td>
<td>607</td>
<td>-13</td>
</tr>
<tr>
<td>2</td>
<td>Cone cGMP-specific 3′,5′-cyclic phosphodiesterase alpha-subunit</td>
<td>Catalytic domain: 3JWQ (536-855)</td>
<td>855</td>
<td>-23</td>
</tr>
<tr>
<td>3</td>
<td>Alpha-1-antiproteinase</td>
<td>1QLP (46-415) (human: 72%)</td>
<td>416</td>
<td>-9</td>
</tr>
<tr>
<td>4</td>
<td>Plasminogen</td>
<td>N/A</td>
<td>812</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>Lactoperoxidase</td>
<td>2IPS (118-712)</td>
<td>712</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>Kininogen, LMW II</td>
<td>N/A</td>
<td>619</td>
<td>-15</td>
</tr>
<tr>
<td>7</td>
<td>NADH-ubiquinone oxidoreductase 75</td>
<td>5O31 (24-727)</td>
<td>727</td>
<td>-9</td>
</tr>
<tr>
<td>8</td>
<td>Alpha-2-HS-glycoprotein</td>
<td>N/A</td>
<td>359</td>
<td>-13</td>
</tr>
<tr>
<td>9</td>
<td>Hemiferrin</td>
<td>1H76 (1-215) (rat: 70%)</td>
<td>216</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>Prothrombin</td>
<td>Thrombin domain: 1TBQ (318, 366, 367-625)</td>
<td>625</td>
<td>-8</td>
</tr>
</tbody>
</table>

Mean 594.8 -7.2

###libraries required###
library(ggplot2)
library(ggrepel)

###theme###
#see https://rstudio-pubs-static.s3.amazonaws.com/3364_d1a578f521174152b46b19d0c83cbe7e.html
themeTIFF <- theme(legend.text = element_text(size = 16), legend.title=element_text(size=16),
axis.title=element_text(size=18),
legend.key.size = unit(0.8, "cm"), axis.text.x = element_text(face = "bold", angle=0, vjust=0.5,
size=16),
axis.text.y = element_text(face = "bold", size=16), title = element_text(size=18))

###molecular function###
AE <- read.csv("./input/string_input/2018_03_05_STRING_analysis/AE/enrichment.Function.tsv", sep="\t",
header=TRUE)
AF <- read.csv("./input/string_input/2018_03_05_STRING_analysis/AF/enrichment.Function.tsv", sep="\t",
header=TRUE)

###KEGG###
AE_kegg <- read.csv("./input/string_input/2018_03_05_STRING_analysis/AE/enrichment.KEGG.tsv", sep="\t",
header=TRUE)
AF_kegg <- read.csv("./input/string_input/2018_03_05_STRING_analysis/AF/enrichment.KEGG.tsv", sep="\t",
header=TRUE)

###select columns and add ID###
AE <- AE[,c(2,3,4)]
AF <- AF[,c(2,3,4)]
AE$ID <- "AE"
AF$ID <- "AF"
AE_kegg <- AE_kegg[,c(2,3,4)]
AF_kegg <- AF_kegg[,c(2,3,4)]
AE_kegg$ID <- "AE"
AF_kegg$ID <- "AF"

###combined dataframes###
COMB <- rbind(AE, AF)
COMB_kegg <- rbind(AE_kegg, AF_kegg)

###plots###
tiff("./output/molecular_function_figure.tiff", height=900, width=1500)
a <- ggplot(COMB)
a <- a + geom_point(aes(reorder(pathway.description, observed.gene.count), observed.gene.count,
                        size = log(false.discovery.rate), fill=ID), pch=21, alpha=0.5)
a <- a + theme_bw() + themeTIFF + coord_flip()
a <- a + xlab("Pathway description") + ylab("Observed gene count")
plot(a)
dev.off()

tiff("./output/kegg_figure.tiff", height=700, width=1500)
a <- ggplot(COMB_kegg)
a <- a + geom_point(aes(reorder(pathway.description, observed.gene.count), observed.gene.count,
                        size = log(false.discovery.rate), fill=ID), pch=21, alpha=0.5)
a <- a + theme_bw() + themeTIFF + coord_flip()
a <- a + xlab("Pathway description") + ylab("Observed gene count")
plot(a)
dev.off()

MaxQuant parameter set 1 (Bos taurus):

