Meta-Workflow

Miao YU

2018-01-06

Contents

Preface

This is a book written in **Bookdown**. You could contribute it by a pull request in Github.

R and **Rstudio** are the softwares needed in this workflow.

The software package used for metabolomics data analysis is **xcms**.

Chapter 1

Introduction

Information in living organism commuicates along the Genomics, Transcriptomics, Proteomics and Metabolomics in Central dogma. Following such stream, we might answer certain problems in different scales from individual, population, community to ecosystem. Metabolomics (i.e., the profiling and quantitation of metabolites in body fluids) is a relatively new field of “omics” studies. Different from other omics studies, metabolomics always focused on small moleculars with much lower mass than polypeptide, around m/z 100-1000. Metabolomics studies are always performed in GC/MS, GC*GC/MS(Tian et al., 2016), LC/MS or NMR. This workflow concerns mass spectrum based metabolomics.

1.1 History

According to this book section(Kusonmano et al., 2016):

- 2000-1500 BC some traditional Chinese doctors who began to evaluate the glucose level in urine of diabetic patients using ants
- 300 BC ancient Egypt and Greece that traditionally determine the urine taste to diagnose human diseases
- 1913 Joseph John Thomson and Francis William Aston mass spectrometry
- 1946 Felix Bloch and Edward Purcell Nuclear magnetic resonance
- late 1960s chromatographic separation technique
- 1971 Pauling’s research team “Quantitative Analysis of Urine Vapor and Breath by Gas–Liquid Partition Chromatography”
- Willmitzer and his research team pioneer group in metabolomics which suggested the promotion of the metabolomics field and its potential applications from agriculture to medicine and other related areas in the biological sciences
- 2007 Human Metabolome Project consists of databases of approximately 2500 metabolites, 1200 drugs, and 3500 food components
- post-metabolomics era high-throughput analytical techniques

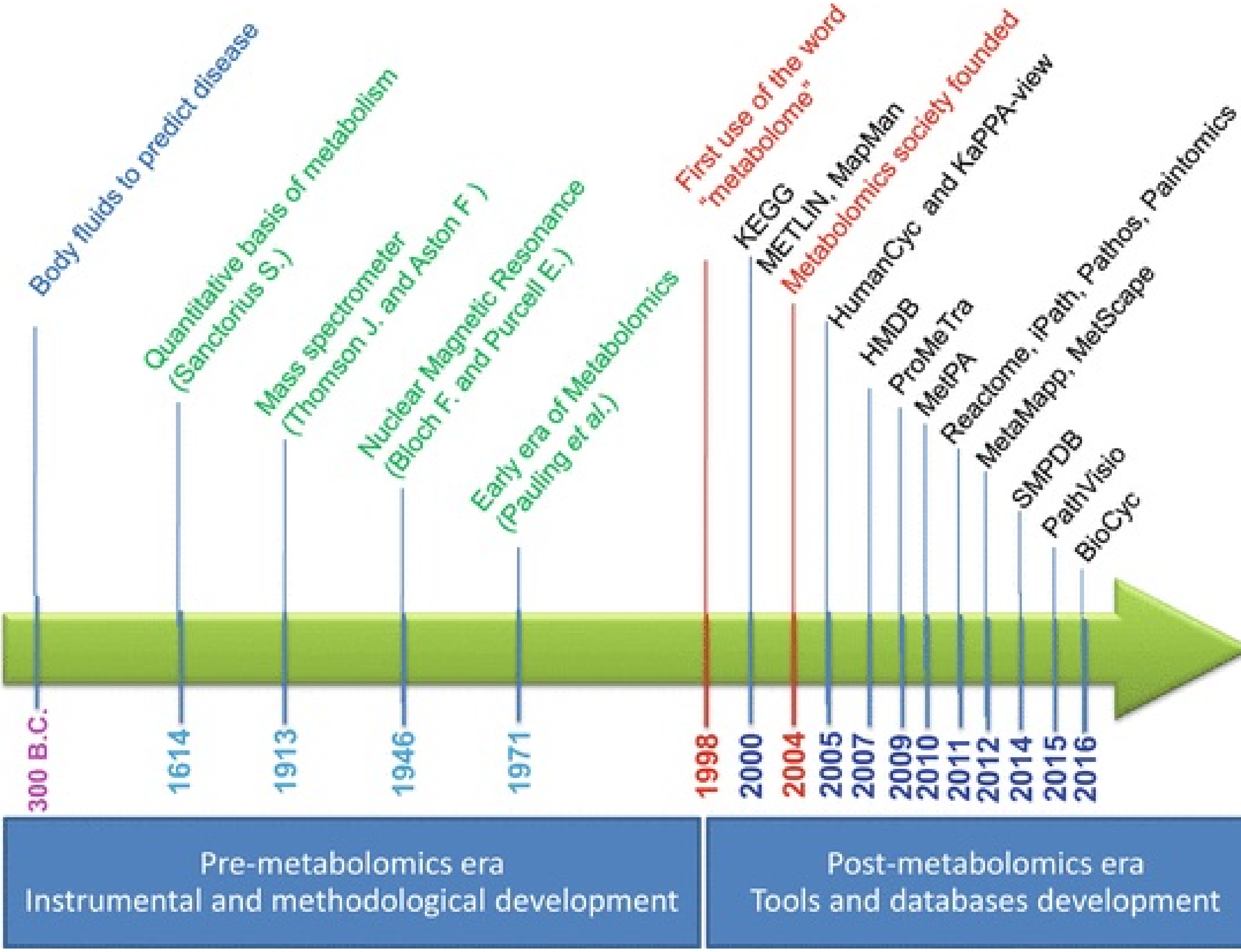

Body fluids to predict disease
Quantitative basis of metabolism
(Sanctorius S.)
Mass spectrometer
(Thomson J. and Aston F.)
Nuclear Magnetic Resonance
(Bloch F. and Purcell E.)
Early era of Metabolomics
(Pauling et al.)
First use of the word
"metabolome"
KEGG
METLIN, MapMan
Metabolomics society founded
HumanCyc and KaPPA-view
HMDB
ProMeTra
MetPA
Reactome, iPath, Pathos, Paintomics
MetaMapp, MetScape
SMPDB
PathVisio
BioCyc
300 B.C.
1614
1913
1946
1971
1998
2000
2004
2005
2007
2009
2010
2011
2012
2014
2015
2016
Pre-metabolomics era
Instrumental and methodological development
Post-metabolomics era
Tools and databases development

Figure 1.1: Metabolomics timeline during pre- and post-metabolomics era

1.2 Reviews and tutorials

Some new reviews and tutorials related to this workflow could be found in those papers(Alonso et al., 2015; Cajka and Fiehn, 2016; Lu and Xu, 2008; Schrimpe-Rutledge et al., 2016; Townsend et al., 2016; Barnes et al., 2016b,a).

