Supplemental Table 1

Blue text shows scripts used to replicate statistical techniques, followed by command line instructions for parameters.

```
xmsPANDA.R
###############################################################################################
#load xmsPANDA
suppressPackageStartupMessages(library(xmsPANDA))
library(getopt)

###Input parameters###########
spec = matrix(c(
  'feature', 'f', 1, 'character',
  'class', 'c', 1, 'character',
  'output', 'o', 1, 'character',
  'selection', 's', 1, 'character',
  'pvalue', 'p', 1, 'double',
  'fdr', 'r', 1, 'double',
  'vip', 'v', 1, 'double',
  'anova', 'a', 0, 'logical'), byrow=TRUE, ncol=4)
opt = getopt(spec)

#feature table input
if (is.null(opt$feature)) {print("needs feature table"); quit();} else {feature_table_file <- normalizePath(opt$feature, mustWork=TRUE)}

#class label input
if (is.null(opt$class)) {print("needs class labels"); quit();} else {class_labels_file <- normalizePath(opt$class, mustWork=TRUE)}

#output location
if (is.null(opt$output)) {outloc <- getwd()} else {outloc <- opt$output}
if(substr(outloc,1,1) != "/") {outloc <- paste(getwd(), "/", outloc, sep="")}
dir.create(outloc,showWarnings=FALSE)

#feature selection method
if (is.null(opt$selection)) {featsel <- "limma"} else {featsel <- opt$selection}

#p-value
if (is.null(opt$pvalue)) {pvalue <- 0.05} else {pvalue <- opt$pvalue}

#fdr value
if (is.null(opt$fdr)) {fdr <- 0.05} else {fdr <- opt$fdr}

#vip-value
if (is.null(opt$vip)) {vip <- 2} else {vip <- opt$vip}

if (is.null(opt$anova)) {
#start: see manual for additional arguments and description
demetabs_res<-diffexp(
	#1) arguments for input files
	feature_table_file=feature_table_file,
	parentoutput_dir=outloc,
	class_labels_file=class_labels_file,
	input.intensity.scale="raw",

	##2) data preprocessing order: 1) summarization, 2) filtering by missing values, 3) imputation; 4) transformation and normalization
	#options for normalization methods: log2quantilenorm, log2transform, znormtransform, lowess_norm, quantile_norm, rangescaling,
paretoscaling, mstus, eigenms_norm,
	#vsn_norm, sva_norm, none, tic_norm, cubicspline_norm, mad_norm
	num_replicates = 1,
	summarize.replicates =TRUE, summary.method="median",summary.na.replacement="knn",
	rep.max.missing.thresh=0.3,
	all.missing.thresh=0.1, group.missing.thresh=0.8, missing.val=0,
	rsd.filt.list = c(1),
	normalization.method="log2transform",

	##3) options for feature selection: "limma","ttest","wilcox","lm1wayanova","lmreg","pls",
	#"pamr","spls","pls","o1pls","MARS","RF","rfesvm","logitreg", "poissonreg",
	#"ttestrepeat","wilcoxrepeat", "lm1wayanovarepeat","limma1wayrepeat","spls1wayrepeat"
	#"lm2wayanova","lm2wayanovarepeat","limma2way","limma2wayrepeat","spls2wayrepeat"
	pairedanalysis = FALSE, featselmethod=c(featsel),
	pvalue.thresh=pvalue,
```

```
        fdrthresh = fdr, fdrmethod="BH",
        kfold=5,networktype="complete",
        analysismode="classification",pls_vip_thresh = vip,
        num_nodes = 8,
        foldchangethresh=0,
        optselect=TRUE,max_comp_sel=3,saveRda=FALSE,pls.permut.count=NA,
        pca.ellipse=TRUE,
        aggregation.method="none",

        #4) arguments for centrality analysis, WGCNA and global clustering analysis (HCA and EM clustering)
        differential.network.analysis=FALSE, wgcnarsdthresh=1,WGCNAmodules=FALSE,globalclustering=FALSE,

        #5) arguments for correlation and network analysis using the selected features
        cor.method="spearman", abs.cor.thresh = 0.4, cor.fdrthresh=0.2,

        #6) arguments for graphical options: see manual for additional arguments
        output.device.type="png",pca.cex.val=4,legendlocation="bottomleft",
        net_node_colors=c("green","red"),
        net_legend=FALSE,aggregation.max.iter=100,
        heatmap.col.opt="redblue",manhattanplot.col.opt=c("darkblue","red3"),
        color.palette=c("journal"),hca_type="two-way",cex.plots=0.6,
        lineplot.lty.option=c("dotted", "solid", "dashed", "dotdash", "longdash", "twodash"),
        timeseries.lineplots=FALSE,lme.modeltype="RI",ylab_text="Intensity",boxplot.type="ggplot",
