Comparing differential abundance (DA) analysis methods for microbiome count data

true

Compiled: 27 July, 2023

Contents

Load libraries and functions

Set working directory

```
workingDirectory <- getwd()
```

Load libraries

```
library(pacman)
library(devtools)
library(SparseDOSSA2)
library(pkgmaker)
library(optparse)
library(parallel)
library(stringi)
library(doParallel)
library(edgeR)
library(zinbwave)
library(SummarizedExperiment)
library(dplyr)
library(readr)
library(tibble)
library(BiocParallel)
library(DESeq2)
library(phyloseq)
library(ROCR)
```

```
library(plyr)
library(ggplot2)
library(ape)
library(miceadds) #source.all
library(ggpubr)
library(grid)
```

Load simulation functions

These .R functions are included in the folders "\simulation_performance" and "\sim_fun".

```
Rsfolder_pathPerf <- "..\\simulation_performance"
source.all(Rsfolder_pathPerf, grepstring="\\.R",  print.source=TRUE )
```

```
## *** source run_performance_simulations.R
## *** source summarizing_performance.R
```

```
Rsfolder_pathSim <- ".\\sim_func"
source.all(Rsfolder_pathSim, grepstring="\\.R",  print.source=TRUE )
```

```
## *** source clean_data_sim.R
## *** source generateMetadata.R
## *** source newsparseDOSSA_Wrapper.R
## *** source newtrigger_sparseDOSSA_Simulator.R
## *** source run_DESeq2.R
## *** source run_DESeq2Zinbwave.R
## *** source run_edgeR.R
## *** source run_edgeRZinbwave.R
## *** source run_limmaVOOM.R
## *** source run_limmaVOOMZinbwave.R
## *** source run_MaAsLin2.R
## *** source run_simulator_SparseDOSSA2.R
## *** source utilityFunctions.fa_230322_new.R
```

Simulations

Simulation parameters

Collect the values set for simulation senarios. The number of microbes (nMicrobes) and read deapth are set from the template dataset. A range of total sample sizes from (ns) in the two groups or experimental conditions and effect sizes or log-fold changes.

```
nMicrobes <- 303
readDepth<- 1883

es=c(0.5,1,2)   #effect sizes
ns=c(10,20,50,100,200) #total samlple sizes
nIterations = 100  #number of simulations
```

Calculate performance measures of DA methods

This includes calculating false discovery rates (FDR), power (sensitivity), specificity, AUC, MCC and F1-scores for several differential abundance methods: Deseq2, edgeR and limma-voom, and their ZINBWaVE weighted counterparts Deseq2-ZINBWaVE, edgeR-ZINBWaVE, limma-voom-ZINBWaVE. We also included MaAsLin2. Because many of these techniques treat taxa with group-wise structured zeros differently in differential

abundance testing and to ensure a fair comparison of these techniques, taxa without group-wise structured zeros are considered in our simulation-based comparisons. The following function reads the simulated data from the folder 'Input' and save the resulting simulation performance measures in the folder "Output".

```
output_performance <- run_performance_simulations(es, ns, nIterations, nMicrobes,readDepth)
#simulation results saved in the folder "output"
```

Summarize the various measures of performance by averaging over the simulations

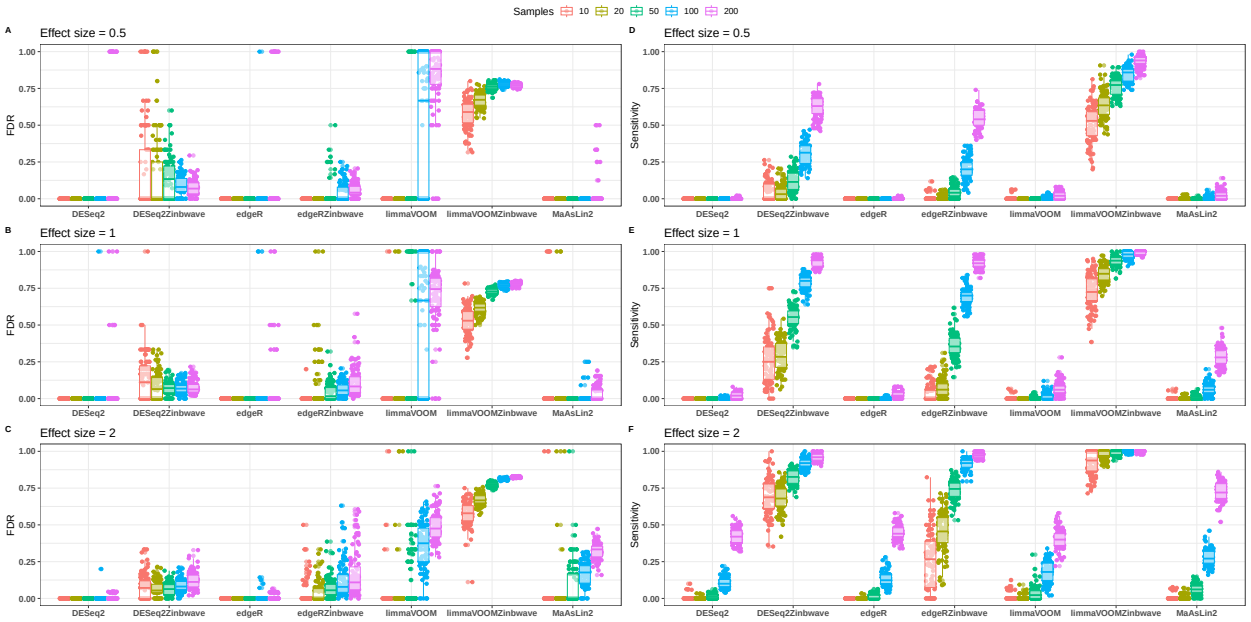
```
methodName <- c('DESeq2','edgeR',
                'limmaVOOM',
                'limmaVOOMZinbwave',
                'DESeq2Zinbwave',
                'edgeRZinbwave',
                'MaAsLin2')

rResultsAll <- summarizing_performance(nMicrobes,readDepth, nIterations, methodName,ns,es)


rResults1All <- rResultsAll[which(rResultsAll$methodName %in%
                  c('DESeq2','DESeq2Zinbwave','edgeR','edgeRZinbwave'
                    ,'limmaVOOM','limmaVOOMZinbwave', 'MaAsLin2')),]

rResults1All$nSubjects=factor(rResults1All$nSubjects,
                              levels=c("10","20","50","100","200"))

names(rResults1All)[names(rResults1All)=="nSubjects"] <- "Samples"
```