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>1.6.0.16</td>
</tr>
<tr>
<td>User name</td>
<td>GUSTAFOJ</td>
</tr>
<tr>
<td>Machine name</td>
<td>IOR268965</td>
</tr>
<tr>
<td>Date of writing</td>
<td>02/19/2018 12:26:38</td>
</tr>
<tr>
<td>Fixed modifications</td>
<td>Carbamidomethyl (C)</td>
</tr>
<tr>
<td>Include contaminants</td>
<td>TRUE</td>
</tr>
<tr>
<td>PSM FDR</td>
<td>0.01</td>
</tr>
<tr>
<td>Protein FDR</td>
<td>0.01</td>
</tr>
<tr>
<td>Site FDR</td>
<td>0.01</td>
</tr>
<tr>
<td>Use Normalized Ratios For Occupancy</td>
<td>TRUE</td>
</tr>
<tr>
<td>Min. peptide Length</td>
<td>5</td>
</tr>
<tr>
<td>Min. score for unmodified peptides</td>
<td>0</td>
</tr>
<tr>
<td>Min. score for modified peptides</td>
<td>40</td>
</tr>
<tr>
<td>Min. delta score for unmodified peptides</td>
<td>0</td>
</tr>
<tr>
<td>Min. delta score for modified peptides</td>
<td>6</td>
</tr>
<tr>
<td>Min. unique peptides</td>
<td>0</td>
</tr>
<tr>
<td>Setting</td>
<td>Value</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>---------------------------------------------------------</td>
</tr>
<tr>
<td>Min. razor peptides</td>
<td>1</td>
</tr>
<tr>
<td>Min. peptides</td>
<td>1</td>
</tr>
<tr>
<td>Use only unmodified peptides and</td>
<td>TRUE</td>
</tr>
<tr>
<td>Modifications included in protein quantification</td>
<td>Oxidation (M); Acetyl (Protein N-term)</td>
</tr>
<tr>
<td>Peptides used for protein quantification</td>
<td>Razor</td>
</tr>
<tr>
<td>Discard unmodified counterpart peptides</td>
<td>TRUE</td>
</tr>
<tr>
<td>Label min. ratio count</td>
<td>2</td>
</tr>
<tr>
<td>Use delta score</td>
<td>FALSE</td>
</tr>
<tr>
<td>iBAQ</td>
<td>FALSE</td>
</tr>
<tr>
<td>iBAQ log fit</td>
<td>FALSE</td>
</tr>
<tr>
<td>Match between runs</td>
<td>FALSE</td>
</tr>
<tr>
<td>Find dependent peptides</td>
<td>FALSE</td>
</tr>
<tr>
<td>Fasta file</td>
<td>E:\MaxQuant\fasta_files\2018_02_19\UP000009136_9913.fasta</td>
</tr>
<tr>
<td>Decoy mode</td>
<td>revert</td>
</tr>
<tr>
<td>Include contaminants</td>
<td>TRUE</td>
</tr>
<tr>
<td>Advanced ratios</td>
<td>TRUE</td>
</tr>
<tr>
<td>Fixed andromeda index folder</td>
<td></td>
</tr>
<tr>
<td>Temporary folder</td>
<td></td>
</tr>
<tr>
<td>Combined folder location</td>
<td></td>
</tr>
<tr>
<td>Second peptides</td>
<td>TRUE</td>
</tr>
<tr>
<td>Stabilize large LFQ ratios</td>
<td>TRUE</td>
</tr>
<tr>
<td>Separate LFQ in parameter groups</td>
<td>FALSE</td>
</tr>
<tr>
<td>Require MS/MS for LFQ comparisons</td>
<td>TRUE</td>
</tr>
<tr>
<td>Calculate peak properties</td>
<td>FALSE</td>
</tr>
<tr>
<td>Main search max. combinations</td>
<td>200</td>
</tr>
<tr>
<td>Advanced site intensities</td>
<td>TRUE</td>
</tr>
<tr>
<td>LFQ norm for sites and peptides</td>
<td>FALSE</td>
</tr>
<tr>
<td>Write msScans table</td>
<td>TRUE</td>
</tr>
<tr>
<td>Write msmsScans table</td>
<td>TRUE</td>
</tr>
<tr>
<td>Write ms3Scans table</td>
<td>TRUE</td>
</tr>
<tr>
<td>Write allPeptides table</td>
<td>TRUE</td>
</tr>
<tr>
<td>Write mzRange table</td>
<td>TRUE</td>
</tr>
<tr>
<td>Write pasefMsmsScans table</td>
<td>TRUE</td>
</tr>
<tr>
<td>Write accumulatedPasefMsmsScans table</td>
<td>TRUE</td>
</tr>
<tr>
<td>Max. peptide mass [Da]</td>
<td>4600</td>
</tr>
<tr>
<td>Min. peptide length for unspecific search</td>
<td>8</td>
</tr>
<tr>
<td>Max. peptide length for unspecific search</td>
<td>25</td>
</tr>
<tr>
<td>Razor protein FDR</td>
<td>TRUE</td>
</tr>
<tr>
<td>Disable MD5</td>
<td>FALSE</td>
</tr>
<tr>
<td>Max mods in site table</td>
<td>3</td>
</tr>
<tr>
<td>Match unidentified features</td>
<td>FALSE</td>
</tr>
<tr>
<td>MS/MS tol. (FTMS)</td>
<td>20 ppm</td>
</tr>
<tr>
<td>Top MS/MS peaks per Da interval. (FTMS)</td>
<td>12</td>
</tr>
<tr>
<td>Da interval. (FTMS)</td>
<td>100</td>
</tr>
<tr>
<td>MS/MS deisotoping (FTMS)</td>
<td>TRUE</td>
</tr>
<tr>
<td>MS/MS deisotoping tolerance (FTMS)</td>
<td>7</td>
</tr>
<tr>
<td>Feature</td>
<td>Value</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>MS/MS deisotoping tolerance unit (FTMS)</td>
<td>ppm</td>
</tr>
<tr>
<td>MS/MS higher charges (FTMS)</td>
<td>TRUE</td>
</tr>
<tr>
<td>MS/MS water loss (FTMS)</td>
<td>TRUE</td>
</tr>
<tr>
<td>MS/MS ammonia loss (FTMS)</td>
<td>TRUE</td>
</tr>
<tr>
<td>MS/MS dependent losses (FTMS)</td>
<td>TRUE</td>
</tr>
<tr>
<td>MS/MS recalibration (FTMS)</td>
<td>FALSE</td>
</tr>
<tr>
<td>MS/MS tol. (ITMS)</td>
<td>0.5 Da</td>
</tr>
<tr>
<td>Top MS/MS peaks per Da interval. (ITMS)</td>
<td>8</td>
</tr>
<tr>
<td>Da interval. (ITMS)</td>
<td>100</td>
</tr>
<tr>
<td>MS/MS deisotoping (ITMS)</td>
<td>FALSE</td>
</tr>
<tr>
<td>MS/MS deisotoping tolerance (ITMS)</td>
<td>0.15</td>
</tr>
<tr>
<td>MS/MS deisotoping tolerance unit (ITMS)</td>
<td>Da</td>
</tr>
<tr>
<td>MS/MS higher charges (ITMS)</td>
<td>TRUE</td>
</tr>
<tr>
<td>MS/MS water loss (ITMS)</td>
<td>TRUE</td>
</tr>
<tr>
<td>MS/MS ammonia loss (ITMS)</td>
<td>TRUE</td>
</tr>
<tr>
<td>MS/MS dependent losses (ITMS)</td>
<td>TRUE</td>
</tr>
<tr>
<td>MS/MS recalibration (ITMS)</td>
<td>FALSE</td>
</tr>
<tr>
<td>MS/MS tol. (TOF)</td>
<td>40 ppm</td>
</tr>
<tr>
<td>Top MS/MS peaks per Da interval. (TOF)</td>
<td>10</td>
</tr>
<tr>
<td>Da interval. (TOF)</td>
<td>100</td>
</tr>
<tr>
<td>MS/MS deisotoping (TOF)</td>
<td>TRUE</td>
</tr>
<tr>
<td>MS/MS deisotoping tolerance (TOF)</td>
<td>0.01</td>
</tr>
<tr>
<td>MS/MS deisotoping tolerance unit (TOF)</td>
<td>Da</td>
</tr>
<tr>
<td>MS/MS higher charges (TOF)</td>
<td>TRUE</td>
</tr>
<tr>
<td>MS/MS water loss (TOF)</td>
<td>TRUE</td>
</tr>
<tr>
<td>MS/MS ammonia loss (TOF)</td>
<td>TRUE</td>
</tr>
<tr>
<td>MS/MS dependent losses (TOF)</td>
<td>TRUE</td>
</tr>
<tr>
<td>MS/MS recalibration (TOF)</td>
<td>FALSE</td>
</tr>
<tr>
<td>MS/MS tol. (Unknown)</td>
<td>0.5 Da</td>
</tr>
<tr>
<td>Top MS/MS peaks per Da interval. (Unknown)</td>
<td>8</td>
</tr>
<tr>
<td>Da interval. (Unknown)</td>
<td>100</td>
</tr>
<tr>
<td>MS/MS deisotoping (Unknown)</td>
<td>FALSE</td>
</tr>
<tr>
<td>MS/MS deisotoping tolerance (Unknown)</td>
<td>0.15</td>
</tr>
<tr>
<td>MS/MS deisotoping tolerance unit (Unknown)</td>
<td>Da</td>
</tr>
<tr>
<td>MS/MS higher charges (Unknown)</td>
<td>TRUE</td>
</tr>
<tr>
<td>MS/MS water loss (Unknown)</td>
<td>TRUE</td>
</tr>
<tr>
<td>MS/MS ammonia loss (Unknown)</td>
<td>TRUE</td>
</tr>
<tr>
<td>MS/MS dependent losses (Unknown)</td>
<td>TRUE</td>
</tr>
<tr>
<td>MS/MS recalibration (Unknown)</td>
<td>FALSE</td>
</tr>
<tr>
<td>Site tables</td>
<td>Oxidation (M)Sites.txt</td>
</tr>
</tbody>
</table>
MaxQuant parameter set 2 (*Homo sapiens*):

