

# Handling Modifications with *MSnID*

***Vladislav A. Petyuk***

October 27, 2020

## Contents

1	Introduction . . . . .	1
2	Reading and Brief Info on Present Mods . . . . .	1
3	Encoding Mods with Characters . . . . .	2
4	Mapping Sites to Protein Sequence . . . . .	3
5	Re-Doing Mapping with Gene IDs . . . . .	4

## 1 Introduction

---

This vignette describes handling modifications of the peptides. Modifications can be biologically-relevant and introduced after protein translation, thus post-translational modifications or PTMs. Modification can also be introduced as artifact during sample processing. Here we use more general term - modification, that encompasses both PTMs, artifacts and intentional modifications during the sample preparation.

## 2 Reading and Brief Info on Present Mods

---

```
> m <- MSnID(".")
> mzids <- system.file("extdata", "phospho.mzid.gz", package="MSnID")
> m <- read_mzIDs(m, mzids)

reading phospho.mzid.gz... DONE!
```

## Handling Modifications with *MSnID*

Method `report_mods` returns the table with masses and their counts within the dataset. This is a quick way to get insight on what is present. The other useful piece of information is the exact masses of modifications. In the later steps we will be using them to encode with characters, typically asterisk, which is a rather common representation of modified peptides.

```
> # to know the present mod masses
> report_mods(m)

      229.1629 57.021463735 79.966330925
           164           15           142
```

### 3 Encoding Mods with Characters

A common way to denote the position and type of modification is with a non-alphanumeric character. E.g. `X.XXXX*XXXX.X` means the modification at 4th residue. Typically it is most interest to map modifications that were dynamic in the MS/MS search. In this example TMT (229.1629) and cystein alkylation (57.021463735) are static modifications. The 79.966330925 is dynamic (that is may or maynot be present) phosphorylation.

Note, `add_mod_symbol` added `peptide_mod` column.

```
> m <- add_mod_symbol(m, mod_mass="79.966330925", symbol="*")
> x <- psms(m) %>%
+   distinct(modification, peptide, peptide_mod)
```

Sample of the table:

modification	peptide	peptide_mod
229.1629 (0), 79.966330925 (3), 79.966330925 (5)	R.SRTHSTSSSLGSGESPF.S	R.SRT*HS*TSSSLGSGESPF.S
229.1629 (0), 79.966330925 (8)	R.ASAVSELS.P.R	R.ASAVSELS*P.R
229.1629 (0), 79.966330925 (1), 79.966330925 (3)	R.SRSPLAIR.R	R.S*RS*PLAIR.R
229.1629 (0), 57.021463735 (8), 229.1629 (10)	R.ILPNPDECDK.V	R.ILPNPDECDK.V
229.1629 (0), 57.021463735 (15)	R.NTTSDVAVVVNDEHCR.T	R.NTTSDVAVVVNDEHCR.T

We can map additional modifications. Although typically, a given study focuses on one PTM at a time. Nonetheless:

```
> m <- add_mod_symbol(m, mod_mass="229.1629", symbol="#")
> m <- add_mod_symbol(m, mod_mass="57.021463735", symbol="^")
> x <- psms(m) %>%
+   distinct(modification, peptide, peptide_mod)
```

Sample of the table:

modification	peptide	peptide_mod
229.1629 (0), 79.966330925 (3), 79.966330925 (5)	R.SRTHSTSSSLGSGESPFGR.S	R.#SRT*HS*TSSSLGSGESPFGR.S
229.1629 (0), 79.966330925 (8)	R.ASAVSELSR.E	R.#ASAVSELS*PR.E
229.1629 (0), 79.966330925 (1), 79.966330925 (3)	R.SRSPLAIR.R	R.#S*RS*PLAIR.R
229.1629 (0), 57.021463735 (8), 229.1629 (10)	R.ILPNPDECDK.V	R.#ILPNPDEC^DK#.V
229.1629 (0), 57.021463735 (15)	R.NTTSDVAVVVNDEHCR.T	R.#NTTSDVAVVVNDEHC^R.T

## 4 Mapping Sites to Protein Sequence

Somewhat conventional form of PTM notation (or non-synonymous mutations) is gene/protein ID followed by AA code in upper case, position in the sequence and original AA shows as low case. For example, phosphorylation of serine at position 473 of AKT1 would look like AKT1-S473s.

