ADP Ribosylated Peptides

Peter Gehrig[1], Kathrin Nowak, Jonas Grossmann[1,2], Christian Panse[1,2,⋆]

[1]Functional Genomics Center Zurich, UZH|ETHZ
Winterthurerstr. 190
CH-8057, Zürich, Switzerland
Telephone: +41-44-63-53912
E-mail⋆: cp@fgcz.ethz.ch
URL: https://www.fgcz.ch

[2]Swiss Institute of Bioinformatics
URL: https://www.sib.swiss

1 Illustrations

This document reproduces Figures 1–5 presented in Gehrig et al. (2020). For a description of the theory behind applications shown here we refer to the original manuscript. The results differ slightly due to technical changes or bugfixes in **protViz** that have been implemented after the manuscript was printed.

Data preprocessing The mass spectrometric data were previously extracted from PRIDE PXD017013 using the Bioconductor package **rawrr** Kockmann and Panse (2021) and the following code snippet.

```
> rawUrl <- paste0("http://ftp.pride.ebi.ac.uk",
+   "/pride/data/archive/2021/05/PXD017013/20171220_15_Muscle_HCD35.raw")
> f <- basename(rawUrl)
> download.file(rawUrl, f )
> scans <- c(9210, 13738, 14908, 7590, 10718)
> ## read spectra
> ## remove peaks with no intensity
> ADPR.ms2 <- rawrr::readSpectrum(f, scans) |>
+   lapply(function(x){
+     idx <- x$intensity > 0
+     list(mZ=x$mZ[idx], intensity=x$intensity[idx], scan=x$scan)
+   })
> ## peak assignments
> ADPR.annotation <-
+   readr::read_delim("/Users/cp//Downloads/2020-05-27_InputToLabelSpectra.tsv",
```

```
+                 "\t", escape_double = FALSE, trim_ws = TRUE)
> ## subsetting
> ADPR.annotation <-
+   ADPR.annotation[,c('scanNr', 'PepSeq', 'mz', 'LabelLow', 'color')] |>
+   as.data.frame()
> ## some render metadata
> ADPR.lim <- readr::read_delim("/Users/cp/Downloads/lim.txt",
+                   ",", escape_double = FALSE, trim_ws = TRUE) |>
+   as.data.frame()
> save(ADPR.annotation, ADPR.ms2, ADPR.lim,
+      file="/tmp/ADPR.RData", compression_level = 9, compress = TRUE)
```

Define helper function

```
> ## Heuristic to determine a useful y-axis range.
> ## While we deal with profile data we have to
> ## find the most intense peak within a mass window.
> .findLocalMaxIntensity <-
+   function(q, mZ, intensity, stepsize = 20, eps = 0.005){
+   n <- length(mZ)
+   idx <- protViz::findNN(q, mZ) |>
+     vapply(function(i){
+     i.max <- i
+
+     for (j in seq(i - stepsize, i + stepsize)){
+       if(0 < j & j <= n)
+         if (intensity[j] > intensity[i.max])
+           i.max <- j
+     }
+     i.max
+   }, FUN.VALUE = 1)
+
+   intensity[idx]
+ }

> ## Adapted protViz::peakplot plot function
> .peakplot <-
+   function(x, mZ, intensity, lim, ...){
+     p.i <- .findLocalMaxIntensity(x$mz, mZ, intensity)
+     sn <- unique(x$scanNr)
+     cutoff <- max(p.i) * lim$rintensity / 100
+
+     plot(intensity ~ mZ,
+          type = 'h',
+          xlab = 'm/z',
+          ylab = 'Relative Intensity [%]',
```

```
+          col = 'lightgrey',
+          xlim = c(lim$xmin, lim$xmax),
+          ylim = c(0,  cutoff),
+          axes = FALSE);
+
+     legend("topright", "",  title= unique(x$PepSeq), bty='n',cex=2)
+     legend("right", sprintf("% 10.3f   %s", x$mz,x$LabelLow),
+            title= "Fragment Ions", bty='n',cex=0.75)
+
+     axis(2, seq(0, max(intensity), length=11), round(seq(0, 100, length = 11)))
+
+     points(x$mz, p.i,  col=x$color, type='h', lwd=2)
+     points(x$mz, p.i,  col=x$color, pch=16,cex=0.5)
+
+     select <- p.i < 0.75 * max(intensity)
+
+     text(x$mz, p.i + 0.0125 * cutoff,
+          x$LabelLow, adj = c(0,0), cex=1.0, srt=90, , col=x$color)
+     idx <- p.i > cutoff
+
+     axis(1)
+     axis(3, x$mz[idx],
+          paste(x$LabelLow[idx], "(", round(100 * p.i[idx] / max(p.i)), "%)", sep=''),
+          cex=0.3)
+     box()
+   }
```