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>1.6.0.16</td>
</tr>
<tr>
<td>User name</td>
<td>GUSTAFOJ</td>
</tr>
<tr>
<td>Machine name</td>
<td>IOR268965</td>
</tr>
<tr>
<td>Date of writing</td>
<td>02/27/2018 14:12:31</td>
</tr>
<tr>
<td>Fixed modifications</td>
<td>Carbamidomethyl (C)</td>
</tr>
<tr>
<td>Include contaminants</td>
<td>TRUE</td>
</tr>
<tr>
<td>PSM FDR</td>
<td>0.01</td>
</tr>
<tr>
<td>Protein FDR</td>
<td>0.01</td>
</tr>
<tr>
<td>Site FDR</td>
<td>0.01</td>
</tr>
<tr>
<td>Use Normalized Ratios For Occupancy</td>
<td>TRUE</td>
</tr>
<tr>
<td>Min. peptide Length</td>
<td>5</td>
</tr>
<tr>
<td>Min. score for unmodified peptides</td>
<td>0</td>
</tr>
<tr>
<td>Min. score for modified peptides</td>
<td>40</td>
</tr>
<tr>
<td>Min. delta score for unmodified peptides</td>
<td>0</td>
</tr>
<tr>
<td>Min. delta score for modified peptides</td>
<td>6</td>
</tr>
<tr>
<td>Min. unique peptides</td>
<td>0</td>
</tr>
<tr>
<td>Min. razor peptides</td>
<td>1</td>
</tr>
<tr>
<td>Min. peptides</td>
<td>1</td>
</tr>
<tr>
<td>Use only unmodified peptides and</td>
<td>TRUE</td>
</tr>
<tr>
<td>Modifications included in protein quantification</td>
<td>Oxidation (M);Acetyl (Protein N-term)</td>
</tr>
<tr>
<td>Peptides used for protein quantification</td>
<td>Razor</td>
</tr>
<tr>
<td>Discard unmodified counterpart peptides</td>
<td>TRUE</td>
</tr>
<tr>
<td>Label min. ratio count</td>
<td>2</td>
</tr>
<tr>
<td>Use delta score</td>
<td>FALSE</td>
</tr>
<tr>
<td>iBAQ</td>
<td>FALSE</td>
</tr>
<tr>
<td>iBAQ log fit</td>
<td>FALSE</td>
</tr>
<tr>
<td>Match between runs</td>
<td>FALSE</td>
</tr>
<tr>
<td>Find dependent peptides</td>
<td>FALSE</td>
</tr>
<tr>
<td>Fasta file</td>
<td>E:\MaxQuant\fasta_files\2018_02_23\UP000005640_9606.fasta</td>
</tr>
<tr>
<td>Decoy mode</td>
<td>revert</td>
</tr>
<tr>
<td>Include contaminants</td>
<td>TRUE</td>
</tr>
<tr>
<td>Advanced ratios</td>
<td>TRUE</td>
</tr>
<tr>
<td>Fixed andromeda index folder</td>
<td></td>
</tr>
<tr>
<td>Temporary folder</td>
<td></td>
</tr>
<tr>
<td>Combined folder location</td>
<td></td>
</tr>
<tr>
<td>Second peptides</td>
<td>TRUE</td>
</tr>
<tr>
<td>Stabilize large LFQ ratios</td>
<td>TRUE</td>
</tr>
<tr>
<td>Separate LFQ in parameter groups</td>
<td>FALSE</td>
</tr>
<tr>
<td>Require MS/MS for LFQ comparisons</td>
<td>TRUE</td>
</tr>
<tr>
<td>Calculate peak properties</td>
<td>FALSE</td>
</tr>
<tr>
<td>Main search max. combinations</td>
<td>200</td>
</tr>
<tr>
<td>Advanced site intensities</td>
<td>TRUE</td>
</tr>
<tr>
<td>LFQ norm for sites and peptides</td>
<td>FALSE</td>
</tr>
</tbody>
</table>
Write msScans table        TRUE
Write msmsScans table       TRUE
Write ms3Scans table        TRUE
Write allPeptides table     TRUE
Write mzRange table         TRUE
Write pasefMsmsScans table  TRUE
Write accumulatedPasefMsmsScans table  TRUE
Max. peptide mass [Da]      4600
Min. peptide length for unspecific search  8
Max. peptide length for unspecific search  25
Razor protein FDR           TRUE
Disable MD5                  FALSE
Max mods in site table      3
Match unidentified features FALSE
MS/MS tol. (FTMS)           20 ppm
Top MS/MS peaks per Da interval. (FTMS)  12
Da interval. (FTMS)         100
MS/MS deisotoping (FTMS)    TRUE
MS/MS deisotoping tolerance (FTMS)  7
MS/MS deisotoping tolerance unit (FTMS)  ppm
MS/MS higher charges (FTMS) TRUE
MS/MS water loss (FTMS)     TRUE
MS/MS ammonia loss (FTMS)   TRUE
MS/MS dependent losses (FTMS) TRUE
MS/MS recalibration (FTMS)  FALSE
MS/MS tol. (ITMS)           0.5 Da
Top MS/MS peaks per Da interval. (ITMS)  8
Da interval. (ITMS)         100
MS/MS deisotoping (ITMS)    FALSE
MS/MS deisotoping tolerance (ITMS)  0.15
MS/MS deisotoping tolerance unit (ITMS)  Da
MS/MS higher charges (ITMS) TRUE
MS/MS water loss (ITMS)     TRUE
MS/MS ammonia loss (ITMS)   TRUE
MS/MS dependent losses (ITMS) TRUE
MS/MS recalibration (ITMS)  FALSE
MS/MS tol. (TOF)            40 ppm
Top MS/MS peaks per Da interval. (TOF)  10
Da interval. (TOF)          100
MS/MS deisotoping (TOF)     TRUE
MS/MS deisotoping tolerance (TOF)  0.01
MS/MS deisotoping tolerance unit (TOF)  Da
MS/MS higher charges (TOF)  TRUE
MS/MS water loss (TOF)      TRUE
MS/MS ammonia loss (TOF)    TRUE
MS/MS dependent losses (TOF) TRUE
**MS/MS recalibration (TOF)** FALSE
**MS/MS tol. (Unknown)** 0.5 Da
**Top MS/MS peaks per Da interval. (Unknown)** 8
**Da interval. (Unknown)** 100
**MS/MS deisotoping (Unknown)** FALSE
**MS/MS deisotoping tolerance (Unknown)** 0.15
**MS/MS deisotoping tolerance unit (Unknown)** Da
**MS/MS higher charges (Unknown)** TRUE
**MS/MS water loss (Unknown)** TRUE
**MS/MS ammonia loss (Unknown)** TRUE
**MS/MS dependent losses (Unknown)** TRUE
**MS/MS recalibration (Unknown)** FALSE

**Site tables** Oxidation (M)Sites.txt

Complete script used to generate Rmarkdown html report:

```{r setup, include=FALSE}

---

---

<title: "Supplementary data analysis methods - Profiling the Serum Protein Corona of Human IAPP Amyloid"
output:
html_document:
toc: true
toc_depth: 2
toc_float: TRUE
self_contained: no
number_sections: TRUE

```
###global settings###
knitr::opts_chunk$set(echo = TRUE, cache=TRUE, autodep=TRUE)

###installs###
#install.packages(c("ggplot2", "gridExtra",
#                   "seqinr", "alakazam",
#                   "sqldf", "stringr",
#                   "ggrepel", "VennDiagram",
#                   "rmarkdown", "knitr"))

###libraries###
library(gridExtra)
library(ggplot2)
library(seqinr)
library(alakazam)
library(sqldf)
library(stringr)
library(ggrepel)
library(VennDiagram)
library(reshape2)
update.packages()

### Introduction ###
The specific research questions addressed in this document include:

1. Which proteins can be identified in IAPP amyloid-coronae formed in-solution (CC method)?
2. Which proteins can be identified in IAPP amyloid-coronae formed under micro-flow conditions (QCM method)?
3. What is the difference in protein corona composition for CC and QCM methods?
4. What proteins are identified for IAPP amyloid-only experiments?

This document details the combination, processing and presentation of summary and proteinGroup text files resulting from MaxQuant Andromeda database queries, where each experiment type was pooled to maximise identifications (e.g. A1+A2+A3,...). Independent searches were performed for selected experiment types (e.g. AF, F).

### Analysis notes ###
## General information
- MaxQuant peptide identifications used a fasta file containing all reviewed and unreviewed Bos taurus entries available as a reference proteome @ ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/reference_proteomes/Eukaryota/.
- Reviewed + unreviewed entries included to maximise identifications.
- Database 1: UP000009136_9913.fasta dated 21/12/2017.
- Database 2: UP000005640_9606.fasta dated 21/12/2017.
- Whole proteome of Bos taurus as plotting background (same fasta as above, 9136_9913).
- nLC-MS chromatograms imported from a separate R script (provided at the end of this document).
- Protein network analysis images were imported from a manual analysis using the online resource STRING (v10.5).
- STRING resource link - see https://string-db.org/cgi/input.pl

## Plotting theme
```{r}
###set theme###
#see https://rstudio-pubs-static.s3.amazonaws.com/3364_d1a578f521174152b46b19d0c83cbe7e.html
themeTIFF <- theme(legend.text = element_text(size = 16), legend.title=element_text(size=16),
axis.title=element_text(size=18),
    legend.key.size = unit(0.8, "cm"), axis.text.x = element_text(face = "bold", angle=0, vjust=0.5, size=16),
axis.text.y = element_text(face = "bold", size=16), title = element_text(size=18))
```

```r
```
```
# Experiment key

<table>
<thead>
<tr>
<th>ID</th>
<th>Method</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>CC</td>
<td>Control - Amyloid only</td>
</tr>
<tr>
<td>F</td>
<td>CC</td>
<td>Control - FBS only</td>
</tr>
<tr>
<td>AF</td>
<td>CC</td>
<td>Amyloid + FBS</td>
</tr>
<tr>
<td>EF</td>
<td>QCM</td>
<td>Control - FBS only</td>
</tr>
<tr>
<td>AE</td>
<td>QCM</td>
<td>Amyloid + FBS</td>
</tr>
</tbody>
</table>

### Combined MaxQuant files

Function to combine MaxQuant summary files for analysis of the nLC-MS/MS data.

This analysis includes:

- Peptide sequences identified,
- Ratio of MS/MS identified : MS/MS submitted,
- The number of peaks, and
- The mass standard deviation (ppm)