        multiple.figures.perpanel = FALSE,add.jitter=FALSE,add.pvalues=FALSE,ggplot.type1=TRUE,
        hca.labRow.value = TRUE, hca.labCol.value = FALSE,hca.cex.legend=0.5,
        limma.contrasts.type=c("contr.sum"),
        plot.boxplots.raw=FALSE,vcovHC.type="HC3"
    )
} else {
  demetabs_res<-diffexp(
        #1) arguments for input files
        feature_table_file=feature_table_file,
        parentoutput_dir=outloc,
        class_labels_file=class_labels_file,
        input.intensity.scale="raw",

        ##2) data preprocessing order: 1) summarization, 2) filtering by missing values, 3) imputation; 4) transformation and normalization
        num_replicates = 1,
        summarize.replicates =TRUE, summary.method="median",summary.na.replacement="halffeaturemin",
        rep.max.missing.thresh=0.3,
        all.missing.thresh=0.5, group.missing.thresh=0.8, missing.val=0,
        log2transform = FALSE, medcenter=FALSE, znormtransform = FALSE,
        quantile_norm = FALSE, lowess_norm = FALSE, madscaling = FALSE,
        rsd.filt.list = c(1),

        ##3) arguments for feature seletion:
        pairedanalysis = TRUE, featselmethod=featsel,pvalue.thresh=pvalue,
        fdrthresh = fdr, fdrmethod="BH",
        kfold=5,networktype="complete",
        samplermindex=NA,numtrees=5000,analysismode="classification",pls_vip_thresh = vip, num_nodes = 10,
        max_varsel = 5, pls_ncomp = 5,pred.eval.method="BER",rocfeatlist=seq(2,10,1),
        rocfeatincrement=TRUE,
        rocclassifier="svm",foldchangethresh=0,
        optselect=TRUE,max_comp_sel=5,saveRda=FALSE,pls.permut.count=100,
        pca.ellipse=TRUE,ellipse.conf.level=0.95,svm.acc.tolerance=5,pamr.threshold.select.max=FALSE,
        aggregation.method="RankAggreg",mars.gcv.thresh=10,

        #4) arguments for WGCNA and global clustering analysis (HCA and EM clustering)
        wgcnarsdthresh=30,WGCNAmodules=FALSE,globalclustering=FALSE,

        #5) arguments for correlation and network analysis using the selected features
        cor.method="spearman", abs.cor.thresh = 0.4, cor.fdrthresh=0.2,
        globalcor=TRUE,target.metab.file=NA,
        target.mzmatch.diff=10,target.rtmatch.diff=NA,max.cor.num=NA,

        #6) arguments for graphical options: see manual for additional arguments
        output.device.type="png",pca.cex.val=4,legendlocation="bottomleft",
        net_node_colors=c("green","red"),
        net_legend=FALSE,heatmap.col.opt="RdBu",sample.col.opt="rainbow"
    )
}
sink(file=NULL)
#########################################################################################
```

```
Rscript xmsPANDA.R -f features.txt -c class.file.txt -s limma -p 0.05 -r 1 -o output.folder
```

```
xmwas.R
##############################################################################################
#load package
library(xMWAS)
library(getopt)

###Input parameters###########
spec = matrix(c(
  'metab', 'm', 1, 'character',
  'factor', 'f', 1, 'character',
  'factorz', 'z', 1, 'character',
  'class', 'c', 1, 'character',
  'output', 'o', 1, 'character',
  'threshold', 't', 1, 'double',
  'numcomp', 'n', 1, 'integer',
  'missing', 'a', 1, 'integer'
  ), byrow=TRUE, ncol=4)
opt = getopt(spec)

#metabolomics table input
if (is.null(opt$metab)) {metab<-NA} else {metab <- read.csv(normalizePath(opt$metab, mustWork=TRUE), sep="\t")}
if (!is.na(metab)) {rownames(metab) <- make.names(metab[,1])}

#factor table input
if (is.null(opt$factor)) {factor<-NA} else {factor <- read.csv(normalizePath(opt$factor, mustWork=TRUE), sep="\t")}
if (!is.na(factor)) {rownames(factor) <- make.names(factor[,1])}

#factor Z table input
if (is.null(opt$factorz)) {factorz<-NA} else {factorz <- read.csv(normalizePath(opt$factorz, mustWork=TRUE), sep="\t")}
if (!is.na(factorz)) {rownames(factorz) <- make.names(factorz[,1])}

#class label input
if (is.null(opt$class)) {class.label.file<-NA} else {
  class.label <- read.csv(normalizePath(opt$class, mustWork=TRUE), sep="\t");
  colnames(class.label)<-c("SampleID","Class");
}

#output location
if (is.null(opt$output)) {output <- getwd()} else {output <- opt$output}