Drawing

```
> scan <- 9210
> idx <- which(vapply(ADPR.ms2, function(x)x$scan, 1) == scan)
> lim <- ADPR.lim[ADPR.lim$scan==scan,]
> .peakplot(
+     x = ADPR.annotation[ADPR.annotation$scanNr == scan,],
+     mZ = ADPR.ms2[[idx]]$mZ,
+     intensity = ADPR.ms2[[idx]]$intensity,
+     lim)
```

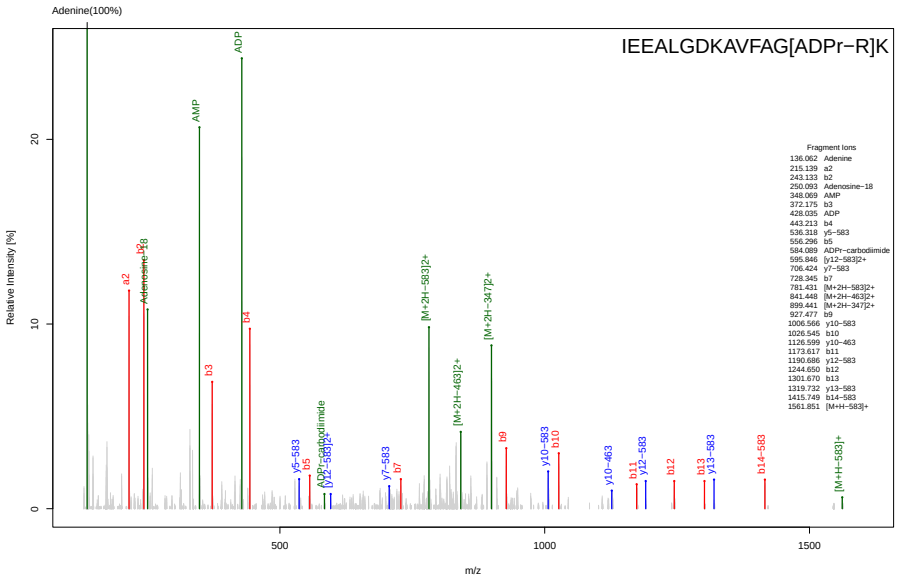

Figure 1: High-resolution HCD fragmentation spectrum of the triply charged peptide `IEEALGDKAVFAGR*K`, which is ADP-ribosylated on the arginine residue. The N-terminal ion series are shown in red, the C-terminal ion series are in blue, and the ADP-ribosylation-specific marker ions and neutral losses from peptide ions are indicated in green.

