SUPPORTING INFORMATION

Target-decoy Based False Discovery Rate Estimation for Large-scale Metabolite Identification

Xusheng Wang[1,&,*], Drew R Jones[2, &,5], Timothy I Shaw [1,3], Ji-Hoon Cho[1], Yuanyuan Wang[2], Haiyan Tan[1], Boer Xie[2], Suiping Zhou[1], Yuxin Li[1,2], and Junmin Peng[1,2,4,*]

[1]St. Jude Proteomics Facility, [2]Department of Structural Biology, [3]Department of Computational Biology, [4]Department of Developmental Neurobiology, St. Jude Children's Research Hospital, 262 Danny Thomas Place, Memphis, TN 38105, USA

[5]Current address: Department of Biochemistry and Molecular Pharmacology, New York University School of Medicine, 550 1st Avenue, New York, NY 10016, USA

[*]Corresponding authors: Xusheng Wang, email: xusheng.wang@stjude.org, and Junmin Peng, email: junmin.peng@stjude.org

Tel: 901-595-7499, Fax: 901-595-3032

TABLE OF CONTENTS

Supplementary Figures

Figure S1. Evaluation of the target-decoy strategy with additional simulation analyses. (a) Evaluation with three different decoy-generating strategies (i.e. +5H, +7H, and +9H). (b) Evaluation with four different mass shifts of precursor ions (i.e. +3.5 Da, +5.5 Da, +7.5 Da, and +9.5 Da).

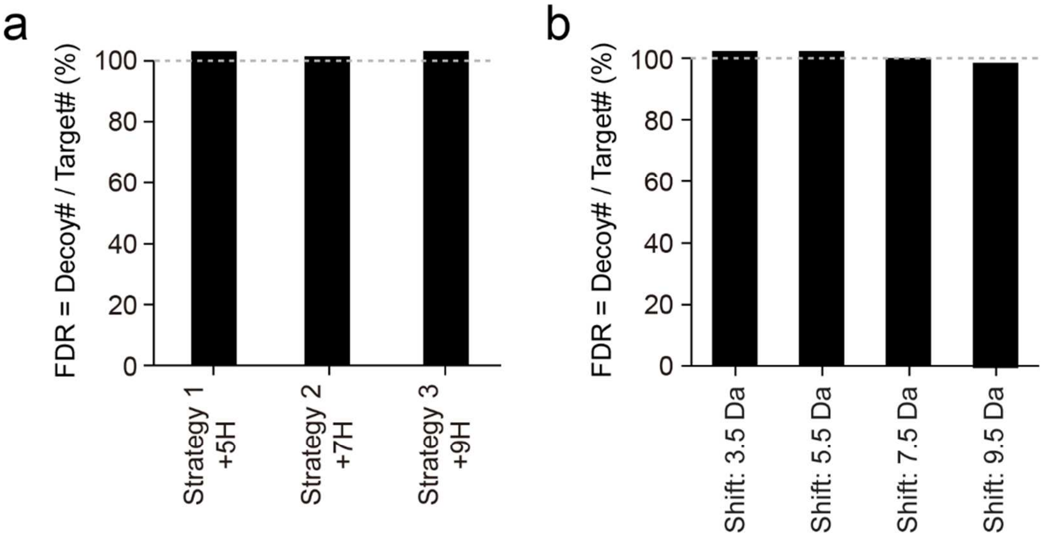
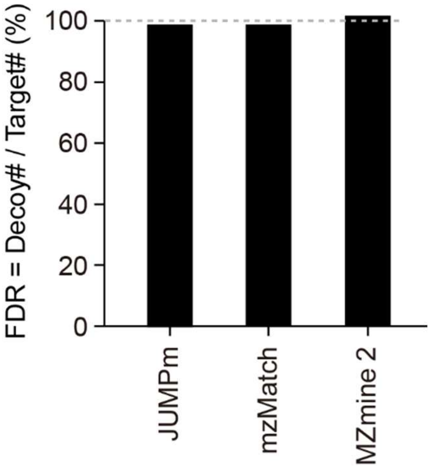


Figure S2. Evaluation of the target-decoy strategy with JUMPm, mzMatch, and MZmine 2.



Supplementary Text

Introduction of JUMPm software tool

JUMPm is a computer program for large-scale metabolite identification of LC-MS/MS analyses. The program is capable of processing stable isotope labeled and unlabeled MS-based metabolomic data. JUMPm uses MS raw data as input and then performs deisotoping, noise characterization, and mass calibration to generate a list of metabolite features. For stable-isotope labeled data, JUMPm determines the formula's labeled elements and stoichiometry. Differentially labeled metabolites are observed in pairing groups of co-eluting ions. These ion pairs are detected by a Pairing score algorithm (Pscore), which uses isotopic mass defects of C and N labels, relative ion intensity, and co-elution of the ion pairs. Finally, unique formulas are identified by the stoichiometry of isotope-labeled elements (C and N) and accurate precursor ion mass. For unlabeled data, metabolite formulas are identified by accurate precursor mass.

