MSstatsShiny: A Multipurpose UI for Reproducible Analysis of Quantitative Proteomic Experiments

Devon Kohler[1], Maanasa Kaza[1], Cristina Pasi[2], Dhaval Mohandas[3], Mateusz Staniak[4], Ting Huang[1], Meena Choi[5], Eduard Sabido[6] , Olga Vitek[1]

[1]Northeastern University, Boston, MA; [2]Universitat Oberta de Catalunya, Barcelona, Spain; [3]EXL Service, New York, NY; [4]University of Wrocław, Wrocław, Poland; [5]Genentech Inc, South San Francisco, CA; [6]Centre for Genomic Regulation, Barcelona, Spain

Introduction

Quantitative mass spectrometry proteomic experiments require robust analytic and modeling techniques to ensure the results are correctly interpreted. There are a variety of different tools that can assist researchers in analysis, however they are generally only suited for one type of experimental design and are usually implemented in coding packages. To address these challenges, we have created the UI MSstats-Shiny, an R-Shiny based UI integrated with the R packages Msstats, MSstatsTMT, and MSstatsPTM providing all users an end-to-end pipeline that can analyze a variety of experimental designs.

1. Design and Implementation

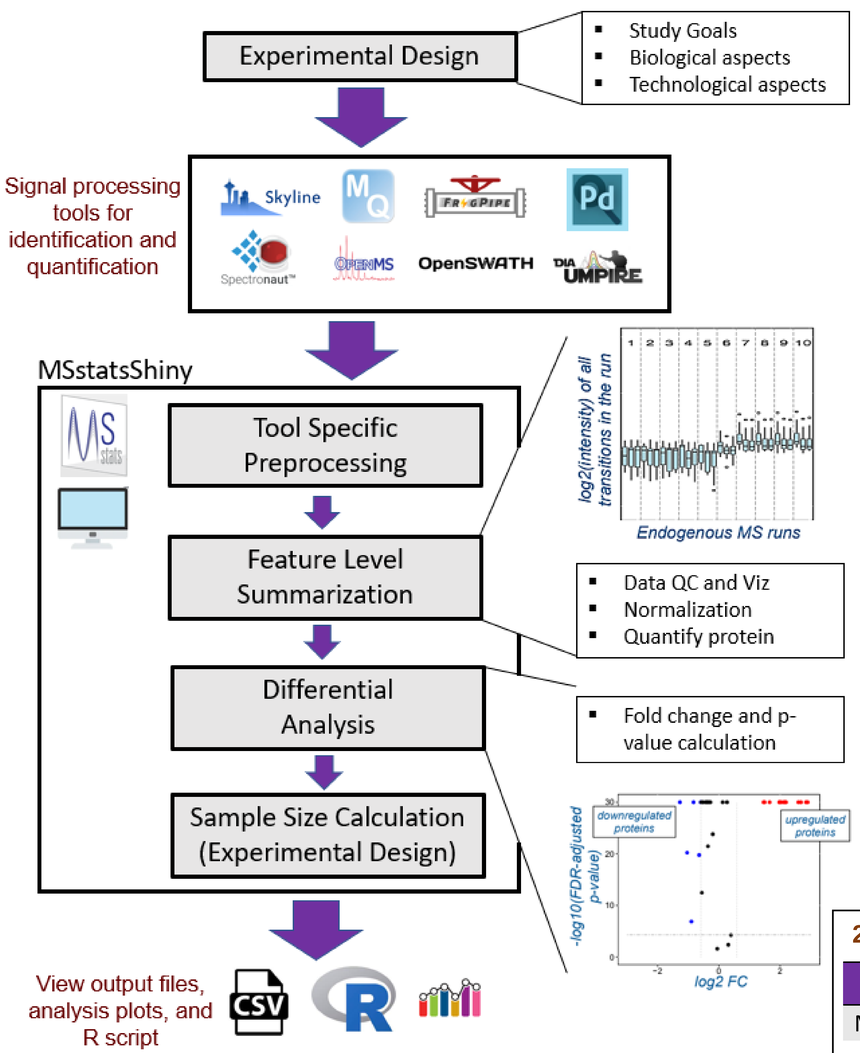

Figure 1: MSstatsShiny is integrated directly with the MSstats family of packages. The analysis steps include tool specific data preprocessing, missing value imputation, data summarization, and differential analysis with mixed-effects linear regression models.