Besides *MSnID* object, the key component to mapping the modifications is the FASTA file with protein sequences.

```
> fst_path <- system.file("extdata", "for_phospho.fasta.gz", package="MSnID")
> fst <- readAAStringSet(fst_path)
```

When we link accession IDs with FASTA entry names, obviously they need to be in the same format. So in this case we have to trip the names in FASTA.

```
> names(fst) <- sub("^[^ ]*" .*$", "\\1", names(fst))
```

The core method for mapping modification sites. The warning message it gives about extra characters in peptide sequences is about "#" and "^" we used to denote TMT modification and alkylation. They are ignored during mapping.

```
> m <- map_mod_sites(m, fst,
+                   accession_col = "accession",
+                   peptide_mod_col = "peptide_mod",
+                   mod_char = "*",
+                   site_delimiter = "lower")
> x <- psms(m) %>%
+   distinct(peptide_mod, SiteID)
```

peptide_mod	SiteID
R.#SRT*HS*TSSSLGSGESPF.S	sp Q9UGV2 NDRG3_HUMAN-T329tS331s
R.#ASAVSELS*PR.E	sp Q9Y2W1 TR150_HUMAN-S243s
R.#S*RS*PLAIR.R	sp Q9UQ35 SRM2_HUMAN-S2044sS2046s
K.#NGVAAEVS*PAK#.E	sp Q9BW71 HIRP3_HUMAN-S125s
R.#S*GPRSAQRR.N	sp P84996 ALEX_HUMAN-S593s

## 5 Re-Doing Mapping with Gene IDs

The most common (human readable and somewhat comprehensible) identifier of proteins and corresponding genes is gene symbol. For example, **sp|P84996|ALEX\_HUMAN** UniProt ID corresponds to **GNAS** gene symbol. In this section we'll remap UniProt IDs to gene symbols and report Site IDs in a more human readable way.

First, we'll download a table converting from one ID to another. There are multiple ways how one can get this type of table. In this example we implicitly use *AnnotationHub* package.

```
> conv_tab <- fetch_conversion_table("Homo sapiens", "UNIPROT", "SYMBOL")
> head(conv_tab)
```

	UNIPROT	SYMBOL
1	P04217	A1BG
2	V9HWD8	A1BG
3	P01023	A2M
5	P18440	NAT1
6	Q400J6	NAT1
7	F5H5R8	NAT1

Re-mapping accessions from IDs indicated in the first columns of the `conv_tab` to the second. The accessions in the *MSnID* object may not be in exactly in the same form as in the database used to fetch `conv_tab` conversion table. Thus, there is the `extraction_pttrn` argument. It extracts the first matching group "\\1" as the proper ID. There are three suggested extraction patterns for UniProt, RefSeq and ENSEMBL. In case the accession is more complicated than that, user can provide a custom extraction pattern.

```
> head(accessions(m))

[1] "sp|Q9UGV2|NDRG3_HUMAN" "sp|075533|SF3B1_HUMAN" "sp|Q13442|HAP28_HUMAN"
[4] "sp|015075|DCLK1_HUMAN" "sp|Q9Y2W1|TR150_HUMAN" "sp|076094|SRP72_HUMAN"

> m <- remap_accessions(m, conv_tab, extraction_pttrn = "\\|([^-]+)(-\\d+)?\\|")
> head(accessions(m))

[1] "NDRG3" "SF3B1" "PDAP1" "DCLK1" "THRAP3" "SRP72"
```

## Handling Modifications with *MSnID*

Since we updated the accessions in the *MSnID* object, we need to provide FASTA file with corresponding entry names if we want to map the PTM sites. If such FASTA file isn't readily available, which is very likely if the *MSnID* object accessions converted to gene symbols. Entry names can be updated using the same conversion table.

```
> fst_path <- system.file("extdata", "for_phospho.fasta.gz", package="MSnID")
> fst_path_2 <- remap_fasta_entry_names(fst_path, conv_tab, "\\|([^-]+)(-\\d+)?\\|")
> library(Biostrings)
> readAAStringSet(fst_path)
```