Also I noticed more and more papers showed a bunch of data process methods as strategy for metabolomics(Watrous et al., 2017; Robbat Jr. et al., 2017). If you only need metabolomics as tools to tell your story, such stratrgy could be a quick start for you.

For software, check this review(Misra and van der Hooft, 2016).

1.3 Platform for metabolomics

1.3.1 XCMS online

XCMS online is hosted by Scripps Institute. If your datasets are not large, XCMS online would be the best option for you. Recently they updated the online version to support more functions for systems biology. They use metlin and iso metlin to annotate the MS/MS data. Pathway analysis is also supported. Besides, to accelerate the process, xcms online employed stream (windows only). You could use stream to connect your instrument workstation to their server and process the data along with the data acquisition automate. They also developed apps for xcms online, but I think apps for slack would be even cooler to control the data processing.

1.3.2 PRIMe

PRIMe is from RIKEN and UC Davis. It supports mzML and major MS vendor formats. They defined own file format ABF and eco-system for omics studies. The software are updated almost everyday. You could use MS-DIAL for untargeted analysis and MRMOROBS for targeted analysis. For annotation, they developed MS-FINDER and statistic tools with excel. This platform could replaced the dear software from company and well prepared for MS/MS data analysis and lipidomics. They are open source, work on Windows and also could run within mathmamtics. However, they don't cover pathway analysis. Another feature is they always show the most recently spectral records from public repositories. You could always get the updated MSP spectra files for your own data analysis.

If you make GC-MS based metabolomics, this paper(Matsuo et al., 2017) could be nice start.

1.3.3 OpenMS

OpenMS is another good platform for mass spectrum data analysis developed with C++. You could use them as plugin of KNIME. I suggest anyone who want to be a data scientist to get familiar with platform like KNIME because they supplied various API for different programme language, which is easy to use and show every steps for others. Also TOPPView in OpenMS could be the best software to visualize the MS data. You could always use the metabolomics workflow to train starter about details in data processing. pyOpenMS and OpenSWATH are also used in this platform. If you want to turn into industry, this platform fit you best because you might get a clear idea about solution and workflow.

1.3.4 MZmine 2

MZmine 2 has three version developed on Java platform and the lastest version is included into MSDK. Similar function could be found from MZmine 2 as shown in XCMS online. However, MZmine 2 do not have pathway analysis. You could use metaboanalyst for that purpose. Actually, you could go into MSDK to find similar function supplied by ProteoSuite and Openchrom. If you are a experienced coder for Java, you should start here.

1.3.5 XCMS

xcms is different from xcms online while they might share the same code. I used it almost every data to run local metabolomics data analysis. Recently, they will change their version to xcms 3 with major update for object class. Their data format would integrate into the MSnbase package and the parameters would be easy to set up for each step. Normally, I will use msconvert-IPO-xcms-xMSannotator-metaboanalyst as workflow to process the offline data. It could accelerate the process by parallel processing. However, if you are not familiar with R, you would better to choose some software above.

1.3.6 Emory MaHPIC

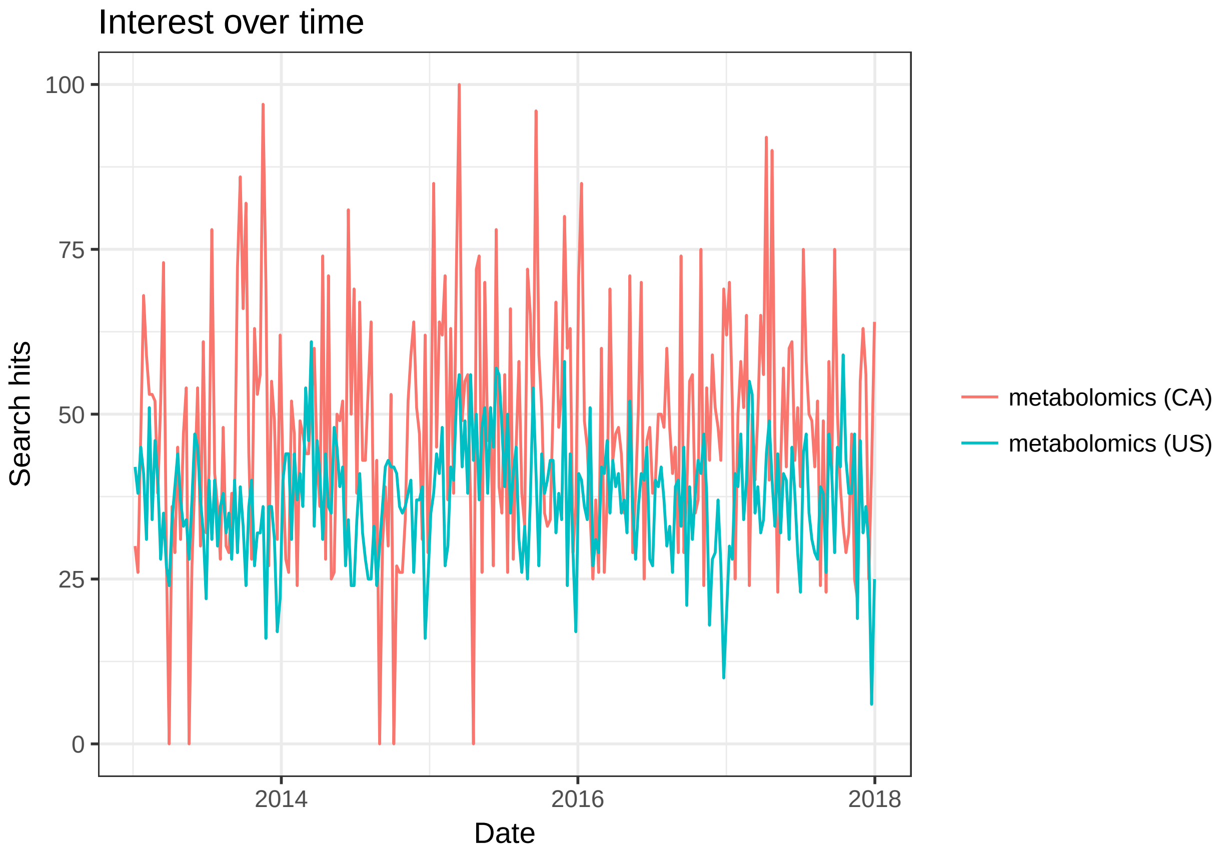
This platform is composed by several R packages from Emory University including apLCMS to collect the data, xMSanalyzer to handle automated pipeline for large-scale, non-targeted metabolomics data, xMSannotator for annotation of LC-MS data and Mummichog for pathway and network analysis for high-throughput metabolomics. This platform would be preferred by someone from environmental science to study exposome. I always use xMSannotator to annotate the LC-MS data.