```
```{r sum}
###set path###
path = "./input/IAPP"

###GETsummaries###
GETsummaries <- function(x){
  #see https://stackoverflow.com/questions/4876813/using-r-to-list-all-files-with-a-specified-extension
  sum_files <- list.files(path = x, pattern = "summary.txt", recursive=TRUE, include dirs=TRUE, ignore.case = TRUE)
  complete_summary <- data.frame()
  for (i in sum_files){
    #see https://stackoverflow.com/questions/8996134/extract-vectors-from-strsplit-list-without-using-a-loop
    Experiment <- rapply(strsplit(i, "/"), function(a) a[1])
    summary <- read.csv(paste(x, "/", i, sep=""), sep="\t", header=TRUE)
    summary <- summary[summary$Raw.file != "Total",]
    summary$Experiment <- Experiment
    id <- rapply(strsplit(as.character(summary$Raw.file), "."), function(a) a[4])
    exp <- rapply(strsplit(as.character(summary$Raw.file), "."), function(a) a[6])
    ID_EXP <- paste(id, exp, sep="-")
    summary$ID_EXP <- ID_EXP
    complete_summary <- rbind(summary, complete_summary)
  }
  return(complete_summary)
}
summaries <- GETsummaries(path)
```

## Summary file plot

```
```{r sumplot}
tiff(file=paste("./output/summary_plots.tiff", sep=""), width=600, height=1000)
#see https://www.statmethods.net/advgraphs/layout.html
a <- ggplot(summaries)
b <- a + geom_bar(stat="identity", aes(ID_EXP, Peptide.Sequences.Identified, fill=Experiment))
b <- b + expand_limits(y=0) + coord_flip() + xlab("Experimental ID") + theme_bw()
#see http://www.cookbook-r.com/Graphs/Legends_(ggplot2)/
b <- b + guides(fill=FALSE)
c <- a + geom_bar(stat="identity", aes(ID_EXP, MS.MS.Identified../MS.MS.Submitted, fill=Experiment))
c <- c + expand_limits(y=0) + coord_flip() + xlab("Experimental ID") + theme_bw() + guides(fill=FALSE)
d <- a + geom_bar(stat="identity", aes(ID_EXP, Mass.Standard.Deviation..ppm., fill=Experiment))
d <- d + expand_limits(y=0) + coord_flip() + xlab("Experimental ID") + theme_bw()
The number of peptide sequences identified (top left), ratio of MS/MS identified : MS/MS submitted (top right), the number of peaks (bottom left) and the mass standard deviation (ppm, bottom right) for the indicated shotgun nLC-MS/MS experiments.

```
\begin{center}
![](./output/summary_plots.tiff)
\end{center}
```

### proteinGroup files
Function to combine MaxQuant outputs for subsequent analyses.
The Experiment ID (see Experiment key) is added to the dataframe here to allow subsetting of the data.
```
```{r group}
GETgroups <- function(x){
  #see https://stackoverflow.com/questions/4876813/using-r-to-list-all-files-with-a-specified-extension
  prot_files <- list.files(path = x, pattern = "proteinGroups.txt", recursive=TRUE, include.dirs=TRUE, ignore.case=TRUE)
  complete_group <- data.frame()
  count <- 0
  for (i in prot_files){
    #see https://stackoverflow.com/questions/8996134/extract-vectors-from-strstrlist-without-using-a-loop
    Experiment <- rapply(strsplit(i, "/"), function(a) a[1])
    prot_group <- read.csv(paste(x, "/", i, sep="""""", sep=""""t", header=TRUE)
    prot_group$Experiment <- Experiment
    count <- count + nrow(prot_group)
    print(paste("Experiment ", Experiment, " identified ", count, " proteins", sep=""""""""""""""""""")
    count <- 0
    complete_group <- rbind(prot_group, complete_group)
  }
  return(complete_group)
}
protgroups <- GETgroups(path)
```

The unique Experiments included in the final proteinGroups data frame are `r unique(protgroups$Experiment)`.
The total identifications for each appear in the table below.
```
```{r}
###count Experiment-specific ID###
table(protgroups$Experiment)
```

# Uniprot database

## Load database

Here, the Bos taurus uniprot database file "UP000009136_9913.fasta" is loaded and the protein ID, protein name, gene name and sequence are extracted.
```
```{r up}
###read uniprot fasta file using seqinr function read.fasta###
UP <- read.fasta(file="./input/fasta/UP000009136_9913.fasta", seqtype="AA", as.string=TRUE)

###function to extract sequences and annotations from fasta files###
GetAnnotSeqFASTA <- function(x){
  UP_sequence <- unlist(getSequence(x, as.string=TRUE))
  UP_annotations <- unlist(getAnnot(x, as.string=TRUE))
  ProtName <- rapply(strsplit(UP_annotations, "IN "), function(a) a[2])
  UP_prot_name <- rapply(strsplit(ProtName, " OS"), fixed=TRUE), function(a) a[1])
  Gene <- rapply(strsplit(UP_annotations, "GN="), fixed=TRUE), function(a) a[2])
  UP_gene <- rapply(strsplit(Gene, " "), fixed=TRUE), function(a) a[1])
```
UP_prot_ID <- rapply(strsplit(UP_annotations, ",\)\", fixed=TRUE), function(a) a[2])
return(data.frame(UP_prot_ID, UP_prot_name, UP_gene, UP_sequence))
}

###apply function to extract sequences and annotations###
dataUP <- GetAnnotSeqFASTA(UP)
str(dataUP)
```
This uniprot Bos taurus reference proteome contains `r nrow(dataUP)` proteins.

## Filter database
These are the amino acid counts across the proteome:
```{r upfilter}
#see https://stackoverflow.com/questions/19476210/counting-the-number-of-each-letter-in-a-vector-of-strings
table(unlist(strsplit(as.character(dataUP$UP_sequence), \"\"), use.names=FALSE))
```
To allow the package functions to work properly when calculating sequence dependent values - isoelectric point (pI), molecular weight (MW) and Grand Average of Hydropathy (GRAVY) - U, B and X residues need to be removed.
```{r}
###remove U, B and X containing sequences###
#see https://stackoverflow.com/questions/6650510/remove-rows-from-data-frame-where-a-row-match-a-string
dataUP <- dataUP[!(str_count(as.character(dataUP$UP_sequence), \"U\")>=1),]
dataUP <- dataUP[!(str_count(as.character(dataUP$UP_sequence), \"B\")>=1),]
dataUP <- dataUP[!(str_count(as.character(dataUP$UP_sequence), \"X\")>=1),]
```
Following removal of U, B and X containing sequences the Uniprot database contains `r nrow(dataUP)` proteins.