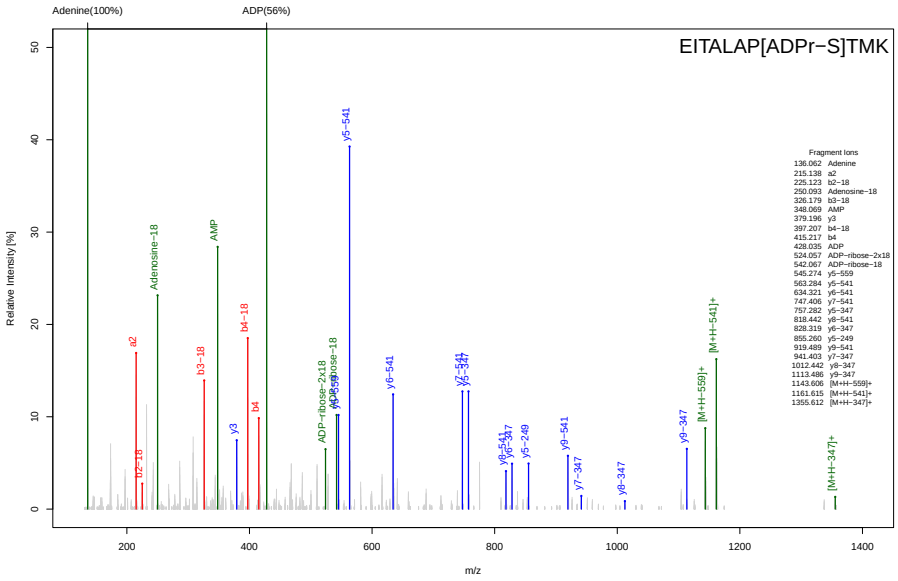

Figure 2: HCD fragmentation spectrum of the doubly charged peptide `EITALAPS*TMK`, which is ADP-ribosylated on the serine residue.

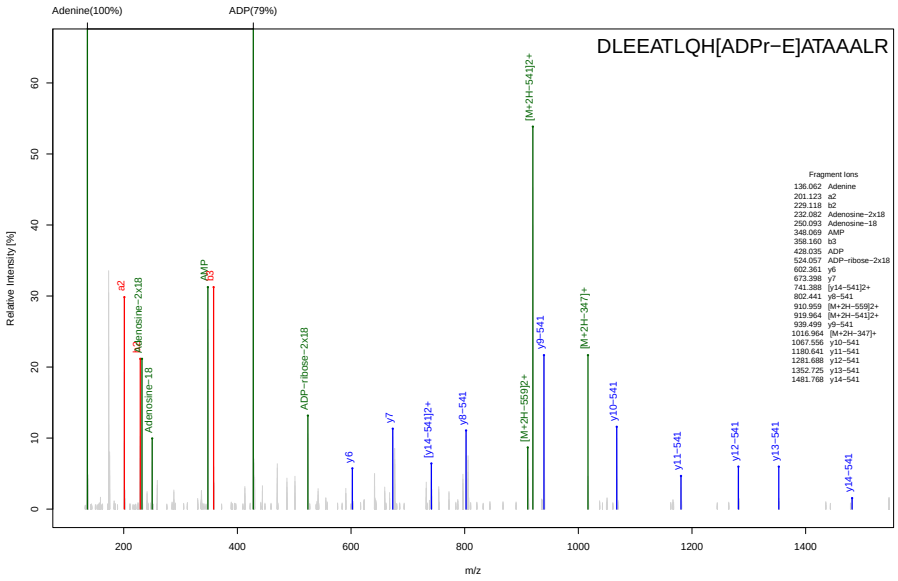

Figure 3: HCD spectrum of the triply charged peptide `DLEEATLQHE*ATAAALR`, which is ADP-ribosylated on the indicated glutamic acid residue.