1.3.7 Others

- MAVEN from Princeton University
- RAMclustR from Colorado State University
- MAIT based on xcms
- enviGCMS from me
- Metabolights for sharing data

1.4 Trends in Metabolomics

```r
library(gtrendsR)
res <- gtrends(c("metabolomics", "metabolomics"), geo = c("CA","US"))
plot(res)
```



1.4.1 Quantitative Metabolomics

Those papers (Kapoore and Vaidyanathan, 2016; Jorge et al., 2016).

1.4.2 High throughput Metabolomics

Those papers (Zampieri et al., 2017)

- Cohort size
- Temporal resolution
- Spatial resolution

1.5 Data sharing

1.5.1 Data hosting

See this paper(Haug et al., 2017):

- MetaboLights EU based
- The Metabolomics Workbench US based
- MetabolomeXchange search engine
- W4M(Guitton et al., 2017)

1.5.2 MS Database with annotation

1.5.2.1 MS/MS

- MassBank
- GNPS
- ReSpect: phytochemicals
- Metlin
- LipidBlast: in silico prediction
- MZcloud
- NIST: No free

1.5.2.2 MS

- Fiehn Lab
- NIST: No free

1.5.3 Compounds Database

- HMDB
- Lipid Maps
- KEGG
- PubChem
- Chemspider
- T3DB

1.5.4 Pathway Database

1.6 Workflow

Chapter 2

Exprimental design(DoE)

Before you perform any metabolomic studies, a clean and meaningful experimental design is the best start. You need at least two groups: treated group and control group. Also you could treat this group infomation as the one primary variable or primary variables to be explored for certain research purposes.

