1 Metabolite Identification

Given MS (Mass Spectrum) data, we can easily perform metabolite identification using the MetPC package. To use the MetPC package, three variables (‘Name’, ‘Area’, and ‘Spectra’) should be included in the data. Suppose that we have two data sets: sample and library.

```r
sdata <- read.csv("Standard_1.csv", header=T)
ldata <- read.csv("Standard_2.csv", header=T)
sdata <- data.frame(Name=sdata$Name, Area=sdata$Area, Spectra=sdata$Spectra)
ldata <- data.frame(Name=ldata$Name, Area=ldata$Area, Spectra=ldata$Spectra)
```

The following figure shows part of the raw data, which were obtained from two dimensional GCxGC/TOF-MS.

![Part of raw data from two dimensional GCxGC/TOF-MS](Figure 1)

Prior to the metabolite identification, peak merging is done by choosing the peak with the largest area.

```r
# Peak merging
psdata <- pmerge(sdata)
pldata <- pmerge(ldata)
```

After peak merging, we calculate some statistics, which are inputs for our statistical hierarchical model. That is, the dissimilarity score (S), and the competition scores ($b_j$ and $b_j^*\$) are calculated by using the `cal_ob` function.

```r
# Calculate competition/dissimilarity score
obdata <- cal_ob(psdata$Spectra, pldata$Spectra)
```
1.1 Metabolite identification: MetID

We perform identification by calling the MetID function, which requires some input such as dissimilarity score, competition score and initial parameters.

```r
# Metabolite identification
MetID(500, pldata, 0.8, obdata, muT=2, muF=15, muF2=50, sigmaT=3, sigmaF=15, sigmaF2=50)
```

The first argument 500 is the number of iterations of the EM algorithm and 0.8 is used as the cutoff of the confidence measure. `pldata` is library data after peak merging and `obdata` is score calculated by using sample and library data. The last 6 arguments are initial values of mean and variance in three component normal mixture model. After performing the process of identification, we get the results below:

| Name                | Confidence |
|---------------------|------------|
| Octadecane          | 0.9999996  |
| Eicosane            | 0.9994737  |
| Undecane            | 0.9994674  |
| Decane              | 0.9994534  |
| 2-cyclohexen-1-one, 3,5,5-trimethyl- | 0.9994470 |
| Phenol, 3-methyl-   | 0.9994220  |
| Hexadecane          | 0.9994173  |
| Heptadecane         | 0.9994067  |
| Phenol, 4-chloro-3-methyl- | 0.9994010 |
| Nonane              | 0.9993744  |
| Benzenamine, N-phenyl- | 0.9993550 |
| Aniline             | 0.9993534  |
| Phenol, 2,4-dichloro- | 0.9993491 |
| Phenol              | 0.9993478  |
| Dimethyl phthalate  | 0.9993340  |
| Pentadecane         | 0.9993321  |
| Hexacosane          | 0.9993309  |
| Dibenzofuran        | 0.9993276  |
| Dodecane            | 0.9993159  |
| Benzyl Alcohol      | 0.9993111  |

Figure 2: Result of metabolite identification

Figure 2 provides the top 20 identification results in terms of confidence measure. All lists are provided in Supplementary Materials II.

1.2 Parameter estimation by the EM-algorithm

For parameter estimation, we assume that score density belongs to the two- or three-component normal mixture. Thus, we implemented three functions: `estpar_tf, estpar_ttf, estpar_tff`. In case of two-component normal mixture, `estpar_tf` is used. For three-component normal mixture, we consider two different scenarios. That is, two component normal mixture can be considered for true score density or false score density depending on the situation. `estpar_tff` is used when the distribution of true score is a mixture model while `estpar_tff` is used when the distribution of false score is a mixture model. As a quick check for the density estimation, we suggest to compare it with the kernel density estimator.

For illustration purpose, we here used `estpar_tff` to estimate the parameters and considered 500 iterations of the EM algorithm. The following code generate Figure 3, which provides trace plots for four parameter estimates selected.

```r
pars <- estpar_tff(500, obdata, muT=2, muF1=15, muF2=50, sigmaT=3, sigmaF1=15, sigmaF2=50)
plotpars(pars, 500)
```
For graphical representation, we used the package ggplot2, which can be easily downloaded from the CRAN. Based on the plot, some performance measures can be checked, for example, convergence rate of the EM algorithm.

1.3 Kernel density estimator

Kernel density estimator (non-parametric version) is used for two different purposes. It is first used when deciding the type of normal mixture: two- or three-component normal mixture. Also, it can be used to check the accuracy of parameter estimates by looking at the overlap of two density estimates.

Two types of density estimates are included in Figure 4. It seems that normal mixture density estimate overlaps the kernel density estimate very well.

2 Biomarker Discovery

The discovery of biomarker metabolites is done under the control of two dimensional local false discovery rate (2d-fdr), which was implemented by Ploner et al. (2006). For biomarker discovery, another data set is considered. The data is pre-processed before it is used, i.e., log-transformation and standardization.
Figure 4: Non-parametric vs Parametric density estimate. Black line is kernel density estimate and red line is parametric density estimation.

```r
dat <- read.table("omija.txt", header=F)  
colnames(dat) <- rep(c("chinese","korean"), c(27,30))  
tdat <- dat[21:(nrow(dat)-20),]  # removing top and bottom  
ctdat <- log(1+tdat)  
for(i in 1:57){ ctdat[,i]=ctdat[,i]/sum(ctdat[,i]) } 

The following code shows how to conduct biomarker discovery by using the **fdr2d** function.

```r
# 2d-fdr  
fdr <- fdr2d(ctdat, colnames(ctdat), nperm=500)  
summary(fdr)
```

*nperm* is the number of permutations of group labels that is used for the estimation of 2d-fdr. Here, we considered 500 permutations. The following code shows how to generate two plots: tornado and volcano plot.

```r
Tornadoplot(fdr, main="Tornado plot", label=T, cex=2, cex.main=2, col="gray50", lcol="darkred")  
Volcanoplot(fdr,55, main="Volcano plot", label=T, cex=2, cex.main=2, col="gray50", lcol="darkred")
```

In the code above, we used 55 as the degree of freedom, which is used to set the color of the contour line. Figure 5 provides two plots: tornado and volcano plot. The only difference is y-axis: log(se) v.s. -log(p-value).

### 3 Software availability

The current version of bioinformatics tool is available at https://github.com/jjs3098/CNU-Bioinformatics-Lab. Furthermore, example data used in our paper are provided as well. The snapshot of the website is given in Figure 6.
Figure 5: The tornado plot (left) and the volcano plot (right)

Figure 6: Snapshot of github website