protViz: Visualizing and Analyzing Mass Spectrometry Related Data in Proteomics

Christian Panse        Jonas Grossmann

September 30, 2014

Vignette for package version ≥ v.0.1.48

Recent changes, updates, and new features:

- 0.2.9 added sigmamix data
- 0.2.5 export aa2mass
- 0.2.0 peakplot margin bug fix

Contents

1 Preliminary Note 3
2 Related Work 3
3 Get Data In – Preprocessing 3
  3.1 Identification - In-silico from Proteins to Peptides . . . . . . . . . . . . . . . 4
4 Peptide Identification 5
  4.1 Computing the Parent Ion Mass ............................................ 5
  4.2 In-silico Peptide Fragmentation ........................................... 6
  4.3 Peptide Sequence – Fragment Ion Matching .................................. 8
  4.4 Modifications . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
  4.5 Labeling Peaklists .............................................................. 11
5 Quantification 15
  5.1 Label-free methods on protein level .................................... 15
  5.2 pgLFQ – LCMS based label-free quantification .......................... 17
  5.3 iTRAQ – Two Group Analysis .............................................. 21
    5.3.1 Sanity Check .............................................................. 21
    5.3.2 On Protein Level ......................................................... 22
    5.3.3 On Peptide Level ....................................................... 23
1 Preliminary Note

protViz is an R package to do quality checks, visualizations and analysis of mass spectrometry data, coming from proteomics experiments. The package is developed, tested and used at the Functional Genomics Center Zurich. We use this package mainly for prototyping, teaching, and having fun with proteomics data. But it can also be used to do data analysis for small scale data sets. Nevertheless, if one is patient, it also handles large data sets.

2 Related Work

The method of choice in proteomics is mass spectrometry. There are already packages in R which deal with mass spec related data. Some of them are listed here:

- MSnbase package (basic functions for mass spec data including quant aspect with iTRAQ data)
- plgcm – spectral counting quantification, applicable to MudPIT experiments
- synapter – MSe (Hi3 = Top3 Quantification) for Waters Q-tof data acquired in MSe mode
- mzR
- isobar iTRAQ/TMT quantification package
- readMzXmlData
  [http://cran.r-project.org/web/packages/readMzXmlData/](http://cran.r-project.org/web/packages/readMzXmlData/)
- msQC

3 Get Data In – Preprocessing

The most time consuming and challenging part for data analysis and visualization is shaping the data that they can easily be processed. In this package, we intentionally left this part away because it is very infrastructure dependent. Moreover we use also commercial tools to analyze data and export the data into R accessible formats. We provide different kind of importers if these formats are available, but with very little effort, one can bring other exports in a similar format which will make it easy to use our package for a variety of tools.
3.1 Identification - In-silico from Proteins to Peptides

For demonstration we use a sequence of peptides derived from a tryptics digest using the Swiss-
sprot FETUA_BOVIN Alpha-2-HS-glycoprotein precursor (Fetuin-A) (Asialofetuin) protein.

fcat and tryptic-digest are commandline programs which are included in the package.
fcat removes the lines starting with > and all ' new line' character within the protein sequence
while tryptic-digest is doing the trptic digest of a protein sequence applying the rule:
cleave after arginine (R) and lysine (K) except followed by proline(P).

$ cat Fetuin.fasta
MKSFVLLFCLAIQLWGCHSIPLDPVAGYKEPACDDPDTEQAALAAVDYINKHLPRGYKHTL
NQIDSVKVWPRPRPTGEVYDIEIDTLTEGRVNLPLCSVRQQTHAVEGDCDIHVLLK
QQQGF我们的字典目录CTSDSPDASEDRVKLCPCPLNLAPLNSRVVHAVEVALATFNAESNGSYL
QLVEISRAQFVLPVSLSVEFABAATDCAKEVVDPTKCNLLAEKQYGFCKGSVIQKALG
GEDVRVTCTFQTVQPQPDGAEAEAPSVDAAGPTPSAAAGPPVSVEVGPSVSVA
PLPLHRAYDLHRTFSGVASVEESSGEAFHVGTKPIVQQPSIPIGGPGVRLCPGRIRYFKI

$ cat Fetuin.fasta | fcat | tryptic-digest
MK
SFVLLFCLAIQLWGCHSIPLDPVAGY
EPACDDPDTEQAALAAVDYINK
HLPR
GYK
HTLNQIDSVK
VWPR
RPTGEVYDIEIDTLTEGRVNLPLCSVR
QQQGF我们的字典目录CTSDSPDASEDRVKLCPCPLNLAPLNSRVVHAVEVALATFNAESNGSYL
QLVEISRAQFVLPVSLSVEFABAATDCAKEVVDPTKCNLLAEKQYGFCKGSVIQKALG
GEDVRVTCTFQTVQPQPDGAEAEAPSVDAAGPTPSAAAGPPVSVEVGPSVSVA
PLPLHRAYDLHRTFSGVASVEESSGEAFHVGTKPIVQQPSIPIGGPGVRLCPGRIRYFKI

I
4 Peptide Identification

The currency in proteomics are the peptides. In proteomics, proteins are digested to so-called peptides since peptides are much easier to handle biochemically than proteins. Proteins are very different in nature some are very sticky while others are soluble in aqueous solutions while again are only sitting in membranes. Therefore, proteins are chopped up into peptides because it is fair to assume, that for each protein, there will be a number of peptides behaving well, so that they can actually be measured with the mass spectrometer. This step introduces another problem, the so-called protein inference problem. In this package here, we do not at all touch upon the protein inference.