The numbers of samples in each group should be carefully calculated. Supposing the metaoblites of certain biological process only have a few metabolites, the first goal of the exprimenal design is to find the differences of each metabolite in different group. For each metabolite, such comparision could be treated as one t-test. You need to perform a Power analysis to get the numbers. For example, we have two groups of samples with 10 samples in each group. Then we set the power at 0.9, which means 1 minus Type II error probability, the standard deviation at 1 and the significance level(Type 1 error probability) at 0.05. Then we get the meanful delta between the two groups should be higher than 1.53367 under this experiment design. Also we could set the delta to get the minimized numbers of the samples in each group. To get those data such as the standard deviation or delta for power analysis, you need to perform pre-experiments.

```r
power.t.test(n=10,sd=1,sig.level = 0.05,power = 0.9)
```

```
##
##      Two-sample t test power calculation
##
##               n = 10
##           delta = 1.53367
##              sd = 1
##       sig.level = 0.05
##           power = 0.9
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```r
power.t.test(delta = 5,sd=1,sig.level = 0.05,power = 0.9)
```

```
##
##      Two-sample t test power calculation
##
##               n = 2.328877
##           delta = 5
##              sd = 1
##       sig.level = 0.05
##           power = 0.9
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

However, since sometimes we could not perform prelimintery experiment, we could directly compute the power based on false discovery rate control. If the power is lower than certain value, say 0.8, we just exclude this peak as significant features. Other study Blaise et al. (2016) show a method based on simulation to estimate the sample size. They usd BY correction to limit the influnces from correlationship. However, the nature of omics study make the power analysis hard to use one numbers and all the methods are trying to find a balance to represent more peaks with least samples(save money).

If there are other co-factors, a linear model or randomizing would be applied to eliminated their influences. You need to record the values of those co-factors for further data analysis. Common co-factors in metabolomic studies are age, gender, location, etc.

If you need data correction, some background or calibration samples are required. However, control samples could also be used for data correction in certain DoE.

Another important factors are instrumentals. High-resolution mass spectrum is always preferred. As shown in Lukas's study Najdekr et al. (2016):

> the most effective mass resolving powers for profiling analyses of metabolite rich biofluids on the Orbitrap Elite were around 60000–120000 fwhm to retrieve the highest amount of information. The region between 400–800 m/z was influenced the most by resolution.

However, elimination of peaks with high RSD% within group were always omited by most study. Based on pre-experiment, you could get a description of RSD% distribution and set cut-off to use stable peaks for further data analysis. To my knowledge, 50% is suitable considering the batch effects.

Chapter 3

Pretreatment

Pretreatment will affect the results of metabolomics. For example, feces collected with 95% ethanol or FOBT would be more reproducible and stable.

Dmitri et.al(Sitnikov et al., 2016) thought the most orthogonal methods to methanol-based precipitation were ion-exchange solid-phase extraction and liquid-liquid extraction using methyl-tertbutyl ether.

3.1 Quenching

Quenching solvent is always used to stop stop enzymatic activity.

In this review(Lu et al., 2017), authors said:

> A classical approach, which works well for many analytes, is boiling ethanol. Although the boiling solvent raises concerns about thermal degradation, it reliably denatures enzymes. In contrast, cold organic solvent may not fully denature enzymes or may do so too slowly such that some metabolic reactions continue, interconverting metabolites during the quenching process.

3.2 Extraction

According to this research(Bennett et al., 2009):

> The total metabolome concentration is approximately 300 mM, whereas the protein concentration is approximately 7 mM., which implies that most cellular metabolites are in free form.

- Tissue samples need to first be pulverized into fine powders