## Calculate protein characteristics
Now the pI, MW and GRAVY can be calculated.
```{r}
###string split uniprot sequences and calculate GRAVY, pI and MW###
split_SEQ <- strsplit(as.character(dataUP$UP_sequence), \"\")
dataUP$pI <- unlist(lapply(split_SEQ, computePI))
dataUP$MW <- unlist(lapply(split_SEQ, pmw))
dataUP$GRAVY <- gravy(dataUP$UP_sequence)
```
# Protein filtering and ID matching
Here, entries with greater than, or equal to, 1 unique peptide are retained. In addition, the first protein ID in the "Protein.IDs" column is extracted and used from here on.
```{r extract}
###Proteins with >= 1 unique peptide retained###
protgroupsMOD <- protgroups[protgroups$Unique.peptides >= 1,]
###Multiple entries for protein IDs removed - first entry retained###
protgroupsMOD$Prot_ID <- rapply(strsplit(as.character(protgroupsMOD$Protein.IDs),\";\"), function(a) a[1])
```
The extracted protein ID in the combined MaxQuant protein groups summary is then used to match to the uniprot dataframe protein ID and extract the GRAVY, pI and MW values for these entries.
```{r}
###EXTRACT by matching Prot_ID in "protgroupsMOD" dataframe to UP_prot_ID from "dataUP"###
#see https://www.r-bloggers.com/manipulating-data-frames-using-sqldf-a-brief-overview/
extract <- sqldf("select * from dataUP inner join protgroupsMOD on protgroupsMOD.Protein.IDs = dataUP.UP_prot_ID")
```
Final count of protein IDs:
```
```r
###count Experiment-specific ID###
table(extract$Experiment)
```

# Data subsets
These data subsets are used for the plots generated by this script.
```
```r [subsets]
###combined experiment subsets###
Q_AF_F <- extract[extract$Exp=="AF" | extract$Exp=="F",]
Q_AE_EF <- extract[extract$Experiment=="AE" | extract$Experiment=="EF",]
Q_AF_AE <- extract[extract$Experiment=="AF" | extract$Experiment=="AE",]
Q_AF_AE_EF_F <- extract[extract$Experiment=="AF" | extract$Experiment=="AE" | extract$Experiment=="EF" | extract$Experiment=="F",]

###single experiment subsets###
Q_A <- extract[extract$Experiment=="A",]
Q_F <- extract[extract$Experiment=="F",]
Q_AF <- extract[extract$Experiment=="AF",]
Q_EF <- extract[extract$Experiment=="EF",]
Q_AE <- extract[extract$Experiment=="AE",]

###venn subsets###
vA <- Q_A[,"UP_prot_ID"]
vF <- Q_F[,"UP_prot_ID"]
vAF <- Q_AF[,"UP_prot_ID"]
vAE <- Q_AE[,"UP_prot_ID"]
vEF <- Q_EF[,"UP_prot_ID"]
```

# General considerations

## Count across samples
```
```r
## Overrepresented proteins across samples
```
```
```r [overlap]
```r
###find proteins across all experiments###
##list these proteins for 4/5 occurrences###
overlap <- rbind(counts[counts$Count==4,],counts[counts$Count==5,])
```
overlap_match <- data.frame()
for(i in overlap$Entry){
    data <- extract[extract$Prot_ID==i,]
    data <- cbind(data, rep(i, nrow(data)))
    overlap_match <- rbind(data, overlap_match)
}
ID_count <- data.frame(table(overlap_match$Prot_ID))
ID_count <- ID_count[ID_count$Freq>=1,]
colnames(ID_count) <- c("Protein_ID", "Count")

There were \`r nrow(ID_count)\` proteins that were identified across 4 or more experiment types (N = 5 total).

```
```{r}
IDs <- ID_count[ID_count$Count==5,]
IDmatch <- data.frame()
for(i in IDs$Protein_ID){
    data <- dataUP[dataUP$UP_prot_ID==i,]
    data$Protein_ID <- i
    IDmatch <- rbind(data, IDmatch)
}
COUNT5 <- data.frame(IDmatch$UP_prot_ID, IDmatch$Protein_ID, IDmatch$UP_prot_name)
colnames(COUNT5) <- c("UP_prot_ID", "Protein_ID", "UP_prot_name")
COUNT5
```

## Count (independent AF searches)

```
```{r}
path <- "/input/IAPP_independent_searches/
###function to extract proteinGroup files from independent Andromeda searches###
GETindependent <- function(x){
    #see https://stackoverflow.com/questions/4876813/using-r-to-list-all-files-with-a-specified-extension
    files_IND <- list.files(path = x, pattern = "proteinGroups.txt", recursive=TRUE, include.dirs=TRUE, ignore.case=TRUE)
    complete_group <- data.frame()
    for (i in files_IND){
        #see https://stackoverflow.com/questions/8996134/extract-vectors-from-strsplit-list-without-using-a-loop
        Experiment <- rapply(strsplit(i, "/"), function(a) a[1])
        prot_group <- read.csv(paste(x, "/", i, sep=""), sep="\t", header=TRUE)
        prot_group$Experiment <- Experiment
        complete_group <- rbind(prot_group, complete_group)
    }
    return(complete_group)
}

###modify data frame###
protgroups_IND <- GETindependent(path)
protgroups_IND$Prot_ID <- rapply(strsplit(as.character(protgroups_IND$Protein.IDs), ";"), function(a) a[1])

###check experiments included###
table(protgroups_IND$Experiment)

###subset based on experiment###
AF1 <- protgroups_IND[protgroups_IND$Experiment == "AF_1",]
AF2 <- protgroups_IND[protgroups_IND$Experiment == "AF_2",]
AF3 <- protgroups_IND[protgroups_IND$Experiment == "AF_3",]
AF4 <- protgroups_IND[protgroups_IND$Experiment == "AF_4",]
AF5 <- protgroups_IND[protgroups_IND$Experiment == "AF_5",]
AF6 <- protgroups_IND[protgroups_IND$Experiment == "AF_6",]
F1 <- protgroups_IND[protgroups_IND$Experiment == "F_1_156",]
###AF venn subsets###
vAF1 <- AF1[,"Prot_ID"]
vAF2 <- AF2[,"Prot_ID"]
vAF3 <- AF3[,"Prot_ID"]
vAF4 <- AF4[,"Prot_ID"]
vAF5 <- AF5[,"Prot_ID"]
vAF6 <- AF6[,"Prot_ID"]

###F venn subsets###
vF1 <- F1[,"Prot_ID"]
vF2 <- F2[,"Prot_ID"]
vF3 <- F3[,"Prot_ID"]
vF4 <- F4[,"Prot_ID"]
vF5 <- F5[,"Prot_ID"]
vF6 <- F6[,"Prot_ID"]
vF7 <- F7[,"Prot_ID"]
vF8 <- F8[,"Prot_ID"]
vF9 <- F9[,"Prot_ID"]

#see https://stackoverflow.com/questions/25019794/venn-diagram-with-item-labels
venn <- venn.diagram(list(AF1 = vAF1, AF2 = vAF2, AF3 = vAF3, AF4 = vAF4, AF5 = vAF5),
                      fill = c("red", "blue", "green", "yellow", "purple"),
                      alpha = 0.4, lwd = 1, cex = 1, cat.cex = 1, filename = NULL)
tiff("./output/venn_AF_independent.tiff", height=300, width=300)
grid.newpage()
grid.draw(venn)
dev.off()

venn <- venn.diagram(list(AF2 = vAF2, AF3 = vAF3, AF4 = vAF4, AF5 = vAF5, AF6 = vAF6),
                      fill = c("blue", "green", "yellow", "purple", "brown"),
                      alpha = 0.4, lwd = 1, cex = 1, cat.cex = 1, filename = NULL)
tiff("./output/venn_AF_independent_2.tiff", height=300, width=300)
grid.newpage()
grid.draw(venn)
dev.off()

venn <- venn.diagram(list(F1 = vF1, F2 = vF2, F3 = vF3, F4 = vF4, F5 = vF5),
                      fill = c("blue", "green", "yellow", "purple", "orange"),
                      alpha = 0.4, lwd = 1, cex = 1, cat.cex = 1, filename = NULL)
tiff("./output/venn_F_independent.tiff", height=300, width=300)
grid.newpage()
grid.draw(venn)
dev.off()

venn <- venn.diagram(list(F3 = vF3, F4 = vF4, F5 = vF5, F6 = vF6, F7 = vF7),
                      fill = c("yellow", "purple", "orange", "brown", "pink"),
                      alpha = 0.4, lwd = 1, cex = 1, cat.cex = 1, filename = NULL)
tiff("./output/venn_F_independent_2.tiff", height=300, width=300)
grid.newpage()
grid.draw(venn)
dev.off()

venn <- venn.diagram(list(F5 = vF5, F6 = vF6, F7 = vF7, F8 = vF8, F9 = vF9),
                      fill = c("orange", "brown", "pink", "gray", "darkblue"),
                      alpha = 0.4, lwd = 1, cex = 1, cat.cex = 1, filename = NULL)
tiff("./output/venn_F_independent_3.tiff", height=300, width=300)
grid.newpage()
grid.draw(venn)
dev.off()
alpha = 0.4, lwd = 1, cex = 1, cat.cex = 1, filename = NULL)
tiff("./output/venn_F_independent_3.tiff", height=300, width=300)
grid.newpage()
grid.draw(venn)
dev.off()

<center>
![](./output/venn_AF_independent.tiff)
![](./output/venn_AF_independent_2.tiff)
![](./output/venn_F_independent.tiff)
![](./output/venn_F_independent_2.tiff)
![](./output/venn_F_independent_3.tiff)
</center>