4.1 Computing the Parent Ion Mass

```r
> library(protViz)
> op<-par(mfrow=c(1,1))
> fetuin<-c('MK', 'SFVLLFCLAQLWGCHSIPLDPVAGYK',
+ 'EPACDDDPDEQAALAADVYINK',
+ 'HLPR', 'GYK', 'HTLNQDISVK', 'VWPR',
+ 'RPTGEVYDIEIETTCHVDPLTPLANCSR',
+ 'QQTQHAVEGDCDIHVLK', 'QDGQFSVLFTK',
+ 'CDSSPDASPDVR', 'K', 'LCPDCPLAPLNDSR',
+ 'VVHAVEVALATFNAESNGSYLQLVEISR',
+ 'AQFVPLPVSVSVEFAAATDCIAK',
+ 'EVVDPTK', 'CNLLAEK', 'QYGFCK',
+ 'GSVIQK', 'ALGGEDVR',
+ 'VTCTLFQTQPVPQPQDDGAEAEAPSAVDAAGPTPSAGPPVAVPVGPSVVAVPLPLHR',
+ 'AHYDLR', 'HTFGVAVESESSGGEAFHVKK',
+ 'TPIVQPSIPGGPVR', 'LCPGR', 'IR', 'YFK', 'I')
> (pm<-parentIonMass(fetuin))
[22] 774.3893 2120.0043 1474.8376 602.3079 288.2030 457.2445 132.1019
```

```r
> op<-par(mfrow=c(2,1))
> plot(pm, ylab="peptide mass [in Da]",
+ main="Fetuin Peptide tryptic digested.")
> hist(pm, xlab="peptide mass [in Da"]
```


4.2 In-silico Peptide Fragmentation

The fragment ions of a peptide can be computed following the rules proposed in [4]. Beside the b and y ions the FUN argument of fragmentIon defines which ions are computed. The default ions being computed are defined in the function defaultIon. The are no limits for defining other forms of fragment ions for ETD (c and z ions) CID (b and y ions).

```r
> defaultIon

function (b, y)
{
    Hydrogen <- 1.007825
    Oxygen <- 15.994915
    Nitrogen <- 14.003074
    c <- b + (Nitrogen + (3 * Hydrogen))
    z <- y - (Nitrogen + (3 * Hydrogen))
    return(cbind(b, y, c, z))
}
```

> peptides<-c('HTLNQIDSVK', 'ALGGEDVR', 'TPIVGQPSIPGPVR')
> pim<-parentIonMass(peptides)
> fi<-fragmentIon(peptides)
> par(mfrow=c(3,1));
> for (i in 1:length(peptides)){
+    plot(0,0,
+        xlab='m/Z',
+        ylab=''
+        xlim=range(c(fi[[i]][[1]]$b,fi[[i]][[1]]$y)),
+        ylim=c(0,1),
+        type='n',
+        axes=FALSE,
+        sub=paste( pim[i], "Da"));
+    box()
+    axis(1,fi[[i]][[1]]$b,round(fi[[i]][[1]]$b,2))
+    pepSeq<-strsplit(peptides[i],"")
+    axis(3,fi[[i]][[1]]$b,pepSeq[[1]])
+    abline(v=fi[[i]][[1]]$b, col='red', lwd=2)
+    abline(v=fi[[i]][[1]]$c, col='orange')
+    abline(v=fi[[i]][[1]]$y, col='blue', lwd=2)
+    abline(v=fi[[i]][[1]]$z, col='cyan')
+    }