In this review(Lu et al., 2017), authors said:

> In our experience, for both cell and tissue specimens, 40:40:20 acetonitrile:methanol:water with 0.1 M formic acid (and subsequent neutralization with ammonium bicarbonate) is generally an effective solvent system for both quenching and extraction, including for ATP and other high-energy phosphorylated compounds. We typically use approximately 1 mL of solvent mix to extract 25 mg of biological specimen. …Thus, although drying is acceptable for most metabolites, care must be taken with redox-active species.

(Luo and Li, 2017) nano LC-MS could be used to analysis small numbers of cells

Chapter 4

Raw data pretreatment

Raw data from the instruments such as LC-MS or GC-MS were hard to be analyzed. To make it clear, the structure of those data could be summarised as:

- To get full infomation in the samples, full scan is perferred
- Full scan is performed synchronously with the seperation process

GC/LC-MS data are usually be shown as a matrix with column standing for retention times and row standing for masses. Noises are so much that such data could not be processed effeciently.

Conversation from the mass-retention time matrix into a vector with selected MS peaks at certain retention time is the basic idea of the Raw data pretreatment. The **Centwave** algorithm(Tautenhahn et al., 2008) based on detection of regions of interest(ROI) and the following Continuous Wavelet Transform (CWT) for the peaks is preferred for high-resolution mass spectrum.

With many groups of samples, you will get another data matrix with column standing for ions at cerntain retention time and row standing for samples after the Raw data pretreatment.

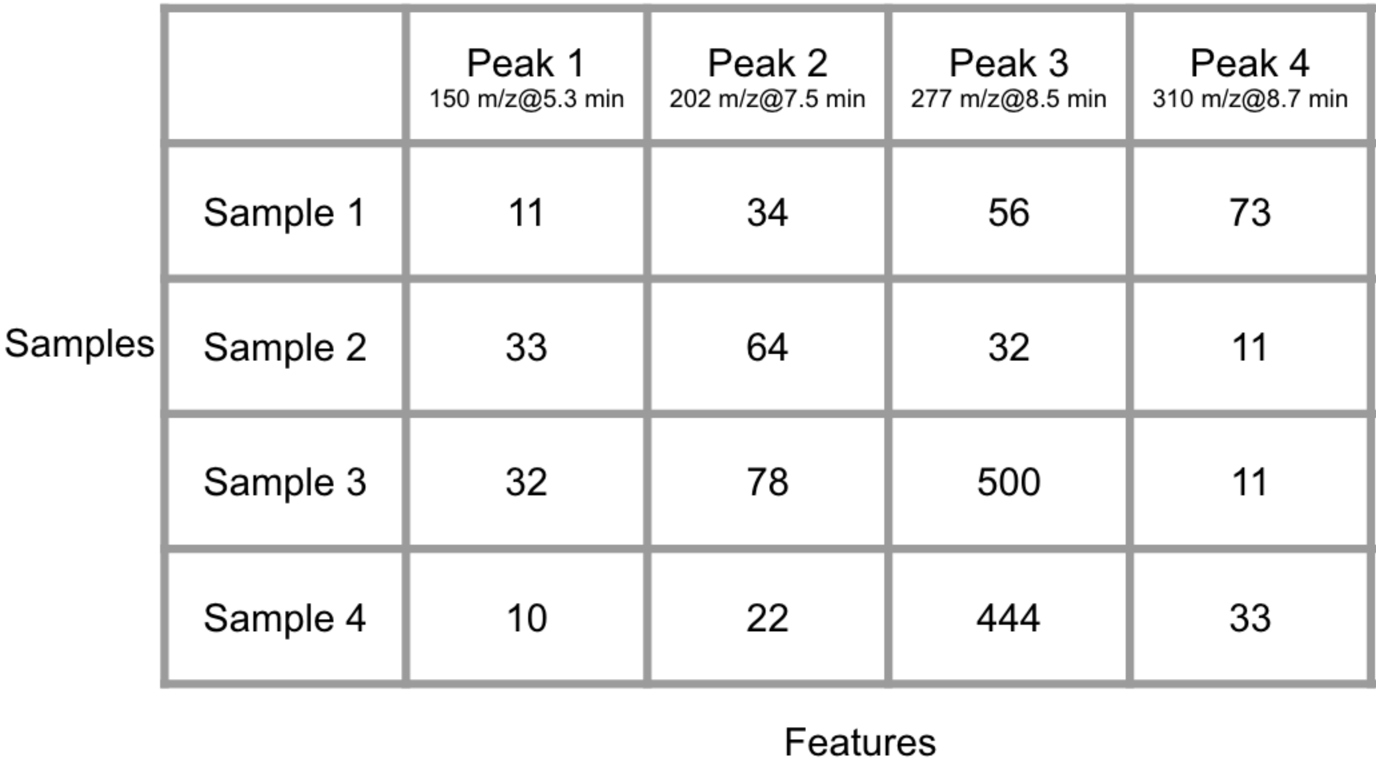
4.1 Spectral deconvolution

Without fracmental infomation about certain compound, the peak extraction would suffer influnces from other compounds. At the same retention time, co-elute compounds might share similar mass. Hard electron ionization methods such as electron impact ionization (EI), APPI suffer this issue. So it would be hard to distighuish the co-elute peaks' origin and deconvolution method(Du and Zeisel, 2013) could be used to seperate different groups according to the similar chromatogragh behiviors. Another computational tool **eRah** could be a better solution for the whole process(Domingo-Almenara et al., 2016). Also the **ADAD-GC3.0** could also be helpful for such issue(Ni et al., 2016).

4.2 Retention Time Correction

However, before you get the peaks, some corrections should be performed such as mass shift and retention time shift. The basic idea behind retention time correction is that use the high quality grouped peaks to make a new retention time. You might choose obiwarp or loess regression method to get the corrected retention time for all of the samples. Remember the original retention times might be changed and you might need cross-correct the data.

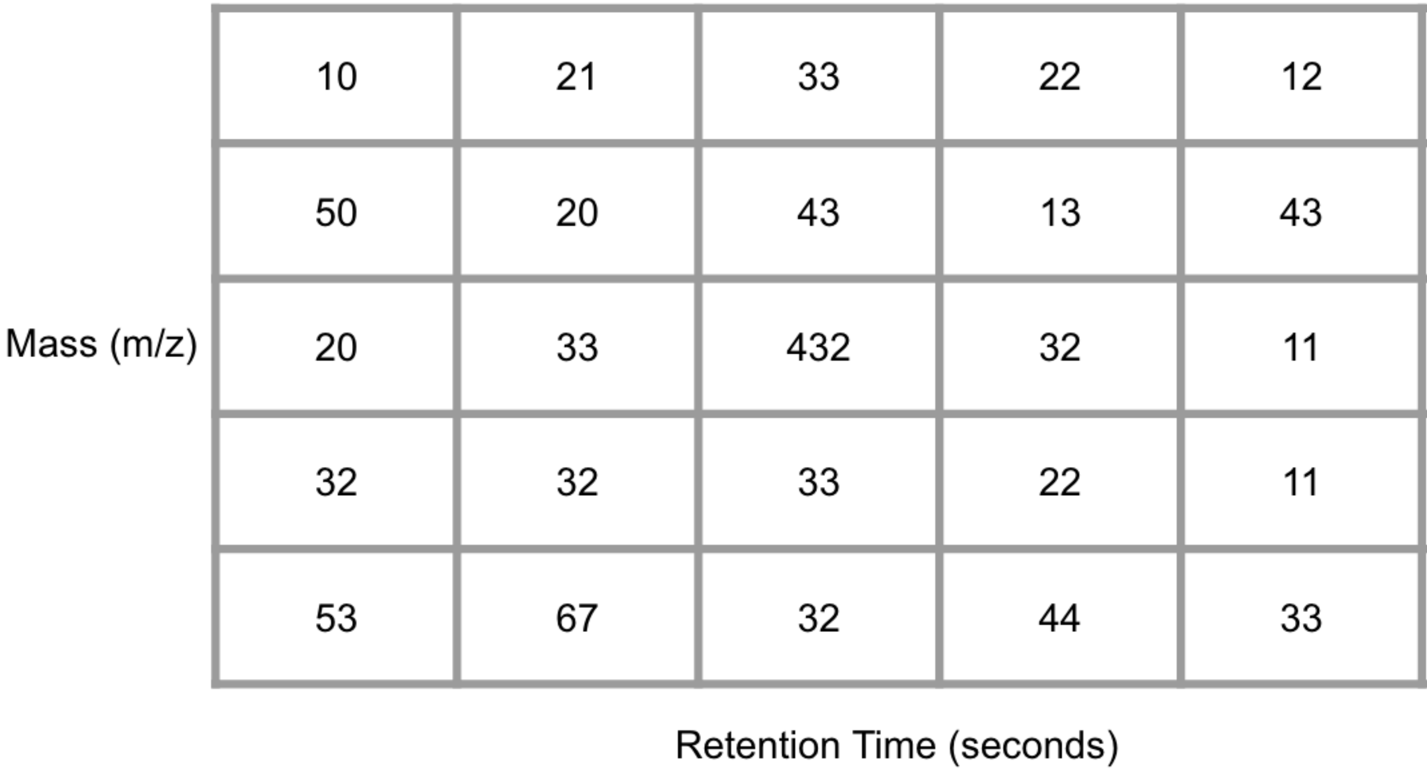
(Fu et al., 2017) show a matlab based shift correction methods



	Peak 1 150 m/z@5.3 min	Peak 2 202 m/z@7.5 min	Peak 3 277 m/z@8.5 min	Peak 4 310 m/z@8.7 min
Sample 1	11	34	56	73
Sample 2	33	64	32	11
Sample 3	32	78	500	11
Sample 4	10	22	444	33

Samples
Features

Figure 4.1: Demo of GC/LC-MS data



	10	21	33	22	12
	50	20	43	13	43
Mass (m/z)	20	33	432	32	11
	32	32	33	22	11
	53	67	32	44	33

Retention Time (seconds)

Figure 4.2: Demo of many GC/LC-MS data

4.3 Dynamic Range

Another issue is the Dynamic Range. For metabolomics, peaks could be below the detection limit or over the detection limit. Such Dynamic range issues might raise the loss of information.

4.3.1 Non-detects

Some of the data were limited by the detect of limitation. Thus we need some methods to impute the data if we don't want to lose information by deleting the NA or 0.

Two major imputation way could be used. The first way is use model-free method such as half the minimum of the values across the data, 0, 1, mean/median across the data(`enviGCMS` package could do this via `getimputation` function). The second way is use model-based method such as linear model, random forest, KNN, PCA. Try `simputation` package for various imputation methods.

Tobit regression is preferred for censored data. Also you might choose maximum likelihood estimation(Estimation of mean and standard deviation by MLE. Creating 10 complete samples. Pool the results from 10 individual analyses).