if(substr(output,1,1) != "/") {output <- paste(getwd(), "/", output, sep="")}
dir.create(output,showWarnings=FALSE)

#number of components
if (is.null(opt$numcomp)) {numcomp <- 2} else {numcomp <- opt$numcomp}

#correlation threshold
if (is.null(opt$threshold)) {threshold<-0.4} else {threshold <- opt$threshold}

#missing threshold
if (is.null(opt$missing)) {missing<-0} else {missing <- opt$missing}

xMat<-data.matrix(metab)
yMat<-data.matrix(factor)
zMat<-data.matrix(factorz)
wMat<-NA

#Please see user manual for description of arguments:
#https://github.com/kuppal2/xMWAS/blob/master/example_manual_tutorial/xMWAS-manual.pdf

#call the run_xmwas() function:
xmwas_res<-run_xmwas(Xome_data=xMat,Yome_data=yMat,Zome_data=zMat,Wome_data=NA,outloc=output,
classlabels=class.label,class_fname=NA,xmwasmethod="pls",plsmode="regression",
max_xvar=10000, #e.g. select top 10000 of the variabels in X dataset based on relative standard deviation; change according to your
dataset; you can also use proportion such as round(nrow(xMat)*0.3) to select top 30% of the variables.
max_yvar=10000, #select top 10000 of the variabels in Y dataset based on relative standard deviation;  change according to your dataset;
you can also use proportion such as round(nrow(yMat)*0.3) to select top 30% of the variables.
max_zvar=10000, #select top 10000 variabels in Z dataset based on relative standard deviation;  change according to your dataset; you can
also use proportion such as round(nrow(zMat)*0.3) to select top 30% of the variables.
max_wvar=10000, #select top 10000 variabels in W dataset based on relative standard deviation;  change according to your dataset; you
can also use proportion such as round(nrow(wMat)*0.3) to select top 30% of the variables.
rsd.filt.thresh=1,
corthresh=threshold, #absolute correlation threshold
keepX=1000, #select up to top 1000 variables in the sPLS model; change according to your dataset
```

```
keepY=1000, #select up to top 1000 variables in the sPLS model; change according to your dataset
keepZ=1000, #select up to top 1000 variables in the sPLS model; change according to your dataset
keepW=1000, #select up to top 1000 variables in the sPLS model; change according to your dataset
pairedanalysis=FALSE, #set to TRUE if repeated measures study design
optselect=FALSE, #perform optimal PLS componenet selection; TRUE or FALSE; set to FALSE for exact Pearson correlation calculation using PLS regression
rawPthresh=0.05, #p-value threshold for correlation based on Student's t-test
numcomps=numcomp, #max number of PLS components to use; set to N-1 (N: number of samples) for exact Pearson correlation calculation using PLS regression
net_edge_colors=c("blue","red"),
net_node_colors=c("orange", "green","cyan","pink"),
Xname="AminoAcidMetabolites", #change the name of dataset X
Yname="Measures", #change the name of dataset Y
Zname="NonAminoAcidMetabolites", #change the name of dataset Z
Wname="W", #change the name of dataset W
net_node_shape=c("square","circle","triangle","star"),
all.missing.thresh=0, #filter based on missing values: set to NA to turn it OFF; otherwise specify a value between: 0 to 1 (e.g. 0.8 to require that at least 80% of the samples have a non-missing value)
missing.val=missing,
seednum=100,label.cex=0.2,vertex.size=6,
interactive=FALSE,max_connections=NA,
centrality_method="eigenvector", #centrality evaluation method
use.X.reference=FALSE,removeRda=TRUE,
compare.classes=FALSE, #compare classes: TRUE or FALSE
class.comparison.allvar=TRUE,
modularity.weighted=TRUE,
globalcomparison=TRUE,
plot.pairwise=FALSE, #plot results for pairwise comparisons: TRUE or FALSE
apply.sparse.class.comparison=FALSE, #perform variable selection in sPLS during class-wise comparison (default: FALSE)
layout.type="fr1"
)

suppressWarnings(try(sink(file=NULL),silent=TRUE))
#################################################################################################
```

Rscript xmwas.R -m features.of.interest.txt -f comparison.features.txt -c class.file.txt -o output.folder -t 0.3

Extra information on these packages can be via their github repositories at kuppal2/xMWAS and kuppal2/xmsPANDA.