## Global faceted analysis of variables

### Unique peptides vs peptides
```{r}
#see http://ggplot2.tidyverse.org/reference/facet_grid.html
b <- ggplot(extract)
b <- b + geom_point(aes(Unique.peptides, Peptides), alpha=0.7, pch=21, fill="gray")
b <- b + facet_grid(. ~ Experiment)
b <- b + theme_bw() + theme(axis.text.x = element_text(angle=90, vjust=0.5))
plot(b)
```

### GRAVY vs pI
```{r}
c <- ggplot(extract)
c <- c + geom_point(aes(GRAVY, pI), alpha=0.7, pch=21, fill="gray")
c <- c + facet_grid( ~ Experiment)
c <- c + theme_bw()
plot(c)
```

### GRAVY vs MW
```{r}
d <- ggplot(extract)
d <- d + geom_point(aes(GRAVY, MW), alpha=0.7, pch=21, fill="gray")
d <- d + facet_grid( ~ Experiment)
d <- d + theme_bw() + scale_y_continuous(limits=c(0,1E+6))
plot(d)
```

# Question 1 - CC coronae

## Overlap of AF/F protein IDs

Comparing FBS only (F), with IAPP amyloids exposed to FBS (AF). These samples were prepared with 1 MDa molecular weight cut-off (MWCO) spin columns (CC method).

The identification overlap was as follows:
```{r}
#see https://stackoverflow.com/questions/25019794/venn-diagram-with-item-labels
venn <- venn.diagram(list(F = vF, AF = vAF), fill=c("red", "blue"),
                      alpha = 0.4, lwd = 1, cex = 1.5, cat.cex = 1.5, filename = NULL)
tiff("./output/venn_F_AF.tiff", height=300, width=300)
grid.newpage()
grid.draw(venn)
dev.off()
```

<center>
## Unique AF proteins

Here, plots are created that summarise the physico-chemical properties of proteins unique to the AF samples, highlighting some potentially interesting amyloid binding proteins that could be investigated in future.

Extracting unique entries from combined AF/F data:
```
```{r}
# see https://stackoverflow.com/questions/16905425/find-duplicate-values-in-r
AF_F_ID_count <- data.frame(table(Q_AF_F$Prot_ID))
AF_F_ID_count <- AF_F_ID_count[AF_F_ID_count$Freq==1,]
###extract unique entries for AF and F###
AF_F_unique <- data.frame()
for(i in AF_F_ID_count$Var1){
data <- Q_AF_F[Q_AF_F$Prot_ID==i,]
data <- cbind(data, i)
AF_F_unique <- rbind(data, AF_F_unique)
}
###check dataframe###
head(AF_F_unique)
```

Confirming Venn diagram counts:
```
```{r}
AF_F_unique_count <- data.frame(table(AF_F_unique$Experiment))
colnames(AF_F_unique_count) <- c("Prot_ID", "Count")
AF_F_unique_count
```

Extract unique AF proteins and export these for STRING network analysis:
```
```{r}
###create subset dataframe###
AF <- AF_F_unique[AF_F_unique$Experiment=="AF",]
F <- AF_F_unique[AF_F_unique$Experiment=="F",]

# see http://r.789695.n4.nabble.com/write-text-file-as-output-without-quotes-td888020.html
write.table(AF$Prot_ID, "/output/exp_AF_protlist.txt", sep="\t", quote=FALSE, row.names = FALSE, col.names = TRUE)
```

The top 12 unique AF proteins, based on unique peptide count were:
```
```{r}
# see https://stackoverflow.com/questions/1296646/how-to-sort-a-dataframe-by-columns
uniqueAF_sort <- AF[ order(AF$Unique.peptides), ]
uniqueAF_sort <- data.frame(uniqueAF_sort$Prot_ID, uniqueAF_sort$UP_prot_name, uniqueAF_sort$Unique.peptides)
colnames(uniqueAF_sort) <- c("Protein ID", "Protein name", "Unique peptides")
uniqueAF_sort[1:12,]
```

## Unique AF/F amino acid composition

Concept inspired by a table in the following reference -> DOI:10.1039/C7EN00466D (see http://pubs.rsc.org/en/content/articlehtml/2017/en/c7en00466d)
```
```{r}
###experimental###
propAF <- data.frame()
for (i in AF_F_unique$UP_sequence){
  seq_vec <- unlist(strsplit(as.character(i), ""))
  info <- AAstat(seq_vec, plot=FALSE)
  all <- AF_F_unique[AF_F_unique$UP_sequence==i,]
  summary <- cbind(all, info$Prop$Aliphatic, info$Prop$Aromatic, info$Prop$Non.polar, info$Prop$Polar, info$Prop$Charged, info$Prop$Basic, info$Prop$Acidic)
```
propAF <- rbind(summary, propAF)

###uniprot###
propUP <- data.frame()
for (i in dataUP$UP_sequence){
  seq_vec <- unlist(strsplit(as.character(i), ""))
  info <- AAstat(seq_vec, plot=FALSE)
  all <- dataUP[dataUP$UP_sequence==i,]
  summary <- cbind(all, info$Prop$Aliphatic, info$Prop$Aromatic, info$Prop$Non.polar,
                   info$Prop$Polar, info$Prop$Charged, info$Prop$Basic, info$Prop$Acidic)
  propUP <- rbind(summary, propUP)
}

###subset based on specific columns###
propAF <- propAF[,c(1,2,3,40,41,43,44,45,46,47,48,49)]
###check colnames###
colnames(propAF)
colnames(propUP)

###update column names###
colnames(propAF) <- c("UP_prot_ID", "UP_prot_name", "UP_gene", "Experiment", "Prot_ID",

###plots###
a <- ggplot(propUP) + geom_point(aes(NonPolar, Acidic), col="gray", alpha=0.4)
a <- a + geom_point(data=propAF[propAF$Experiment=="F",], aes(NonPolar, Acidic), fill = "blue", pch=21)
a <- a + geom_point(data=propAF[propAF$Experiment=="AF",], aes(NonPolar, Acidic), fill = "yellow", pch=21)
a <- a + theme_bw() + coord_fixed()
pplot(a)
...

## Unique AF proteins (GRAVY/pI + proteome)
Here, the Bos taurus proteome (gray) is overlaid with all unique FBS only proteins (F, blue), as well as those proteins unique to amyloid + FBS (AF, yellow). Labels are gene names, outside the box indicated on the plot.
The top panel labels unique AF proteins outside the boundaries indicated by the highlighted blue box. The bottom panel labels unique AF proteins (GRAVY <= -1 and GRAVY >= 0 and pI <= 4.5 and pI >= 7).
```{r}
###plot###
tiff("./output/AF_unique_GRAVY_pl_BosT.tiff", width=2500, height=5000, res=300)
#themes, see http://ggplot2.tidyverse.org/reference/theme.html
#geom_rect, see https://stackoverflow.com/questions/4733182/how-to-highlight-time-ranges-on-a-plot
a <- ggplot(dataUP)
a <- a + geom_point(data=propAF, aes(GRAVY, pl), col="gray", alpha=0.1)
a <- a + geom_point(data=F, aes(GRAVY, pl), pch=21, size=5, fill="blue")
a <- a + geom_point(data=AF, aes(GRAVY, pl), pch=21, size=3, fill="yellow")
a <- a + theme_bw() + themeTIFF + xlab("GRAVY") + ylab("pI")
b <- a + geom_rect(data=data.frame(xmin=-1.5, xmax=0.5, ymin=4, ymax=7),
                   aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
                   fill="lightblue", col="black", alpha=0.4,
                   inherit.aes = FALSE)
b <- b + geom_label_repel(data=AF[AF$GRAVY<=-1.5 | AF$GRAVY>=0.5 | AF$pl<=-4 | AF$pl>=10,],
                          aes(GRAVY, pl, label=UP_gene), size=4)
c <- a + scale_y_continuous(limits=c(4,10)) + scale_x_continuous(limits=c(-1.5,0.5))
c <- c + geom_rect(data=data.frame(xmin=-1, xmax=0, ymin=4.5, ymax=7),
                   aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
                   fill="lightblue", col="black", alpha=0.4,
                   inherit.aes = FALSE)
c <- c + geom_label_repel(data=AF[AF$GRAVY<=-1 | AF$GRAVY>=0 | AF$pl<=-4.5 | AF$pl>=7,],
                          aes(GRAVY, pl, label=UP_gene), size=4)
The following figure includes all AF and unique AF proteins, with the scatter markers size based on number of unique peptides. Unique AF proteins with >= 10 unique peptides (yellow) and all AF proteins with >= 20 unique peptides (white) are labelled.

```{r}
###plot###
tiff("./output/AF_unique_GRAVY_pl_all_AF_unique_pep.tiff", width=700, height=700)
d <- ggplot(dataUP) + geom_point(data=dataUP, aes(GRAVY, MW), col="gray", alpha=0.1)
d <- d + geom_rect(data=data.frame(xmin=-1.5, xmax=0.5, ymin=0, ymax=6E+5),
    aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
    fill="lightblue", col="black", alpha=0.4, inherit.aes = FALSE)
d <- b + geom_label_repel(data=AF[M$GRAVY<=-1.5 | AF$GRAVY>=0.5 | AF$MW>=6E+5,],
    aes(GRAVY, MW, label=UP_gene, size=4))
d <- c + guides(fill=FALSE)
grid.arrange(b, c, nrow=2)
dev.off()
```
## Overlap of AE/EF protein IDs