```
x <- rnorm(1000,1)
x[x<0] <- 0
y <- x*10+1
library(AER)
```

```
## Loading required package: car
## Loading required package: lmtest

## Loading required package: zoo

## Warning in FUN(X[[i]], ...): restarting interrupted promise evaluation

## 
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
## 
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Loading required package: survival
```

```
tfit <- tobit(y ~ x, left = 0)
summary(tfit)
```

```
## 
## Call:
## tobit(formula = y ~ x, left = 0)
## 
## Observations:
##          Total  Left-censored     Uncensored Right-censored 
##           1000              0           1000              0 
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)   1.0000     0.4370   2.288   0.0221 *  
## x            10.0000     0.3162  31.623   <2e-16 ***
## Log(scale)    2.1827     0.0000     Inf   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Scale: 8.87 
## 
## Gaussian distribution
## Number of Newton-Raphson Iterations: 1 
## Log-likelihood: -3102 on 3 Df
## Wald-statistic:  1000 on 1 Df, p-value: < 2.22e-16
```

4.3.2 Over Detection Limit

CorrectOverloadedPeaks could be used to correct the Peaks Exceeding the Detection Limit issue(Lisec et al., 2016).

4.4 RSD Filter

Some peaks need to be rule out due to high RSD%. See Exprimental design(DoE)

4.5 Power Analysis Filter

As shown in Exprimental design(DoE), the power analysis in metabolomics is ad-hoc since you don't know too much before you perform the experiment. However, we could perform power analysis after the experiment done. That is, we just rule out the peaks with a lower power in exsit Exprimental design.

4.6 Normalization

Variances among the samples across all the extracted peaks might be affected by factors other than the experiment design. To make the samples comparable, normailization across the samples are always needed. There are more than 20 methods to make normalization. We could devided those methods into two category: unsupervised and supervised.

Unsupervised methods only consider the normalization peaks intensity distribution across the samples. For example, quantile calibration try to make the intensity distribution among the samples similar. Such methods are preferred to explore the inner structures of the samples. Internal standards or pool QC samples also belong to this category. However, it's hard to take a few peaks standing for all peaks extracted.

Supervised methods will use the group information or batch information in experimental design to normalize the data. A linear model is always used to model the unwanted variances and remove them for further analysis.

Since the real batch effects are always unknown, it's hard to make validation for different normalization methods. Wu et.al preferred to make comparision between new methods and conventional methods(Wu and Li, 2016). Li et.al developed NOREVA to make comparision among 25 correction method(Li et al., 2017). Another idea is use spiked-in samples to validate the methods(Franceschi et al., 2012), which might be good for targeted analysis instead of non-targeted analysis.

Relative log abundance (RLA) plots(De Livera et al., 2012) and heatmap often used to show the variances among the samples.

(Thonusin et al., 2017) some methods for batch correction in excel

4.6.1 Unsupervised methods

4.6.1.1 Distribution of intensity

Intensity collects from LC/GC-MS always showed a right-skewed distribution. Log transformation is often necessary for further statistical analysis. In some case, a Log-transformated intensity could be visulizated easily.

4.6.1.2 Centering

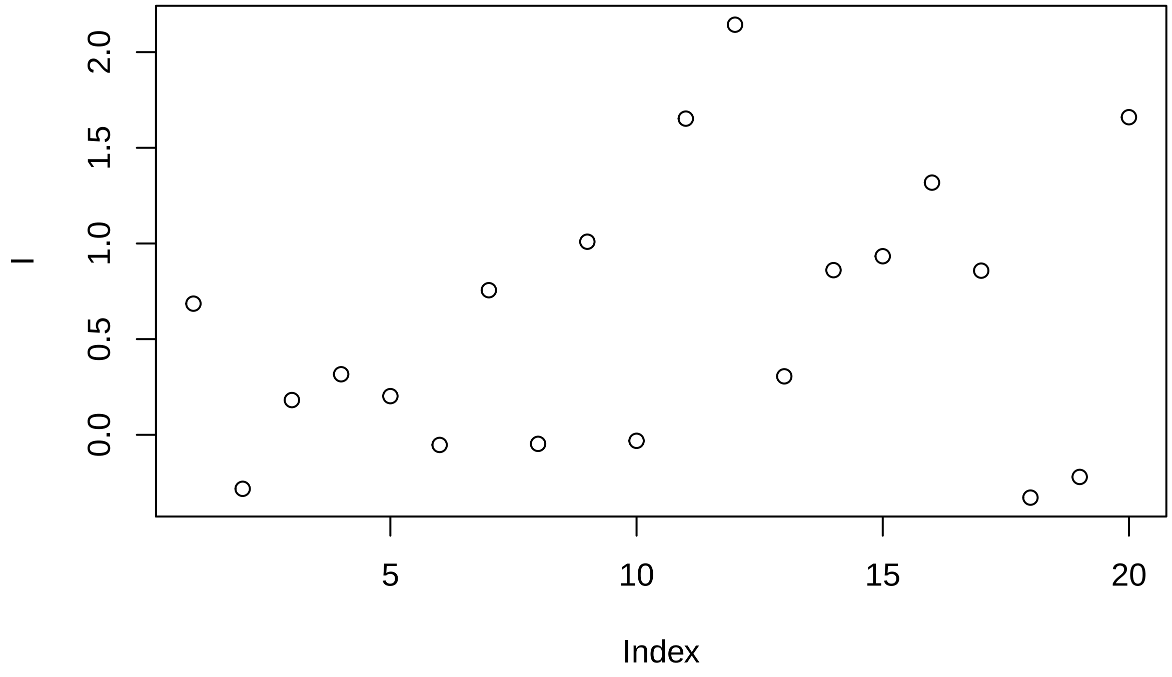
For peak p of sample s in batch b, the corrected abundance I is:

$$\hat{I}_{p,s,b} = I_{p,s,b} - mean(I_{p,b}) + median(I_{p,qc})$$

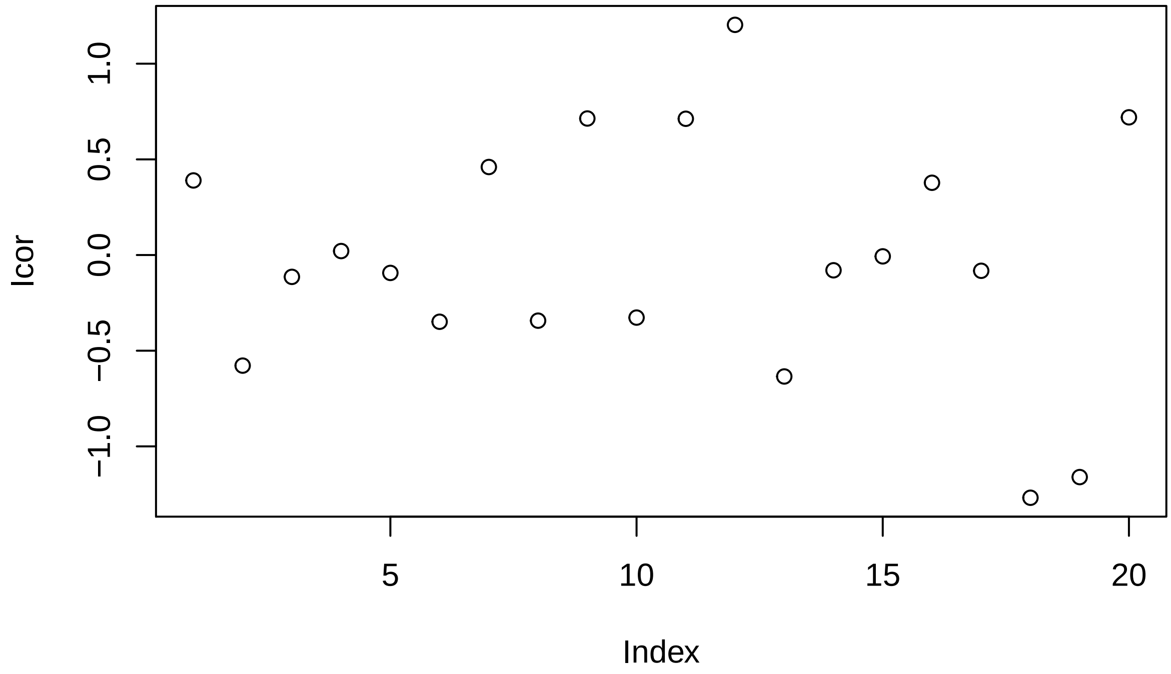
For example, we have the intensities of one peak from ten samples in two batches like the following demo:

```
set.seed(42)
# raw data
I = c(rnorm(10,mean = 0, sd = 0.5),rnorm(10,mean = 1, sd = 0.5))
```

```
# batch
B = c(rep(0,10),rep(1,10))
# qc
Iqc = c(rnorm(1,mean = 0, sd = 0.5),rnorm(1,mean = 1, sd = 0.5))
# corrected data
Icor = I - c(rep(mean(I[1:10]),10),rep(mean(I[11:20]),10)) + median(Iqc)
# plot the result
plot(I)
```



```
plot(Icor)
```



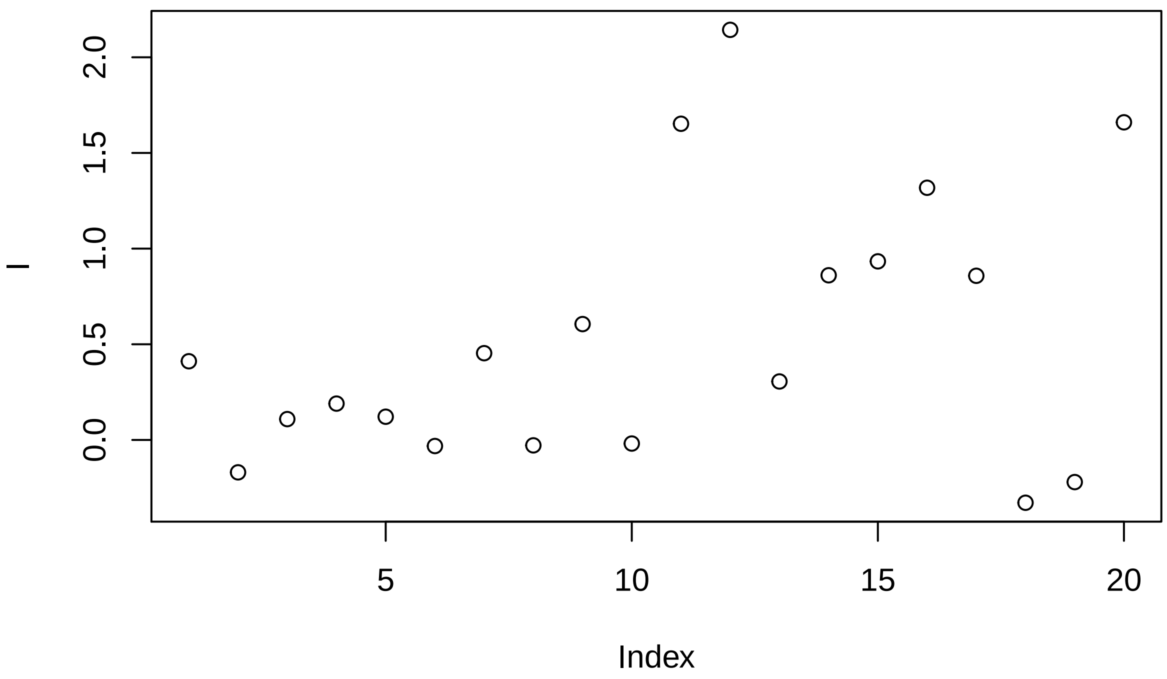
4.6.1.3 Scaling

For peak p of sample s in certain batch b, the corrected abundance I is:

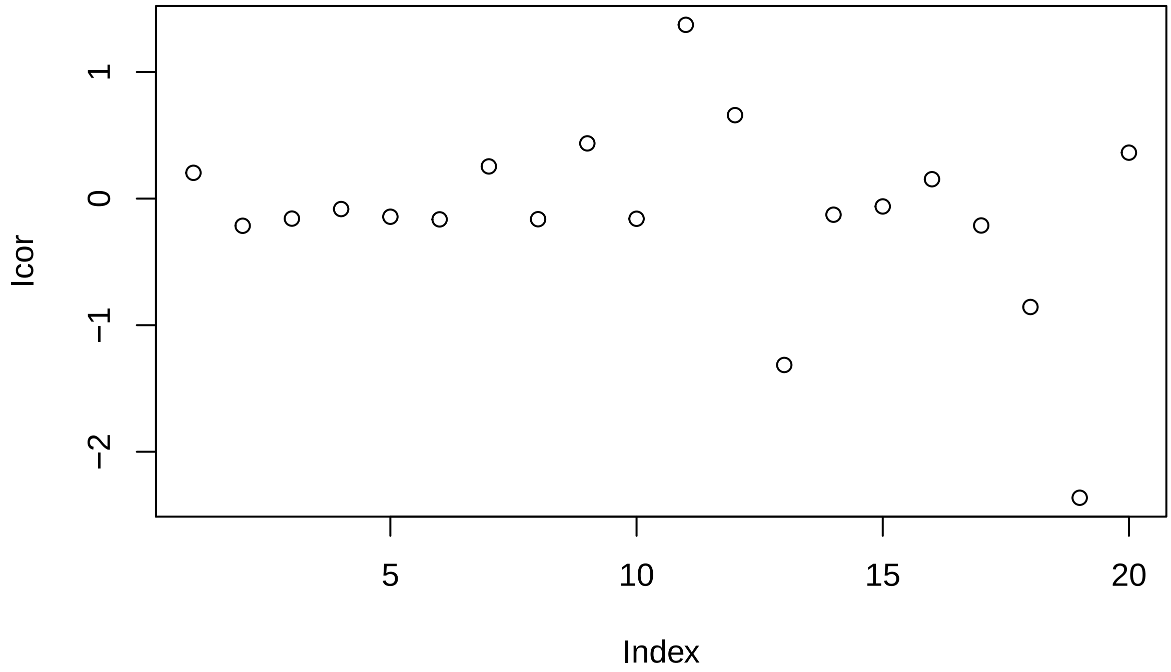
$$\hat{I}_{p,s,b} = \frac{I_{p,s,b} - mean(I_{p,b})}{std_{p,b}} * std_{p,qc,b} + mean(I_{p,qc,b})$$

For example, we have the intensities of one peak from ten samples in two batches like the following demo:

```r
set.seed(42)
# raw data
I = c(rnorm(10,mean = 0, sd = 0.3),rnorm(10,mean = 1, sd = 0.5))
# batch
B = c(rep(0,10),rep(1,10))
# qc
Iqc = c(rnorm(1,mean = 0, sd = 0.3),rnorm(1,mean = 1, sd = 0.5))
# corrected data
Icor = (I - c(rep(mean(I[1:10]),10),rep(mean(I[11:20]),10)))/c(sd(I[1:10]),sd(I[11:20]))*c(rep(0.3,10),
# plot the result
plot(I)
```