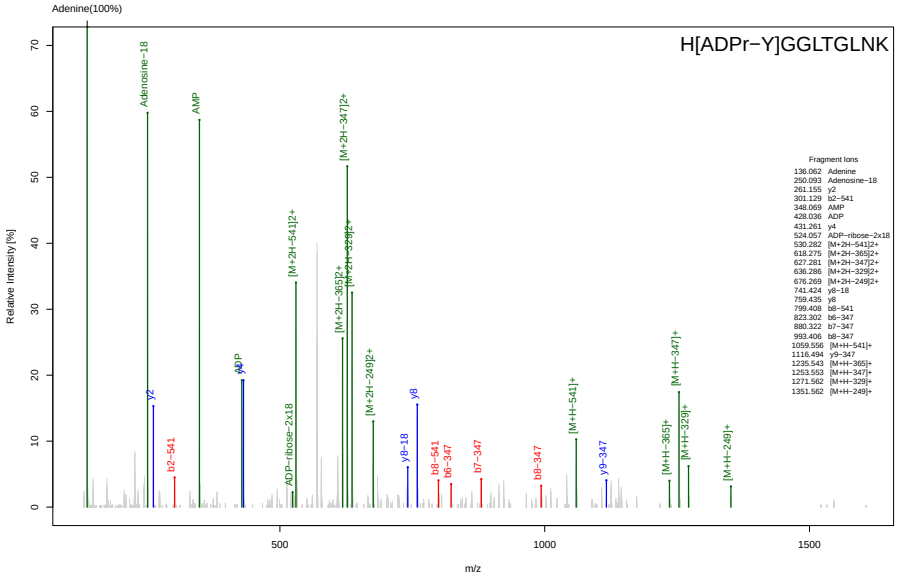

Figure 4: HCD spectrum of the doubly charged peptide `HY*GGLTGLNK`, which is ADP-ribosylated on the tyrosine residue.

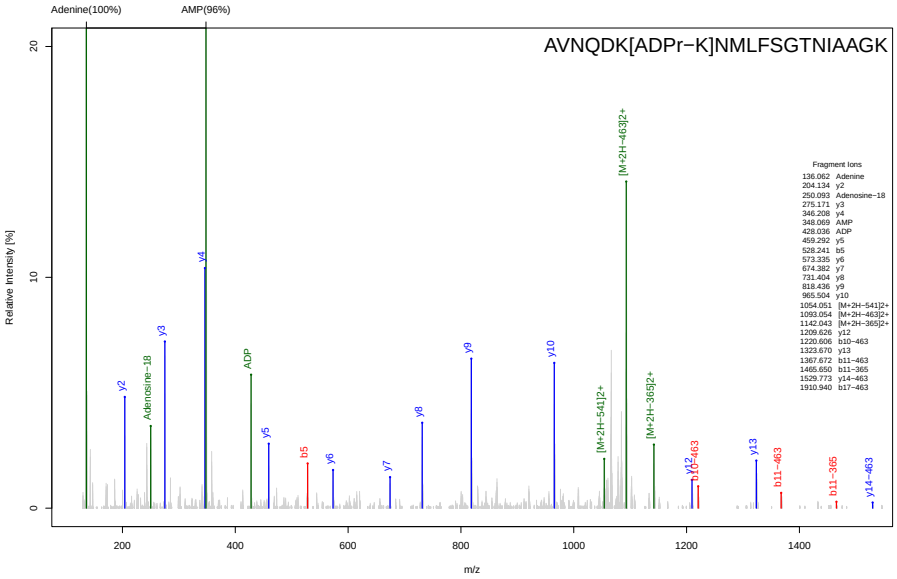

Figure 5: HCD spectrum of the triply charged peptide `AVN-QDKK*NMLFSGTNIAAGK`, which is primarily ADP-ribosylated on the indicated lysine and to a minor extent on the preceding lysine.

References

Peter M. Gehrig, Kathrin Nowak, Christian Panse, Mario Leutert, Jonas Grossmann, Ralph Schlapbach, and Michael O. Hottiger. Gas-phase fragmentation of ADP-ribosylated peptides: Arginine-specific side-chain losses and their implication in database searches. *Journal of the American Society for Mass Spectrometry*, 32(1):157–168, November 2020. doi: 10.1021/jasms.0c00040. URL https://doi.org/10.1021/jasms.0c00040.

Tobias Kockmann and Christian Panse. The rawrr R package: Direct access to orbitrap data and beyond. *Journal of Proteome Research*, 2021. doi: 10.1021/acs.jproteome.0c00866. URL https://doi.org/10.1021/acs.jproteome.0c00866.