mZ 1154.616401 Da

mZ 816.420981 Da

mZ 1474.837601 Da
The next lines compute the singly and doubly charged fragment ions of the HTLNQIDSVK peptide. Which are usually the ones that can be used to make an identification.

```r
> Hydrogen<-1.007825
> (fi.HTLNQIDSVK.1<-.fragmentIon('HTLNQIDSVK'))[[1]]

<table>
<thead>
<tr>
<th></th>
<th>b</th>
<th>y</th>
<th>c</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>138.0662</td>
<td>147.1128</td>
<td>155.0927</td>
<td>130.0863</td>
</tr>
<tr>
<td>2</td>
<td>239.1139</td>
<td>246.1812</td>
<td>256.1404</td>
<td>229.1547</td>
</tr>
<tr>
<td>3</td>
<td>352.1979</td>
<td>333.2132</td>
<td>369.2245</td>
<td>316.1867</td>
</tr>
<tr>
<td>4</td>
<td>466.2409</td>
<td>448.2402</td>
<td>483.2674</td>
<td>431.2136</td>
</tr>
<tr>
<td>5</td>
<td>594.2994</td>
<td>561.3242</td>
<td>611.3260</td>
<td>544.2977</td>
</tr>
<tr>
<td>6</td>
<td>707.3835</td>
<td>689.3828</td>
<td>724.4100</td>
<td>672.3563</td>
</tr>
<tr>
<td>7</td>
<td>822.4104</td>
<td>803.4258</td>
<td>839.4370</td>
<td>786.3992</td>
</tr>
<tr>
<td>8</td>
<td>909.4425</td>
<td>916.5098</td>
<td>926.4690</td>
<td>899.4833</td>
</tr>
<tr>
<td>9</td>
<td>1008.5109</td>
<td>1017.5575</td>
<td>1025.5374</td>
<td>1000.5309</td>
</tr>
<tr>
<td>10</td>
<td>1136.6058</td>
<td>1154.6164</td>
<td>1153.6324</td>
<td>1137.5899</td>
</tr>
</tbody>
</table>

> (fi.HTLNQIDSVK.2<-(fi.HTLNQIDSVK.1[[1]] + Hydrogen) / 2)

<table>
<thead>
<tr>
<th></th>
<th>b</th>
<th>y</th>
<th>c</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>69.53701</td>
<td>74.06031</td>
<td>78.05028</td>
<td>65.54704</td>
</tr>
<tr>
<td>2</td>
<td>120.06085</td>
<td>123.59452</td>
<td>128.57412</td>
<td>115.08124</td>
</tr>
<tr>
<td>3</td>
<td>176.60288</td>
<td>167.11053</td>
<td>185.11615</td>
<td>158.59726</td>
</tr>
<tr>
<td>4</td>
<td>233.62434</td>
<td>224.62400</td>
<td>242.13761</td>
<td>216.11073</td>
</tr>
<tr>
<td>5</td>
<td>297.65636</td>
<td>281.16603</td>
<td>306.16691</td>
<td>272.65276</td>
</tr>
<tr>
<td>6</td>
<td>354.19666</td>
<td>345.19532</td>
<td>362.70894</td>
<td>336.68205</td>
</tr>
<tr>
<td>7</td>
<td>411.70913</td>
<td>402.21679</td>
<td>420.22241</td>
<td>393.70351</td>
</tr>
<tr>
<td>8</td>
<td>455.22515</td>
<td>458.75882</td>
<td>463.73842</td>
<td>450.24554</td>
</tr>
<tr>
<td>9</td>
<td>504.75935</td>
<td>509.28266</td>
<td>513.27262</td>
<td>500.76938</td>
</tr>
<tr>
<td>10</td>
<td>568.80683</td>
<td>577.81211</td>
<td>577.32010</td>
<td>569.29884</td>
</tr>
</tbody>
</table>
```

### 4.3 Peptide Sequence – Fragment Ion Matching

Given a peptide sequence and a tandem mass spectrum. For the assignment of a candidate peptide an in-silico fragment ion spectra $fi$ is computed. The function `findNN` determines for each fragment ion the closested peak in the MS2. If the difference between the in-silico mass and the measured mass is inside the 'accuracy' mass window of the mass spec device the in-silico fragment ion is considered as potential hit.

```r
> peptideSequence<-'HTLNQIDSVK'
> spec<-list(scans=1138,
+   title="178: (rt=22.3807) [20080816_23_fetuin_160.RAW]",
+   rtinseconds=1342.8402,
+   charge=2,
+   mZ=c(195.139940, 221.211970, 239.251780, 290.221750,
+   316.300770, 333.300050, 352.258420, 448.384360, 466.348830,
+   496.207570, 509.565910, 538.458310, 547.253380, 556.173940,
+   568.80683, 577.81211, 577.32010, 569.29884)
```
> fi <- fragmentIon(peptideSequence)
> n <- nchar(peptideSequence)
> by.mZ <- c(fi[[1]]$b, fi[[1]]$y)
> by.label <- c(paste("b", 1:n, sep = ""), paste("y", n:1, sep = ""))
> # should be a R-core function as findInterval!
> idx <- findNN(by.mZ, spec$mZ)
> mZ.error <- abs(spec$mZ[idx] - by.mZ)
> plot(mZ.error[mZ.error.idx <- order(mZ.error)],
+     main = "Error Plot",
+     pch = 'o',
+     cex = 0.5,
+     sub = 'The error cut-off is 0.6Da (grey line).',
+     log = 'y')
> abline(h = 0.6, col = 'grey')
> text(1:length(by.label),
+      mZ.error[mZ.error.idx],
+      by.label[mZ.error.idx],
+      cex = 0.75, pos = 3)
The graphic above is showing the mass error of the assignment between the MS2 spec and the singly charged fragment ions of HTLNQISDVK. The function psm is doing the peptide sequence matching. Of course, the more theoretical ions match (up to a small error tolerance, given by the system) the actually measured ion series, the more likely it is, that the measured spectrum indeed is from the inferred peptide (and therefore the protein is identified).