Plot FDR and power(sensitivity) across effect and sample sizes.

```
#FDR
rResults1All2=subset(rResults1All,effectSize==0.5)
e1=ggplot(rResults1All2, aes(methodName , FDR, color=Samples)) +
  geom_point(position = position_jitterdodge()) +
  geom_boxplot(alpha=0.6)+
  theme_bw()+
  theme(text = element_text(size = 12), axis.text = element_text(face="bold")) +
  xlab("Methods") + ylab("FDR") +
  ggtitle("Effect size = 0.5")


rResults1All5=subset(rResults1All,effectSize==1)
e2=ggplot(rResults1All5, aes(methodName , FDR, color=Samples)) +
  geom_point(position = position_jitterdodge()) +
  geom_boxplot(alpha=0.6)+
  theme_bw()+
  theme(text = element_text(size = 12), axis.text = element_text(face="bold")) +
  xlab("Methods") + ylab("FDR") +
  ggtitle("Effect size = 1")

rResults1All10=subset(rResults1All,effectSize==2)
e5=ggplot(rResults1All10, aes(methodName , FDR, color=Samples)) +
  geom_point(position = position_jitterdodge()) +
```

```
  geom_boxplot(alpha=0.6)+
  theme_bw()+
  theme(text = element_text(size = 12), axis.text = element_text(face="bold")) +
  xlab("Methods") + ylab("FDR") +
  ggtitle("Effect size = 2")



#Sensitivity
rResults1All2=subset(rResults1All,effectSize==0.5)
s1=ggplot(rResults1All2, aes(methodName , Sensitivity, color=Samples)) +
  geom_point(position = position_jitterdodge()) +
  geom_boxplot(alpha=0.6)+
  theme_bw()+
  theme(text = element_text(size = 12), axis.text = element_text(face="bold")) +
  xlab("Methods") + ylab("Sensitivity")+
  ggtitle("Effect size = 0.5")

rResults1All5=subset(rResults1All,effectSize==1)
s2=ggplot(rResults1All5, aes(methodName , Sensitivity, color=Samples)) +
  geom_point(position = position_jitterdodge()) +
  geom_boxplot(alpha=0.6)+
  theme_bw()+
  theme(text = element_text(size = 12), axis.text = element_text(face="bold")) +
  xlab("Methods") + ylab("Sensitivity")+
  ggtitle("Effect size = 1")

rResults1All10=subset(rResults1All,effectSize==2)
s5=ggplot(rResults1All10, aes(methodName , Sensitivity, color=Samples)) +
  geom_point(position = position_jitterdodge()) +
  geom_boxplot(alpha=0.6)+
  theme_bw()+
  theme(text = element_text(size = 12), axis.text = element_text(face="bold")) +
  xlab("Methods") + ylab("Sensitivity")+
  ggtitle("Effect size = 2")



performanceSim=ggarrange( e1 + rremove("xlab"), s1 + rremove("xlab"),
                          e2 + rremove("xlab"), s2 + rremove("xlab"),
                          e5 + rremove("xlab"), s5 + rremove("xlab"),
                          common.legend = TRUE,
                          labels = c("A", "D", "B", "E", "C", "F"),  #NULL
                          ncol = 2, nrow =3,
                          align = "hv",
                          font.label = list(size = 10,
                                            color = "black", face = "bold",
                                            family = NULL, position = "top")
)
#annotate_figure(fredSim1, bottom = textGrob("Methods", gp = gpar(cex = 1.3)))
```

```
performanceSim
```




```
#save plot
# ggsave(filename = "C:\\Users\\fenta\\Documents\\OneDrive\\Documents\\2021\\Amsterdam\\papers\\Differe
#        width=18, height=10)
# performanceSim
# dev.off()
```

Session info

```
#sessionInfo()
```