AAStringSet object of length 45:

	width	seq	names
[1]	715	MSSPKRSSKPSMSLAPSGSSMPT...DATTSKATLPGERSSSSSSSKLA	sp B3KS81 SRRM5_H...
[2]	740	MSFGRDMELEHFDERDKAQRYSR...NSESEDYSPSSSETVRSPNSPF	sp 015075 DCLK1_H...
[3]	508	MTQTLKYASRVFHRVWAPELGA...VRPKTRTVLVPERSINLQFLDR	sp 015528 CP27B_H...
[4]	247	MDAFTRTNQTQGRDRLFRATQY...GLVSSIAGMITVAYPQMKLKTR	sp 075192 PX11A_H...
[5]	1304	MAKIAKTHEDIEAQIREIQGKKA...HYPRIYNDKNTYIRYELDYIL	sp 075533 SF3B1_H...
...	...	...	...
[41]	2752	MYNGIGLPTPRGSGTNGYVQRNL...SHKRRRETSPRPMRHRSSRSP	sp Q9UQ35 SRRM2_H...
[42]	1087	MTTESGSDSESKPDQEAEPQEAA...DMSVTKVVVHKETEITPEDGED	sp Q9Y2J2 E41L3_H...
[43]	955	MSKTNKSKSGSRSSRSASRSR...IEDDESGTENREEKDNIQPTTE	sp Q9Y2W1 TR150_H...
[44]	1467	MSDESASGSDPDLDPDVELEDAE...EVGFSSNDDKDDDVIEVTGK	sp Q9Y4B4 ARIP4_H...
[45]	313	MSDLLLGLIGGLTLLLLLTLA...GTEPLGTTKWLWEPTAPEKGKE	sp Q9Y6I9 TX264_H...

```
> readAAStringSet(fst_path_2)
```

AAStringSet object of length 45:

	width	seq	names
[1]	1439	MASSETAIRWAEPGLGKGPQRRR...QMRSSLSADLRQAHSRLRGSCLF	AKNA
[2]	1491	MNGVAFCLVGIPRPEPRPPQLP...DTGSLQSQPPRRSAASRLHQCL	ARHGAP23
[3]	3046	MRGRRGRPPKQPAAPAAERCAPA...VQKLKGFKASRSHNNKLQSTAS	BPTF
[4]	1249	MSSMWSEYTIGGVKIYFPYKAYP...EIEIKNFKPSPSKNKGMFPGFK	BRIP1
[5]	619	MAAAPPLSKAEYLKRYLSGADAG...FARLASKKAVEELAYKWSVEDM	BUD13
...	...	...	...
[41]	313	MSDLLLGLIGGLTLLLLLTLA...GTEPLGTTKWLWEPTAPEKGKE	TEX264
[42]	955	MSKTNKSKSGSRSSRSASRSR...IEDDESGTENREEKDNIQPTTE	THRAP3
[43]	843	MDKENSVDVSAAPADLKISNISVQ...PGVYTRVSNFVPIHKYVPSLL	TMPRSS7
[44]	979	MSPLKIHGPIRISMQTGITKWK...LETEKNSQSLSTEVGKTRQAL	USP37
[45]	241	MNSGRPETMENLPALYTIFQGEV...SKDSKAACKKKKKKKKKKHKE	ZCCHC17

Now we can execute the same remapping, but using gene symbols as protein IDs.

```
> fst <- readAAStringSet(fst_path_2)
> m <- map_mod_sites(m, fst,
+                     accession_col = "accession",
+                     peptide_mod_col = "peptide_mod",
+                     mod_char = "*",
```

## Handling Modifications with *MSnID*

```
+           site_delimiter = "lower")
> x <- psms(m) %>%
+   distinct(peptide_mod, SiteID)
```

peptide_mod	SiteID
R.#SRT*HS*TSSSLGSGESPFSR.S	NDRG3-T329tS331s
R.#ASAVSELS*PR.E	THRAP3-S243s
R.#S*RS*PLAIR.R	SRRM2-S2044sS2046s
K.#NGVAAEVS*PAK#.E	HIRIP3-S125s
R.#S*GPRSAQRR.N	GNAS-S593s