### 4.4 Modifications

```r
> library(protViz)
> ptm.0 <- cbind(AA="-",
+     mono=0.0, avg=0.0, desc="unmodified", unimodAccID=NA)
> ptm.616 <- cbind(AA='S',
+     mono=-27.010899, avg=NA, desc="Substituition",
+     unimodAccID=616)
> ptm.651 <- cbind(AA='N',
+     mono=27.010899, avg=NA, desc="Substituition",
+     unimodAccID=651)
> m <- as.data.frame(rbind(ptm.0, ptm.616, ptm.651))
> genMod(c('TAFDEIAEELDTLNEESYK','TAFDEIAEELDTLSEESYK'), m$AA)

[[1]]
[1] "0000000000000000000" "0000000000000200000" "0000000000000001000"
[4] "0000000000000100200"
```
4.5 Labeling Peaklists

The labeling of the spectra can be done with the `peakplot` function.

```r
> data(msms)
> op<-par(mfrow=c(2,1))
> peakplot("TAFDEAIAELDTLSEESYK", msms[[1]])
```

$mZ.Da.error

```r
$[1] 232.331344 161.294234 14.225824 -0.032616 -0.143306 0.032244
```

$mZ.ppm.error

```r
$[1] 2.276532e+06 9.318407e+05 4.443342e+04 -7.494702e+01 -2.539851e+02
```

11
[[31] -4.321937e+01 4.781134e-01 -5.770282e+01 1.165934e+03 -4.572174e+03 1.545734e+03 -4.106862e+00 -1.262129e+04 -5.104441e+01 -1.318183e+04 2.768725e+03 1.971690e+03 -1.038832e+04 -9.644849e+03 -9.694247e+03 -1.764117e+04 -7.595171e+04 1.570497e+06 1.406678e+05 4.491332e+04 -1.656190e+03 -1.710589e+03 1.726789e+02 1.973544e+04 -3.850849e+02 -1.529356e+04 -8.747728e+02 -7.562373e+02 -1.010347e+03 -1.986529e+03 1.072648e+04 1.114745e+04 4.725878e+03 1.229618e+04 -2.293808e+04 -6.903096e+04

$idx
[1] 1 1 1 3 14 21 38 49 64 87 91 97 102 106 110 113 115 116 116
[39] 1 1 1 3 16 24 41 52 67 88 93 97 104 107 110 113 115 116 116
[58] 1 1 2 11 22 40 53 68 88 93 98 103 106 108 111 114 116 116 116

$label
[1] "b1" "b2" "b3" "b4" "b5" "b6" "b7" "b8" "b9" "b10" "b11" "b12"
[13] "b13" "b14" "b15" "b16" "b17" "b18" "b19" "y1" "y2" "y3" "y4" "y5"
[25] "y6" "y7" "y8" "y9" "y10" "y11" "y12" "y13" "y14" "y15" "y16" "y17"
[37] "y18" "y19" "c1" "c2" "c3" "c4" "c5" "c6" "c7" "c8" "c9" "c10"
[49] "c11" "c12" "c13" "c14" "c15" "c16" "c17" "c18" "c19" "z1" "z2" "z3"
[61] "z4" "z5" "z6" "z7" "z8" "z9" "z10" "z11" "z12" "z13" "z14" "z15"
[73] "z16" "z17" "z18" "z19"

$score
[1] -1

$sequence
[1] "TAFDEAIAELDLNEESYK"

$fragmentIon

<table>
<thead>
<tr>
<th>b</th>
<th>y</th>
<th>c</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>102.0550</td>
<td>147.1128</td>
<td>119.0815</td>
</tr>
<tr>
<td>2</td>
<td>173.0921</td>
<td>310.1761</td>
<td>190.1186</td>
</tr>
<tr>
<td>3</td>
<td>320.1605</td>
<td>397.2082</td>
<td>337.1870</td>
</tr>
<tr>
<td>4</td>
<td>435.1874</td>
<td>526.2508</td>
<td>452.2140</td>
</tr>
<tr>
<td>5</td>
<td>564.2300</td>
<td>655.2933</td>
<td>581.2566</td>
</tr>
<tr>
<td>6</td>
<td>635.2671</td>
<td>769.3363</td>
<td>652.2937</td>
</tr>
<tr>
<td>7</td>
<td>748.3512</td>
<td>882.4203</td>
<td>765.3777</td>
</tr>
<tr>
<td>8</td>
<td>819.3883</td>
<td>983.4680</td>
<td>836.4148</td>
</tr>
<tr>
<td>9</td>
<td>948.4309</td>
<td>1098.4950</td>
<td>965.4574</td>
</tr>
<tr>
<td>10</td>
<td>1061.5149</td>
<td>1211.5790</td>
<td>1078.5415</td>
</tr>
<tr>
<td>11</td>
<td>1176.5419</td>
<td>1340.6216</td>
<td>1193.5684</td>
</tr>
<tr>
<td>12</td>
<td>1277.5896</td>
<td>1411.6587</td>
<td>1294.6161</td>
</tr>
<tr>
<td>13</td>
<td>1390.6736</td>
<td>1524.7428</td>
<td>1407.7002</td>
</tr>
</tbody>
</table>
$mZ.Da.error
[1]  245.264254  174.227144  27.158734  14.444434  0.021404 -0.111266
 [7] -0.039926 -0.021626 -0.121916 -8.079236 -0.158376 -0.153156
[13] -0.094316 -0.022946 -0.186736 -0.092226 -0.120456 -0.151686
[19] -128.246646  200.206409  37.143079  0.078909  0.062269  0.129769
[25]  0.103729  0.060869 -0.051451 -18.048351 -0.027511 -0.025601
[31] -0.006211  0.020529 -0.048781 -0.024771 -9.166311  6.953579
[37] -45.209531 -146.257211  228.237705  157.200595  10.132185 -2.582115
[49] -3.671525  200.206409  37.143079  0.078909  0.062269  0.129769
[55] -17.147005 -17.178235 -145.273195  217.232958  54.169628  17.105458
[61] -0.833452 -1.260332 -0.899352 -3.098942 -1.173512 -1.021802
[67] -0.939162 -1.007752 -1.377062 -3.022622  16.977768  17.001778
[73]  7.860238  23.980128 -28.182982 -129.230662