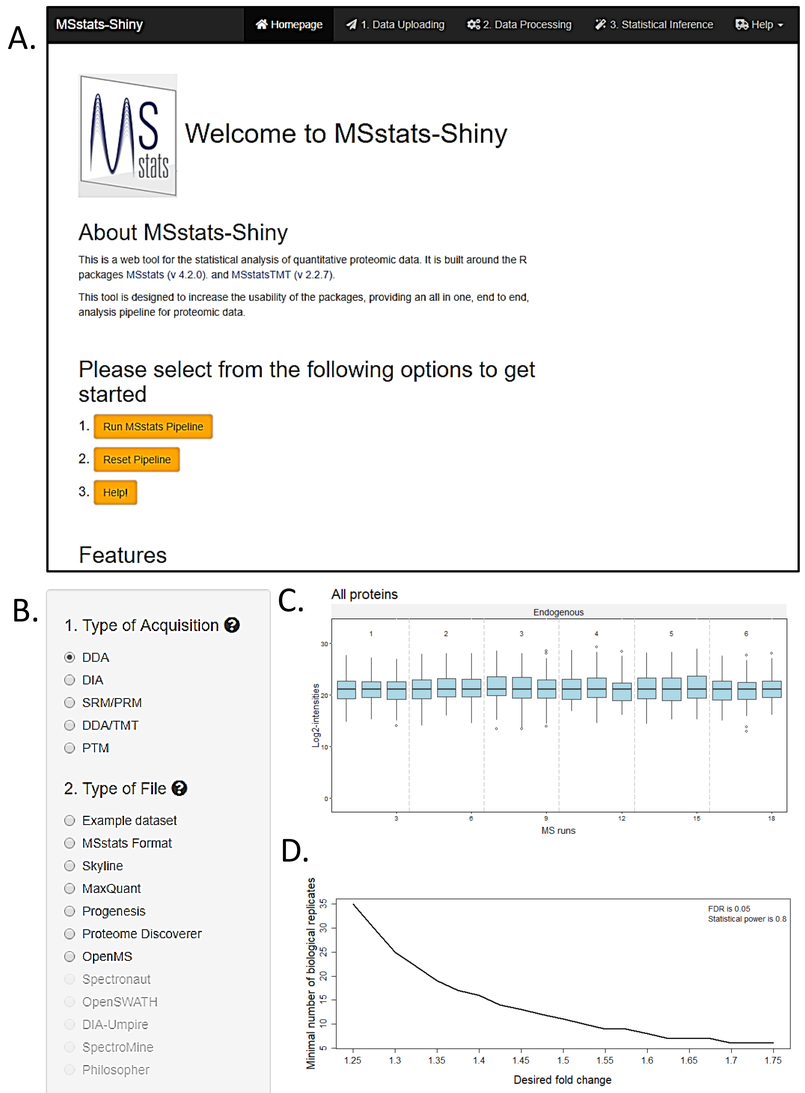

Figure 2: MSstatsShiny UI. (A) Homepage. (B) Tool specific processing. (C) QC plot in data summarization. (D) Sample size calculation.

2. Application can be used to analyze many different experiments

Study	Acquisition	Proteins	Upload	Summarization	Model	Ram	MSV ID
Masculins et al. (UI)	TMT	10043	15 sec	15 min 52 sec	9 min 12 sec	5 GB	000088966
Stark et al. (UI)	DIA	280	12 min	54 min 31 sec	1 min 24 sec	25 GB	000086623
Stark et al. (code)	DIA	280	3 min 8 sec	52 min 22 sec	1 min 11 sec	19 GB	000086623

Table 1: Performance measurements for case studies using MSstatsShiny. Analyses were run on a machine with an Intel i7-7700k @ 4.2GHz and 32 GB of RAM.

3. Code generation enables scalable and reproducible analysis

In large experiments the RAM usage and processing time of the analysis can be unsustainable in Shiny. To remedy these situations, MSstatsShiny can generate code that the user can use to analyze their experiment.

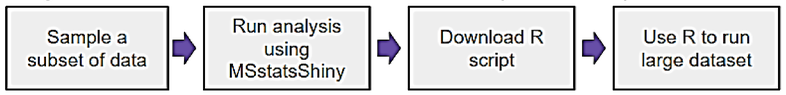

Figure 3: An example workflow to generate R code to analyze large experimental files

```
library(MSstats)

# Read data
data <- read.csv("Enter filepath here")

# use MSstats for protein summarization
summarized <- MSstats:::dataProcess(data,
                                    normalization = 'equalizeMedians',
                                    logTrans = 2,
                                    nameStandards = NULL,
                                    featureSubset = 'topN',
                                    n_top_feature = 3,
                                    summaryMethod="TMP",
                                    censoredInt='0',
                                    MBimpute=TRUE,
                                    remove50missing=FALSE,
                                    maxQuantileforCensored=0.999)
dataProcessPlots(data=summarized,
                                type="Enter ProfilePlot or QCPlot Here",
                                ylimUp = F,
                                ylimDown = F,
                                which.Protein = "Enter Protein to Plot Here",
                                summaryPlot = TRUE,
                                address = FALSE)

# Create the contrast matrix
contrast.matrix <- NULL
comparison <- matrix(c(1, -1, 0, 0, 0, 0),nrow=1)
contrast.matrix <- rbind(contrast.matrix, comparison)
comparison <- matrix(c(1, 0, -1, 0, 0, 0),nrow=1)
contrast.matrix <- rbind(contrast.matrix, comparison)
row.names(contrast.matrix)<-c("C1 vs C2","C1 vs C3")
colnames(contrast.matrix)<-c("C1","C2","C3","C4","C5","C6")

# Model-based comparison
model <- MSstats:::groupComparison(contrast.matrix, summarized)
groupComparisonPlots(data=model$ComparisonResult,
                                type="Enter VolcanoPlot, Heatmap, or ComparisonPlot",
                                which.Comparison="all",
                                which.Protein="all",
                                address="")
```

Figure 4: Example script generated via MSstatsShiny. The parameters selected by the user are automatically converted into R code.

4. Conclusion

MSstatsShiny is a Shiny-based UI that provides the proteomics community with a data analysis option that is general, reproducible, and scalable. The platform is open source and is available for anyone to use.

5. References

1. Stark, K., Goncharov, T., Varfolomeev, E. et al. Genetic inactivation of RIP1 kinase activity in rats protects against ischemic brain injury. Cell Death Dis 12, 379 (2021). https://doi.org/10.1038/s41419-021-03651-6
2. Maculins T. et al. Multiplexed proteomics of autophagy-deficient murine macrophages reveals enhanced antimicrobial immunity via the oxidative stress response. eLife 10:e62320 (2021). https://doi.org/10.7554/eLife.62320

Code Availability

1. https://github.com/Vitek-Lab/MSstats-Shiny/