Once the metabolite formulas are identified, JUMPm searches the associated MS/MS spectra against a user-defined structure database (e.g. YMDB, HMDB, or PubChem) to detect structure candidates and rank the candidates by Matching scores (Mscore). The Mscore for each structure candidate is calculated by comparing theoretical (*in silico*) MS/MS fragment ions with the observed MS/MS peaks. Finally, JUMPm outputs the best candidate for each formula in a table, consisting of MS scan information, formula, structure, and matching score.

Detailed procedures for testing mzMatch

(1) Install mzMatch program in RStudio (version 1.0.143) using the following commands.

```
source("http://bioconductor.org/biocLite.R")
biocLite(c("xcms", "multtest", "mzR"))

install.packages(c("rJava", "R.utils","XML", "snow", "caTools", "bitops", "ptw",
"gplots", "tcltk2"))
source ("http://puma.ibls.gla.ac.uk/mzmatch.R/install_mzmatch.R")
install.packages("http://puma.ibls.gla.ac.uk/mzmatch.R/mzmatch.R.tar.gz")
```

(2) Construct an HMDB decoy database based on the HMDB target database provided by mzMatch tool. We wrote an in-house Perl script to facilitate this process.

(3) Search the same LC-MS/MS raw data used for JUMPm against the constructed target-decoy database. We first extracted peaks with centwave algorithm from XCMS:

```
xseto <- xcmsSet(sampleList$filenames, method='centWave', ppm=2,
peakwidth=c(10,100), snthresh=10, prefilter=c(3,1000),
integrate=1, mzdiff=0.01, verbose.columns=TRUE, fitgauss=FALSE)
```

(4) Export the detected signals to the PeakML file format:

```
PeakML.xcms.write.SingleMeasurement(xset=xseto,
filename=sampleList$outputfilenames, ionisation="detect", ppm=5,
addscans=0, ApodisationFilter=TRUE, nSlaves=4)
```

(5) Change the precursor ion mass by adding 4.5 Da:

```
raw_data<-PeakML.Read("peakml/12C_RP_Pos_1_c.peakml")
raw_data$peakDataMtx[,1]<-raw_data$peakDataMtx[,1]+4.5
raw_data$peakDataMtx[,2]<-raw_data$peakDataMtx[,2]+4.5
raw_data$peakDataMtx[,3]<-raw_data$peakDataMtx[,3]+4.5
for (i in 1: length(raw_data$chromDataList[])){raw_data$chromDataList[[i]][1,1]<-
raw_data$chromDataList[[i]][1,1]+4.5}
for (i in 1: length(raw_data$chromDataList[])){raw_data$chromDataList[[i]][1,2]<-
raw_data$chromDataList[[i]][1,2]+4.5}
for (i in 1: length(raw_data$chromDataList[])){raw_data$chromDataList[[i]][1,3]<-
raw_data$chromDataList[[i]][1,3]+4.5}

PeakML.Write (raw_data,outFileName=sampleList$outputfilenames)
```

(6) Combine and related peaks using the following command:

```
INPUTDIR <- "combined_RTcorr"
FILESf <- dir (INPUTDIR,full.names=TRUE,pattern="\\.peakml$")
mzmatch.ipeak.Combine(i=paste(FILESf,collapse=","), v=T, rtwindow=60,
 o="final_RT_corr_combined.peakml", combination="set", ppm= 5)
```

```
mzmatch.ipeak.sort.RelatedPeaks(i="final_RT_corr_combined.peakml", v=T,
o="final_combined_related.peakml",

basepeaks="final_combined_basepeaks.peakml", ppm=5, rtwindow=30)
```

(7) Identify peaks from the composite target-decoy database:

```
mzmatch.ipeak.util.Identify(i="final_combined_related.peakml", v=T,
o="final_combined_related_identified.peakml", ppm=100, databases=DBS)
```

where DBS is the constructed target-decoy database.

(8) Convert results in peakml format into text file:

```
mzmatch.ipeak.convert.ConvertToText (

i="final_combined_related_identified.peakml",
o= "final_combined_related_identified_processed.txt", databases=DBS,
annotations="identification,ppm,adduct,relation.ship")
```

Detailed procedures for testing MZmine 2

(1) Download MZmine 2 (version 2.31) from the website (https://github.com/mzmine/mzmine2/releases).
(2) Use the same set of simulated LC-MS/MS raw data for JUMPm and mzMatch.
(3) Construct an HMDB target-decoy database, which is the same database used for mzMatch.
(4) Extract peaks and generate features using the function of Chromatogram build, provided by MZmine 2.
(5) Export features in XML format, and change each mass by adding 4.5 Da. After changing the mass of features, import the modified XML file.
(6) Perform Custom database search against the composite target-decoy database with *m/z* tolerance of 100 ppm.