$mZ.ppm.error
[1]  2.403257e+06  1.006558e+06  8.482850e+04  3.319130e+04  3.793488e+01
 [6] -1.751484e+02 -5.335196e+01 -2.639286e+01 -1.285450e+02 -7.611043e+03
[11] -1.346114e+02 -1.198789e+02 -6.782037e+01 -1.552813e+01 -1.162198e+02
[16] -5.313198e+01 -6.608212e+01 -7.638202e+01 -6.066594e+04  1.360904e+06
[21]  1.917483e+05  1.986591e+02  1.183257e+02  1.980319e+02  1.397352e+02
[26]  7.115774e+01 -5.379332e+01 -1.684426e+02 -2.322450e+01 -1.948903e+01
[31] -4.485617e+00  1.370673e+01 -3.109508e+01 -1.458996e+01 -5.056331e+03
[36]  3.547913e+03 -2.226035e+04 -6.860121e+04  1.916651e+06  8.268554e+05
[41]  3.004915e+04 -5.709941e+04  2.798859e+03  4.173588e+03  1.177069e+04
[46] -1.352074e+03  1.259905e+03  1.237534e+04 -3.076091e+03  7.417604e+02
[51] -1.216230e+04  2.020566e+03 -1.060078e+04 -9.766434e+03 -9.319787e+03
[56] -8.576627e+03 -6.817113e+04  1.669915e+06  1.847849e+05  4.499286e+04
[61] -1.636709e+03 -1.974616e+03 -1.239974e+03 -3.696333e+03 -1.249174e+03
[66] -9.690310e+02 -8.043928e+02 -7.772361e+02 -1.006903e+03 -2.041339e+03
[71]  1.094110e+04  1.011538e+04  4.376983e+03  1.234257e+04 -1.399411e+04
[76] -6.110297e+04

$idx
[1]  1  1  1  3  11  20  39  45  64  90  96 106 116 121 126 129 131 133 133
[20]  1  1  2  7  24  38  49  65  90  97 110 115 122 123 127 130 132 133 133
[39]  1  1  2  3  13  23  40  47  67  91  98 108 116 122 126 129 131 133 133
[58]  1  1  2  6  21  36  47  62  90  95 108 113 121 123 127 130 132 133 133
```r
$label
[1] "b1" "b2" "b3" "b4" "b5" "b6" "b7" "b8" "b9" "b10" "b11" "b12"
[13] "b13" "b14" "b15" "b16" "b17" "b18" "b19" "y1" "y2" "y3" "y4" "y5"
[25] "y6" "y7" "y8" "y9" "y10" "y11" "y12" "y13" "y14" "y15" "y16" "y17"
[37] "y18" "y19" "c1" "c2" "c3" "c4" "c5" "c6" "c7" "c8" "c9" "c10"
[49] "c11" "c12" "c13" "c14" "c15" "c16" "c17" "c18" "c19" "z1" "z2" "z3"
[61] "z4" "z5" "z6" "z7" "z8" "z9" "z10" "z11" "z12" "z13" "z14" "z15"
[73] "z16" "z17" "z18" "z19"

$score
[1] -1

$sequence
[1] "TAFDEAIAELTLSEESYK"

$fragmentIon
     b         y        c        z
   1  102.0550  147.1128  119.0815  130.0863
   2  173.0921  310.1761  190.1186  293.1496
   3  320.1605  397.2082  337.1870  380.1816
   4  435.1874  526.2508  452.2140  509.2242
   5  564.2300  655.2933  581.2566  638.2668
   6  635.2671  742.3254  652.2937  725.2988
   7  748.3512  855.4094  765.3777  838.3829
   8  819.3883  956.4571  836.4148  939.4306
   9  948.4309 1071.4841  965.4574 1054.4575
  10 1061.5149 1184.5681 1078.5415 1167.5416
  11 1176.5419 1313.6107 1193.5684 1296.5842
  12 1277.5896 1384.6478 1294.6161 1367.6213
  13 1390.6736 1497.7319 1407.7002 1480.7053
  14 1477.7056 1568.7690 1494.7322 1551.7424
  15 1606.7482 1697.8116 1623.7748 1680.7850
  16 1735.7908 1812.8385 1752.8174 1795.8120
  17 1822.8229 1959.9069 1839.8494 1942.8804
  19 2113.9811 2131.9917 2131.0077 2114.9652

> par(op)
```
5 Quanification

For an overview on Quantitative Proteomics read [1, 2]. The authors are aware that meaningful statistics usually require much higher number of biological replicates. In almost all cases there are not more than three to six repitions. For the moment there are limited options due to the availability of machine time and the limits of the technologies.

