MDstatsDIAMS for Peptide-Level Differential Analysis of DIA Data

Namgil Lee*, Hojin Yoo†, Juhyoung Kim‡, Heejung Yang§

July 10, 2025

Contents

This is a tutorial for performing differential analysis with **MDstatsDIAMS** using quantitative DIA reports obtained from various software platforms.

```
library(MDstatsDIAMS)
```

Converting DIA Reports into Standard Format

To convert various report formats into **MDstatsDIAMS** standard format, **MDstatsDIAMS** provides both direct methods and indirect methods. For direct methods, **Spectronaut**, **MaxQuant**, **Skyline**, and **MSstats** report formats can be directly converted into standard report format. For indirect methods, **MSstats** converters can convert various report formats into **MSstats** report format, and **MDstatsDIAMS** can convert it into standard report format. Here are some examples.

> ***NOTE:*** All the report data files in this tutorial are available at https://doi.org/10.5281/zenodo.15653979 with the DOI 10.5281/zenodo.15653979. Each file can be accessed directly via the url https://zenodo.org/records/15653980/files/%60file_name‘.

Importing Spectronaut Report

A **Spectronaut** report can be converted into standard format directly.

```
# Load Spectronaut report
# Alternatively, a bigger report file containing all proteins is available by
# > arrow::read_parquet(paste0(
#     "/Users/namgil/Documents/Projects/MDstatsDIAMS/data/",
#     "lip_quant_staurosporine_hela_sn_report.parquet"))
sn_report <- read.delim(
  paste0(
    "/Users/namgil/Documents/Projects/MDstatsDIAMS/data/",
```

*Kangwon National University, and Bionsight Inc., namgil.lee@kangwon.ac.kr
†Bionsight Inc.
‡Kangwon National University
§Kangwon National University, and Bionsight Inc., heejyang@kangwon.ac.kr

```
    "sample_spectronaut_report.tsv"
  )
)

standard_report <- convert_sn_to_standard(sn_report)
```

Importing MaxQuant Tables

MaxQuant report tables consist of `evidence.txt`, `msms.txt`, and `proteinGroups.txt` files. They can be converted into a standard report as follows.

```
## Load MaxQuant evidence.txt and msms.txt files
mq_root <- paste0("/Users/namgil/Documents/Projects/MDstatsDIAMS/",
                  "data/lip_quant_staurosporine_hela_mq_report_4conds/")

mq_ev <- read.delim(paste0(mq_root, "evidence.txt"))
mq_msms <- read.delim(paste0(mq_root, "msms.txt"))

##  It is required to prepare an annotation data frame in prior.
mq_an <- create_annotation_df(
  evidence = mq_ev,
  n_replicates_per_condition = 4
)

## The **MaxQuant** is converted into standard format.
standard_report <- convert_mq_to_standard(mq_ev, mq_msms, mq_an)
```

Importing Skyline Transition Report

A **Skyline** transition report can be converted into standard format after an annotation data is prepared.

```
# Load Skyline transition report
sk_report <- arrow::read_parquet(
  paste0(
    "/Users/namgil/Documents/Projects/MDstatsDIAMS/data/",
    "lip_quant_staurosporine_hela_sk_transition_4conds.parquet"
  )
)

# Make an annotation data frame
sk_an <- data.frame(
  Condition = rep(paste0("CON", c(1, 4, 5, 8)), each = 4),
  Replicate = paste0("StauroDoseResp-", c(1:4, 17:24, 33:36)),
  Run = paste0("StauroDoseResp-", c(1:4, 17:24, 33:36))
)

# Convert to standard format
standard_report <- convert_sk_to_standard(sk_report, annotation = sk_an)
```

Importing MSstats Report

MSstats report can be generated by running MSstatsConvert::*toMSstatsFormat(). In this way, various report formats can be preprocessed and converted into **MSstats** report format. An **MSstats** report can then be converted into **MDstatsDIAMS** standard format by using the function `convert_ms_to_standard()`:

standard_report <- convert_ms_to_standard(ms_stats_report)

Running Statistical Tests of Mean Differences Between Two Groups

In **MDstatsDIAMS**, peptide-level differential analysis methods are available, including fundamental t-test methods (`paired`: paired t-test, `independent`: independent samples t-test, `shrinkage`: shrinkage t-test), as well as external methods (`msstatslip`: **MSstatsLiP**, `rots`: **ROTS**).