The identification overlap was as follows:

```r
#see https://stackoverflow.com/questions/25019794/venn-diagram-with-item-labels
venn <- venn.diagram(list(AE = vAE, EF = vEF), fill = c("yellow", "blue"),
  alpha = 0.4, lwd = 1, cex = 1.5, cat.cex = 1.5, filename = NULL)

tif("./output/venn_AE_EF.tiff", width=2500, height=2500, res=300)
grid.newpage()
grid.draw(venn)
dev.off()
```

![](./output/venn_AE_EF.tiff)

## Overlay of unique AE/EF proteins

```r
#see https://stackoverflow.com/questions/16905425/find-duplicate-values-in-r
AE_EF_ID_count <- data.frame(table(Q_AE_EF$Prot_ID))
AE_EF_ID_count <- AE_EF_ID_count[AE_EF_ID_count$Freq==1,]

AE_EF_unique <- data.frame()
for(i in AE_EF_ID_count$Var1){
  data <- Q_AE_EF[Q_AE_EF$Prot_ID==i,]
  data <- cbind(data, i)
  AE_EF_unique <- rbind(data, AE_EF_unique)
}

AE_EF <- data.frame(table(AE_EF_unique$Experiment))
AE_EF <- AE_EF[AE_EF$Freq>0,]
colnames(AE_EF) <- c("Prot_ID", "Count")
Q2_AE <- AE_EF_unique[AE_EF_unique$Experiment=="AE",]
Q2_EF <- AE_EF_unique[AE_EF_unique$Experiment=="EF",]
```

There were a total of `r nrow(Q_EF)` and `r nrow(Q_AE)` IDs for experiments EF and AE, respectively. Of these there were `r nrow(Q2_EF)` and `r nrow(Q2_AE)` unique protein identifications for EF and AE, respectively.

The top 12 unique AE proteins, based on unique peptide count were:

```r
uniqueAE_sort <- Q2_AE[ order(-Q2_AE$Unique.peptides), ]
uniqueAE_sort <- data.frame(uniqueAE_sort$Prot_ID, uniqueAE_sort$UP_prot_name,
  uniqueAE_sort$Unique.peptides)
colnames(uniqueAE_sort) <- c("Protein ID", "Protein name", "Unique peptides")
uniqueAE_sort[1:12,]
```

Here, the protein list for unique AE proteins was exported for STRING DB analysis.

```r export_AE
#see http://r.789695.n4.nabble.com/write-text-file-as-output-without-quotes-td888020.html
write.table(Q2_AE$Prot_ID, "./output/exp_AE_protlist.txt", sep="\t", quote=FALSE, row.names = FALSE,
col.names = FALSE)
```

The unique EF proteins were:

```r
uniqueEF_sort <- Q2_EF[ order(-Q2_EF$Unique.peptides), ]
uniqueEF_sort <- data.frame(uniqueEF_sort$Prot_ID, uniqueEF_sort$UP_prot_name,
  uniqueEF_sort$Unique.peptides)
colnames(uniqueEF_sort) <- c("Protein ID", "Protein name", "Unique peptides")
uniqueEF_sort
```
## Unique AE/EF amino acid composition

Concept inspired by a table in the following reference - DOI:10.1039/C7EN00466D (see http://pubs.rsc.org/en/content/articlehtml/2017/en/c7en00466d)

```{r}
propAE <- data.frame()
for (i in AE_EF_unique$UP_sequence){
  seq_vec <- unlist(strsplit(as.character(i), ""))
  info <- AAstat(seq_vec, plot=FALSE)
  all <-(AE_EF_unique[AE_EF_unique$UP_sequence==i,])
  summary <- cbind(all, info$Prop$Aliphatic, info$Prop$Aromatic, info$Prop$Non.polar, info$Prop$Polar, info$Prop$Charged, info$Prop$Basic, info$Prop$Acidic)
  propAE <- rbind(summary, propAE)
}
###subset and rename columns###
propAE <- propAE[,c(1,2,3,40,41,43,44,45,46,47,48,49)]
###check column names###
colnames(propAE)
###update column names###
###plots###
a <- ggplot(propUP) + geom_point(aes(NonPolar, Acidic), col="gray", alpha=0.4)
a <- a + geom_point(data=propAE[propAE$Experiment=="AE",], aes(NonPolar, Acidic), fill = "yellow", pch=21)
a <- a + geom_point(data=propAE[propAE$Experiment=="EF",], aes(NonPolar, Acidic), fill = "blue", pch=21)
a <- a + theme_bw() + coord_fixed()
plot(a)
```

## Unique AE/EF proteins (GRAVY/pI/MW + proteome)

Here, the Bos taurus proteome (gray) is overlaid with unique AE (yellow) and unique EF (blue) proteins - plotting GRAVY/pl and GRAVY/MW.

```{r}
```
```{r}
#see https://stackoverflow.com/questions/25019794/venn-diagram-with-item-labels
venn <- venn.diagram(list(AF = vAF, AE = vAE), fill = c("red", "blue"),
                    alpha = 0.4, lwd = 1, cex = 2.5, cat.cex = 2.5, filename = NULL)
tiff("./output/venn_AF_AE_F_EF.tiff", height=300, width=300)
grid.newpage()
grid.draw(venn)
dev.off()

# Question 3 - CC vs QCM

## Overlap of AF/AE protein IDs
The next comparison of interest was to consider the difference, if any, between the amyloids + FBS prepared by
either the CC method (AF) or the QCM method (AE).

The identification overlap for these experiments was as follows:
```
# Question 4 - Control nLC-MS/MS

## A

Amyloid only control for CC method.

Sample A nLC-MS/MS analyses identified a total of \`r nrow(Q_A)` proteins.

```{r}
Q_A_sort <- Q_A[ order(-Q_A$Unique.peptides), ]
A_list <- data.frame(Q_A_sort$Prot_ID, Q_A_sort$UP_prot_name, Q_A_sort$Unique.peptides)
colnames(A_list) <- c("Protein ID", "Protein name", "Unique peptides")
A_list[1:12,]
```

The identification overlap for A/F/AF was:

```{r}
# see https://stackoverflow.com/questions/25019794/venn-diagram-with-item-labels
venn <- venn.diagram(list(A = vA, F = vF, AF=vAF), fill = c("green", "blue", "yellow"),
alpha = 0.4, lwd = 1, cex = 2.5, cat.cex = 2.5, filename = NULL)
tiff("./output/venn_A_F_AF.tiff", width=2500, height=2500, res=300)
grid.newpage()
grid.draw(venn)
dev.off()
```

```{r}
![](./output/venn_A_F_AF.tiff)
```

## A (using human .fasta file)

```{r}
A_human <- read.csv("./input/Ahuman/combined/txt/proteinGroups.txt", sep="t", header=TRUE)
A_human$Prot_ID <- rapply(strsplit(as.character(A_human$Protein.IDs),";"), function(a) a[1])
A_human$Prot_name <- rapply(strsplit(as.character(A_human$Protein.names),";"), function(a) a[1])
```

Sample A nLC-MS/MS analyses using a human fasta file for the Andromeda search identified a total of \`r nrow(Ahuman)` proteins.