5.1 Label-free methods on protein level

The data set fetuinLFQ contains a subset of our results described in [3]. The example below shows a visualization using trellis plots. It graphs the abundance of four protein in dependency from the fetuin concentration spiked into the sample.

```r
> library(lattice)
> data(fetuinLFQ)
> cv<-1-1:7/10
> t<-trellis.par.get("strip.background")
> t$col<-(rgb(cv,cv,cv))
> trellis.par.set("strip.background",t)
> print(xyplot(abundance~conc|prot*method,
+ groups=prot,
+ ...)
```
The plot shows the estimated concentration of the four proteins using the top three most intense peptides. The Fetuin peptides are spiked in with increasing concentration while the three other yeast proteins are kept stable in the background.

5.2 pgLFQ – LCMS based label-free quantification

LCMS based label-free quantification is a very popular method to extract relative quantitative information from mass spectrometry experiments. At the FGCZ we use the software ProgenesisLCMS for this workflow [http://www.nonlinear.com/products/progenesis/lc-ms/overview/](http://www.nonlinear.com/products/progenesis/lc-ms/overview/). Progenesis is a graphical software which does the aligning between several LCMS experiments, extracts signal intensities from LCMS maps and annotates the mastermap with peptide and protein labels.

```r
> data(pgLFQfeature)
> data(pgLFQprot)
> featureDensityPlot<-function(data, n=ncol(data), nbins=30){
+   my.col<-rainbow(n);
+   mids<-numeric()
+   density<-numeric()
+   for (i in 1:n) {
+       h<-hist(data[,i],nbins, plot=FALSE)
+       mids<-c(mids, h$mids)
+       density<-c(density, h$density)
+   }
+   plot(mids,density, type='n')
+   for (i in 1:n) {
+       h<-hist(data[,i],nbins, plot=FALSE)
+       lines(h$mids,h$density, col=my.col[i])
+   }
+   legend("topleft", names(data), cex=0.5,
+           text.col=my.col
+   )
+ }
> par(mfrow=c(1,1));
> featureDensityPlot(asinh(pgLFQfeature$"Normalized abundance"),
+ nbins=25)
```
The featureDensityPlot shows the normalized signal intensity distribution (asinh transformed) over 24 LCMS runs which are aligned in this experiment.

```r
> op<-par(mfrow=c(1,1),mar=c(18,18,4,1),cex=0.5)
> samples<-names(pgLFQfeature$"Normalized abundance")
> image(cor(asinh(pgLFQfeature$"Normalized abundance")),
+       col=gray(seq(0,1,length=20)),
+       main='pgLFQfeature correlation',
+       axes=FALSE)
> axis(1,at=seq(from=0, to=1,
+               length.out=length(samples)),
+       labels=samples, las=2)
> axis(2,at=seq(from=0, to=1,
+               length.out=length(samples)), labels=samples, las=2)
> par(op)
```
This image plot shows the correlation between runs on feature level (values are asinh transformed). White is perfect correlation while black indicates a poor correlation.

```r
op<-par(mfrow=c(1,1),mar=c(18,18,4,1),cex=0.5)
image(cor(asinh(pgLFQprot$"Normalized abundance")),
+ main='pgLFQprot correlation',
+ axes=FALSE,
+ col=gray(seq(0,1,length=20)))
axis(1,at=seq(from=0, to=1,
+ length.out=length(samples)), labels=samples, las=2)
axis(2,at=seq(from=0, to=1,
+ length.out=length(samples)), labels=samples, las=2)
par(op)
```
This figure shows the correlation between runs on protein level (values are asinh transformed). White is perfect correlation while black indicates a poor correlation. Stricking is the fact that the six biological replicates for each condition cluster very well.

```r
> par(mfrow=c(2,2),mar=c(6,3,4,1))
> ANOVA<-pgLFQaov(pgLFQprot$"Normalized abundance",
+   groups=as.factor(pgLFQprot$grouping),
+   names=pgLFQprot$output$Accession,
+   idx=c(15,16,196,107),
+   plot=TRUE)
```
This figure shows the result for four proteins which either differ significantly in expression across conditions (green boxplots) using an analysis of variance test, or non differing protein expression (red boxplot).