The `run_ttests()` will run statistical methods specified by `method_names =` for comparing a base condition (e.g., "DMSO") with all the other conditions. If `method_names = NULL` (default), all the available methods will be run.

```
### Select a subset of 100 proteins to reduce time cost. ###
### You can skip these lines.                             ###
proteins <- unique(standard_report$protein_id)
n_sampled_proteins <- min(100, length(proteins))
set.seed(1111)
sampled <- sample(proteins, n_sampled_proteins)

standard_report <- standard_report[standard_report$protein_id %in% sampled, ]
#################################################################

test_results <- run_ttests(
  report = standard_report, method_names = NULL
)
```

```
## Running the test methods: msstatslip rots paired independent shrinkage
```

```
## msstatslip ...
```

```
##   |                                                                      |
##   |                                                                      |
##   |                                                                      |
##   |                                                                      |
##   |                                                                      |
##   |                                                                      |
##   |                                                                      |
##   |                                                                      |
##   |                                                                      |
##   |                                                                      |
##   |                                                                      |
##   |                                                                      |
##   |                                                                      |
##   |                                                                      |
```

```
## rots ...
```

```
## paired ...
```

```
## independent ...
```

```
## shrinkage ...
```

```
print(paste(c("Methods:", names(test_results)), collapse = " "))
```

```
## [1] "Methods: msstatslip rots paired independent shrinkage"
```

```
print(paste(c("Comparisons:", names(test_results[[1]])), collapse = " "))
```

```
## [1] "Comparisons: DMSO/100mM DMSO/100nM DMSO/100pM DMSO/10mM DMSO/10nM DMSO/1mM DMSO/1nM"
```

The test results obtained by `run_ttests()` have `p.value` column in each table for every comparisons. The **MDstatsDIAMS** can compute local false discovery rate (lfdr) score and append it into the table as `lfdr` column.

```
lfdr_results <- compute_lfdr_result(test_results)
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr

## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr

## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr

## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr

## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr

## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr

## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr

## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr

## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr

## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0

## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr

## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
```

```
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
```

```
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
```

```
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
```

```
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

```
## Step 1... determine cutoff point
## Step 2... estimate parameters of null distribution and eta0
```

```
## Step 3... compute p-values and estimate empirical PDF/CDF
## Step 4... compute q-values and local fdr
```

Making Summary Tables and Plots

The **MDstatsDIAMS** can compute the numbers of precursor peptides that were found significant between two conditions by the table row `Rejected = TRUE`. The `p.value` or an alternative significance score such as `lfdr` can be supplied for determining the numbers of precursor peptides with significant changes in mean log-quantity.

```
tables <- compute_contingency_tables(
  lfdr_results, alpha = 0.05, q_value_column = 'lfdr'
)