The top 12 proteins in this list, based on unique peptides counts, were:

```{r}
A_human_sort <- A_human[ order(-A_human$Unique.peptides), ]
A_human_sort <- data.frame(A_human_sort$Prot_ID, A_human_sort$Prot_name,
A_human_sort$Unique.peptides)
colnames(A_human_sort) <- c("Protein ID", "Protein name", "Unique peptides")
A_human_sort[1:12,]
```

# Extract data consistency check

Random subset of 30 rows from extract dataframe checked to make sure that IDs, names and sequences align correctly.

```{r}
# see https://stackoverflow.com/questions/8273313/sample-random-rows-in-dataframe
CHECK <- extract[sample(nrow(extract), 30), ]
write.table(CHECK, "./output/extract_consistency_check.txt", sep="t")
```
# Check of MaxQuant parameters
This section checks the consistency of the parameters used for MaxQuant.

```r
path <- "/input/IAPP"
GETparameters <- function(x)
    # see https://stackoverflow.com/questions/4876813/using-r-to-list-all-files-with-a-specified-extension
    parameter_files <- list.files(path = x, pattern = "parameters.txt", recursive=TRUE, ignore.case = TRUE)
    complete_parameters <- data.frame()
    for (i in parameter_files)
        # see https://stackoverflow.com/questions/8996134/extract-vectors-from-strsplit-list-without-using-a-loop
        Experiment <- rapply(strsplit(i, "/"), function(a) a[1])
        parameters <- read.csv(paste(x, "/", i, sep=""), sep="\t", header=TRUE)
        colnames(parameters) <- c("Parameter", Experiment)
        # see http://www.r-tutor.com/r-introduction/data-frame/data-frame-column-slice
        complete_parameters <- cbind(complete_parameters, parameters[2])
    return(complete_parameters)

parameters <- GETparameters(path)

# see https://stackoverflow.com/questions/28628384/count-number-of-unique-values-per-row
apply(parameters[,2:6], 1, function(x)length(unique(x)))

Row four contains multiple unique entries. This row contains reports the following variable:

```r
parameters[4,1]
```

***

# Chromatogram plotting script

Package xcms license is GPL (>= 2) + a file LICENSE available @ https://bioconductor.org/packages/release/bioc/licenses/xcms/LICENSE.

```r
#source("https://bioconductor.org/biocLite.R")
biocLite("xcms")
```

++++++++++++++++++++++
### libraries required###
++++++++++++++++++++++
#library(xcms)
#library(ggplot2)
#library(gridExtra)
#library(stringr)
#library(plyr)

++++++++++++++++++++++
### citations###
++++++++++++++++++++++
#citation("xcms")
#citation("ggplot2")
#citation("gridExtra")
#citation("stringr")

++++++++++++++++++++++
### notes###
++++++++++++++++++++++
### Process selected directory###

```r
#DATA <- lcms_vis(path)
write.table(DATA, ".\complete_TIC_data.txt", sep="\t")
```

### Create combined chromatograms###

```r
for (i in unique(DATA$experiment)){
  plotDATA <- subset(DATA, DATA$experiment==i)
  pdf(paste("./chromatogram_overlays/chromatograms_", i,".pdf",sep=""), width=15, height=15)
  # see https://www.statmethods.net/advgraphs/layout.html
  par(mfrow=c(1,2))
  plotChrom(data, base=TRUE)
  plotChrom(data, base=FALSE)
  dev.off()
  TIC <- data.frame(uniqueID, ID, experiment, replicate, data@scantime, data@tic)
  #complete_TIC_data <- rbind(complete_TIC_data, TIC)
  }
  return(complete_TIC_data)
  }
```
```r
# a = a + theme_bw() + theme(legend.text = element_text(size = 20), legend.title=element_text(size=25),
#     axis.title=element_text(size=25), legend.key.size = unit(0.8, "cm"),
#     axis.text.x = element_text(size=20), axis.text.y = element_text(size=20),
#     plot.title = element_text(size=25))
# plot(a)
# dev.off()
#
### session info ###
```

## General attributions

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rmarkdown basics</td>
<td><a href="http://rmarkdown.rstudio.com/authoring_basics.html">http://rmarkdown.rstudio.com/authoring_basics.html</a></td>
</tr>
<tr>
<td>Rmarkdown format</td>
<td><a href="http://rmarkdown.rstudio.com/markdown_document_format.html">http://rmarkdown.rstudio.com/markdown_document_format.html</a></td>
</tr>
<tr>
<td>css style</td>
<td><a href="https://stackoverflow.com/questions/30446905/rmarkdown-type-change">https://stackoverflow.com/questions/30446905/rmarkdown-type-change</a></td>
</tr>
<tr>
<td>css body/TOC modifications</td>
<td><a href="https://rpubs.com/stevepowell99/floating-css">https://rpubs.com/stevepowell99/floating-css</a></td>
</tr>
<tr>
<td>reshape</td>
<td><a href="https://www.statmethods.net/management/reshape.html">https://www.statmethods.net/management/reshape.html</a></td>
</tr>
<tr>
<td>ggrepel</td>
<td><a href="https://cran.r-project.org/web/packages/ggrepel/vignettes/ggrepel.html">https://cran.r-project.org/web/packages/ggrepel/vignettes/ggrepel.html</a></td>
</tr>
<tr>
<td>ggplot theme</td>
<td><a href="https://rstudio-pubs-static.s3.amazonaws.com/3364_d1a578f521174152b46b19d0c83cbe7e.html">https://rstudio-pubs-static.s3.amazonaws.com/3364_d1a578f521174152b46b19d0c83cbe7e.html</a></td>
</tr>
<tr>
<td>Stand alone error</td>
<td><a href="https://github.com/rstudio/rmarkdown/issues/228">https://github.com/rstudio/rmarkdown/issues/228</a></td>
</tr>
<tr>
<td>Subsetting</td>
<td><a href="http://adv-r.had.co.nz/Subsetting.html">http://adv-r.had.co.nz/Subsetting.html</a></td>
</tr>
</tbody>
</table>

Additional attributions appear throughout this document. Citations, session information and package links appear below.
## Session info
```
{r sessioninfo}
sessionInfo()
```

## Package links
<table>
<thead>
<tr>
<th>Name</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>gridExtra</td>
<td><a href="https://CRAN.R-project.org/package=gridExtra">https://CRAN.R-project.org/package=gridExtra</a></td>
</tr>
<tr>
<td>ggplot2</td>
<td><a href="http://ggplot2.org">http://ggplot2.org</a></td>
</tr>
<tr>
<td>seqinr</td>
<td><a href="https://cran.r-project.org/web/packages/seqinr/">https://cran.r-project.org/web/packages/seqinr/</a></td>
</tr>
<tr>
<td>alakazam</td>
<td><a href="http://doi.org/10.1126/scitranslmed.3008879">http://doi.org/10.1126/scitranslmed.3008879</a></td>
</tr>
<tr>
<td>sqldf</td>
<td><a href="https://CRAN.R-project.org/package=sqldf">https://CRAN.R-project.org/package=sqldf</a></td>
</tr>
<tr>
<td>ggrepel</td>
<td><a href="https://CRAN.R-project.org/package=ggrepel">https://CRAN.R-project.org/package=ggrepel</a></td>
</tr>
<tr>
<td>VennDiagram</td>
<td><a href="https://cran.r-project.org/web/packages/VennDiagram/index.html">https://cran.r-project.org/web/packages/VennDiagram/index.html</a></td>
</tr>
<tr>
<td>stringr</td>
<td><a href="https://cran.r-project.org/web/packages/stringr/index.html">https://cran.r-project.org/web/packages/stringr/index.html</a></td>
</tr>
<tr>
<td>reshape2</td>
<td><a href="https://cran.r-project.org/web/packages/reshape2/">https://cran.r-project.org/web/packages/reshape2/</a></td>
</tr>
</tbody>
</table>

## Package citations
```
{r citations}
citation("gridExtra")
citation("ggplot2")
citation("seqinr")
citation("alakazam")
citation("sqldf")
citation("ggrepel")
citation("VennDiagram")
citation("stringr")
citation("reshape2")
#citation("xcms")
```

References