5.3 iTRAQ – Two Group Analysis

The data for the next section is an iTRAQ-8-plex experiment where two conditions are compared (each condition has 4 biological replicates)

5.3.1 Sanity Check

```r
> data(iTRAQ)
> x<-rnorm(100)
> par(mfrow=c(3,3),mar=c(6,4,3,0.5));
> for (i in 3:10){
+   qqnorm(asinh(iTRAQ[,i]),
+   main=names(iTRAQ)[i])
+   qqline(asinh(iTRAQ[,i]), col='grey')
+ }
> b<-boxplot(asinh(iTRAQ[,c(3:10)]), main='boxplot iTRAQ')
```
A first quality check to see if all reporter ion channels are having the same distributions. Shown in the figure are Q-Q plots of the individual reporter channels against a normal distribution. The last is a boxplot for all individual channels.

5.3.2 On Protein Level

```R
> data(iTRAQ)
> group1Protein<-numeric()
> group2Protein<-numeric()
> for (i in c(3,4,5,6))
+   group1Protein<-cbind(group1Protein,
+                       asinh(tapply(iTRAQ[,i], paste(iTRAQ$prot), sum, na.rm=TRUE)))
> for (i in 7:10)
+   group2Protein<-cbind(group2Protein,
+                       asinh(tapply(iTRAQ[,i], paste(iTRAQ$prot), sum, na.rm=TRUE)))
> par(mfrow=c(2,3),mar=c(6,3,4,1))
> for (i in 1:nrow(group1Protein)){
+   boxplot.color="#ffcccc"
+   tt.p_value<-t.test(as.numeric(group1Protein[i,]),
+                       as.numeric(group2Protein[i,]))$p.value
+   if (tt.p_value < 0.05)
```
This figure shows five proteins which are tested if they differ across conditions using the four biological replicates with a t-test.

### 5.3.3 On Peptide Level

The same can be done on peptide level using the protViz function iTRAQ2GroupAnalysis.

```r
> data(iTRAQ)
> q<-iTRAQ2GroupAnalysis(data=iTRAQ,
```
```r
+ group1=c(3,4,5,6),
+ group2=7:10,
+ INDEX=paste(ITRAQ$prot, ITRAQ$peptide),
+ plot=FALSE)
> q[1:10,]

<table>
<thead>
<tr>
<th>name</th>
<th>p_value</th>
<th>Group1.area113</th>
<th>Group1.area114</th>
</tr>
</thead>
<tbody>
<tr>
<td>O95445 AFLLTTPR</td>
<td>0.056</td>
<td>1705.43</td>
<td>1459.10</td>
</tr>
<tr>
<td>O95445 DGLCVPR</td>
<td>0.161</td>
<td>2730.41</td>
<td>1852.90</td>
</tr>
<tr>
<td>O95445 MKDGLCVPR</td>
<td>0.039</td>
<td>28726.38</td>
<td>15409.81</td>
</tr>
<tr>
<td>O95445 NQEACELSNN</td>
<td>0.277</td>
<td>4221.31</td>
<td>4444.28</td>
</tr>
<tr>
<td>O95445 SLTSCLDSK</td>
<td>0.036</td>
<td>20209.66</td>
<td>14979.02</td>
</tr>
<tr>
<td>P02652 AGTELVNLFSYFVELGTQPA</td>
<td>0.640</td>
<td>4504.97</td>
<td>4871.88</td>
</tr>
<tr>
<td>P02652 AGTELVNLFSYFVELGTQPAT</td>
<td>0.941</td>
<td>67308.30</td>
<td>46518.21</td>
</tr>
<tr>
<td>P02652 AGTELVNLFSYFVELGTQPATQ</td>
<td>0.338</td>
<td>4661.54</td>
<td>3971.82</td>
</tr>
<tr>
<td>P02652 EPCVESLVSQYFQTVTDYGK</td>
<td>0.115</td>
<td>4544.56</td>
<td>4356.51</td>
</tr>
<tr>
<td>P02652 EQLTPLIK</td>
<td>0.053</td>
<td>24596.42</td>
<td>22015.94</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>name</th>
<th>Group1.area115</th>
<th>Group1.area116</th>
<th>Group2.area117</th>
<th>Group2.area118</th>
<th>Group2.area119</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>770.65</td>
<td>3636.40</td>
<td>3063.48</td>
<td>4046.73</td>
<td>2924.49</td>
</tr>
<tr>
<td></td>
<td>1467.65</td>
<td>2266.88</td>
<td>2269.57</td>
<td>3572.32</td>
<td>2064.82</td>
</tr>
<tr>
<td></td>
<td>19050.13</td>
<td>58185.02</td>
<td>51416.05</td>
<td>70721.05</td>
<td>38976.42</td>
</tr>
<tr>
<td></td>
<td>2559.23</td>
<td>6859.71</td>
<td>5545.12</td>
<td>11925.66</td>
<td>6371.50</td>
</tr>
<tr>
<td></td>
<td>12164.94</td>
<td>37572.56</td>
<td>30687.57</td>
<td>39176.99</td>
<td>34417.66</td>
</tr>
<tr>
<td></td>
<td>2760.53</td>
<td>9213.41</td>
<td>6728.62</td>
<td>14761.96</td>
<td>7796.29</td>
</tr>
<tr>
<td></td>
<td>33027.14</td>
<td>111629.30</td>
<td>94531.76</td>
<td>168775.00</td>
<td>83526.72</td>
</tr>
<tr>
<td></td>
<td>2564.39</td>
<td>8269.73</td>
<td>6045.30</td>
<td>13724.92</td>
<td>7426.84</td>
</tr>
<tr>
<td></td>
<td>2950.48</td>
<td>6357.90</td>
<td>6819.99</td>
<td>10265.84</td>
<td>7012.92</td>
</tr>
<tr>
<td></td>
<td>18424.56</td>
<td>49811.91</td>
<td>33197.47</td>
<td>67213.62</td>
<td>40030.86</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>name</th>
<th>Group2.area121</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5767.87</td>
</tr>
<tr>
<td></td>
<td>2208.92</td>
</tr>
<tr>
<td></td>
<td>60359.72</td>
</tr>
<tr>
<td></td>
<td>15656.92</td>
</tr>
<tr>
<td></td>
<td>54439.22</td>
</tr>
<tr>
<td></td>
<td>18681.60</td>
</tr>
<tr>
<td></td>
<td>168032.50</td>
</tr>
<tr>
<td></td>
<td>17214.87</td>
</tr>
<tr>
<td></td>
<td>14279.22</td>
</tr>
<tr>
<td></td>
<td>87343.38</td>
</tr>
</tbody>
</table>

### 6 Pressure Profiles QC

A common problem with mass spec setup is the pure reliability of the high pressure pump. The following graphics provide visualizations for quality control.

An overview of the pressure profile data can be seen by using the `ppp` function.