print(tables)
```

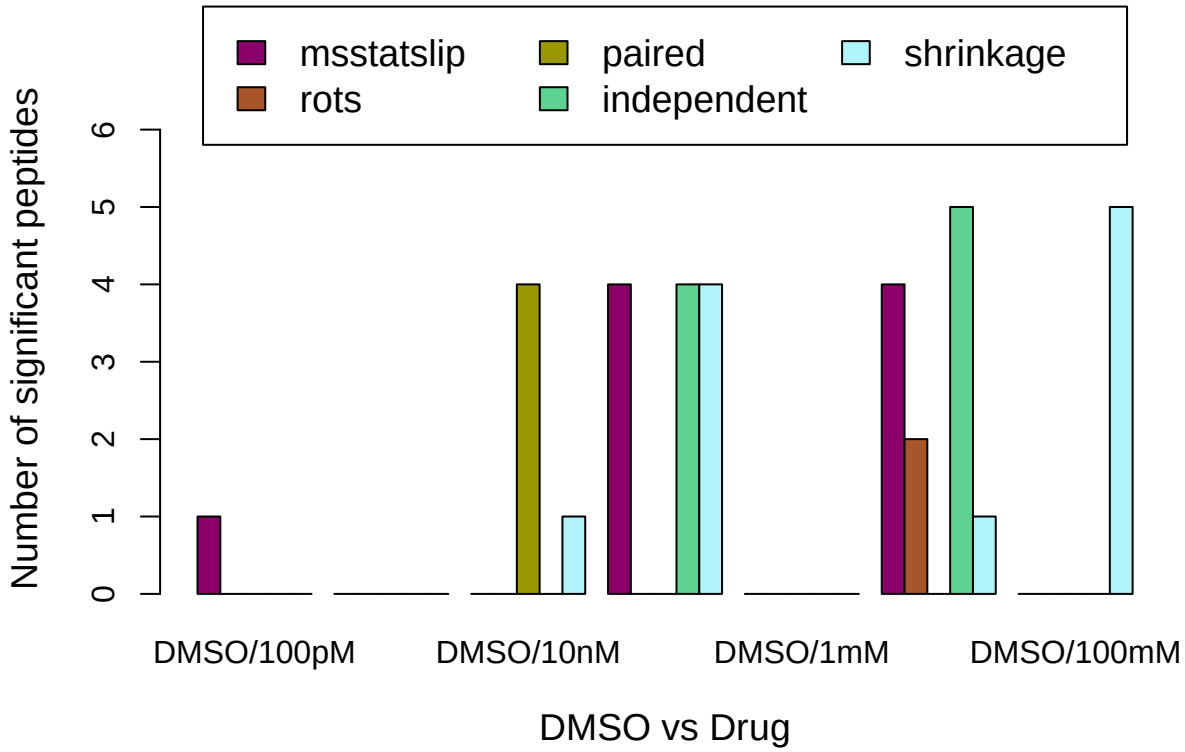
```
## $`DMSO/100mM`
##   Rejected msstatslip rots paired independent shrinkage
## 1    FALSE         21   27     23          21        12
## 2     TRUE          0    0      0           0         5
##
## $`DMSO/100nM`
##   Rejected msstatslip rots paired independent shrinkage
## 1    FALSE         17   23     24          17        13
## 2     TRUE          4    0      0           4         4
##
## $`DMSO/100pM`
##   Rejected msstatslip rots paired independent shrinkage
## 1    FALSE         19   25     23          20        15
## 2     TRUE          1    0      0           0         0
##
## $`DMSO/10mM`
##   Rejected msstatslip rots paired independent shrinkage
## 1    FALSE         16   24     24          15        15
## 2     TRUE          4    2      0           5         1
##
## $`DMSO/10nM`
##   Rejected msstatslip rots paired independent shrinkage
## 1    FALSE         21   25     19          21        15
## 2     TRUE          0    0      4           0         1
##
## $`DMSO/1mM`
##   Rejected msstatslip rots paired independent shrinkage
## 1    FALSE         20   25     23          20        16
## 2     TRUE          0    0      0           0         0
##
## $`DMSO/1nM`
##   Rejected msstatslip rots paired independent shrinkage
## 1    FALSE         18   24     21          18        12
## 2     TRUE          0    0      0           0         0
```

The **MDstatsMDIAMS** provides a function for drawing bar plots of the numbers of precursor peptides of significant changes across comparisons at a given significance level. With `rejected = TRUE`, it draws numbers of significant precursor peptides.

```
n_tables <- length(tables)

# Sort comparisons by drug doses from low to high
if (n_tables == 3) {
  table_conds <- substr(names(tables), 6, 10)
  drug_dose <- 10 ** as.numeric(gsub("CON", "", table_conds)) / 1000
  ordered <- order(drug_dose)
} else {
  table_conds <- substr(names(tables), 6, 10)
  drug_dose <- as.numeric(
    gsub("100pM", "0.1", gsub("nM", "", gsub("mM", "000", table_conds))))
  ordered <- order(drug_dose)
}

bar_plot_contingency_tables(
  tables = tables[ordered], rejected = TRUE, scale_factor = 1,
  ylab = "Number of significant peptides",
  ylim = c(0, n_sampled_proteins * 1.2 + 3),
  xlab = "DMSO vs Drug",
  cex.lab = 1.2,
  add_legend = TRUE,
  legend_ncol = 3,
  legend_cex = 1.2
)
```



The **MDstatsMDIAMS** can draw line plots for the numbers of precursor peptides of significant changes for every statistical methods. By `rejected = TRUE`, it draws numbers of significant precursor peptides.

```
line_plot_contingency_tables(
  x = drug_dose[ordered], tables = tables[ordered], rejected = TRUE,
```

```
  xlab = "DMSO vs Drug (nM)",
  scale_factor = 1, log = "x",
  ylab = "Number of significant peptides", ylim = c(0, n_sampled_proteins * 1.2 + 3),
  cex.lab = 1.2, cex = 2, lwd = 2.3,
  add_legend = TRUE, legend_coord = "topleft", legend_cex = 1.2
)
```