```r
> data(pressureProfile)
> ppp(pressureProfile)
```
The lines plots the pressure profiles data on a scatter plot 'Pc' versus 'time' grouped by time range (no figure because of too many data items).

The Trellis xyplot shows the Pc development over each instrument run to a specified relative run time (25,30,....).

```r
> pp.data<-pps(pressureProfile, time=seq(25,40,by=5))
> print(xyplot(Pc ~ as.factor(file) | paste("time =",
+ as.character(time), "minutes"),
+ panel = function(x, y){
+ m<-sum(y)/length(y)
+ m5<-(max(y)-min(y))*0.05
+ panel.abline(h=c(m-m5,m,m+m5),
+ col=rep("#ffcccc",3),lwd=c(1,2,1))
+ panel.grid(h=-1, v=0)
+ panel.xyplot(x, y)
+ },
+ ylab='Pc [psi]',
+ layout=c(1,4),
+ sub='The three red lines indicate the average plus min 5%.',
+ scales = list(x = list(rot = 45)),
+ data=pp.data))
```

The three red lines indicate the average plus min 5%.
While each panel in the `xyplot` above shows the data to a given point in time, we try to use the `levelplot` to get an overview of the whole pressure profile data.

```r
> pp.data<-pps(pressureProfile, time=seq(0,140,length=128))
> print(levelplot(Pc ~ time * as.factor(file),
+     main='Pc(psi)',
+     data=pp.data,
+     col.regions=rainbow(100)[1:80]))
```

7 Session information

```r
> sessionInfo()

R version 3.1.1 (2014-07-10)
Platform: x86_64-pc-linux-gnu (64-bit)

locale:
[1] LC_CTYPE=en_US.UTF-8 LC_NUMERIC=C
[3] LC_TIME=en_US.UTF-8 LC_COLLATE=C
[5] LC_MONETARY=en_US.UTF-8 LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=en_US.UTF-8 LC_NAME=C
[9] LC_ADDRESS=C LC_TELEPHONE=C
```

26
attached base packages:
[1] stats graphics grDevices utils datasets methods base

other attached packages:
[1] lattice_0.20-29 protViz_0.2.9

loaded via a namespace (and not attached):
[1] Rcpp_0.11.2 grid_3.1.1 tools_3.1.1

> packageDescription('protViz')

Package: protViz
Type: Package
Title: Visualizing and Analyzing Mass Spectrometry Related Data in Proteomics
Version: 0.2.9
Date: 2014-09-30
Author: Christian Panse <cp@fgcz.ethz.ch>, Jonas Grossmann 
<jg@fgcz.ethz.ch>, Simon Barkow-Oesterreicher <sb@fgcz.ethz.ch>
Maintainer: Christian Panse <cp@fgcz.ethz.ch>
Depends: R (>= 3.0.2), methods
Imports: Rcpp (>= 0.9.9)
Suggests: lattice, RUnit, BiocGenerics
Description: This R package helps with quality checks, visualizations and analysis of mass spectrometry data, coming from proteomics experiments. The package is developed, tested and used at the Functional Genomics Center Zurich. We use this package mainly for prototyping, teaching, and having fun with proteomics data. But it can also be used to do data analysis for small scale data sets.
License: GPL-3
URL: http://fgcz-data.uzh.ch/~cpanse/protViz/
LazyData: true
Built: R 3.1.1; x86_64-pc-linux-gnu; 2014-09-30 08:13:18 UTC; unix

-- File: /tmp/Rtmpz1UyJ4/Rinst7a632ccfeacf/protViz/Meta/package.rds

References