```
plot(Icor)
```

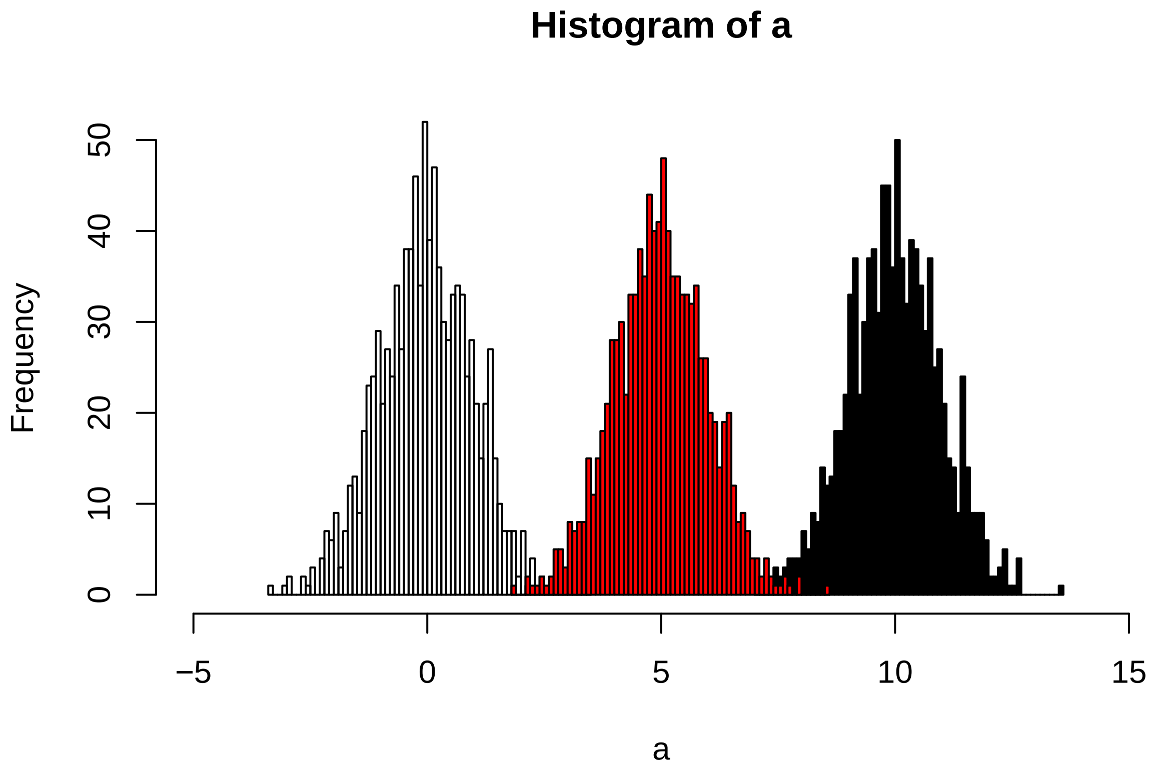


4.6.1.4 Quantile

The idea of quantile calibration is that alignment of the intensities in certain samples according to quantiles in each sample.

Here is the demo:

```
set.seed(42)
a <- rnorm(1000)
# b sufferred batch effect with a bias of 10
b <- rnorm(1000,10)
hist(a,xlim=c(-5,15),breaks = 50)
hist(b,col = 'black', breaks = 50, add=T)
# quantile normalized
cor <- (a[order(a)]+b[order(b)])/2
# reorder
cor <- cor[order(order(a))]
hist(cor,col = 'red', breaks = 50, add=T)
```

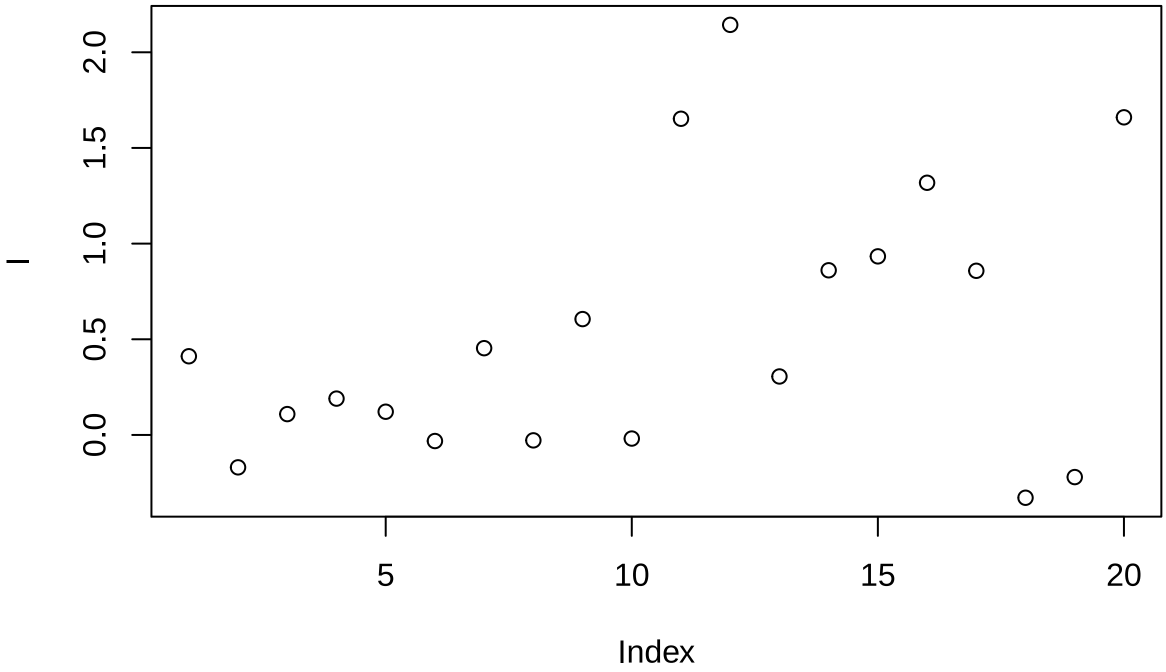


4.6.1.5 Ratio based calibraton

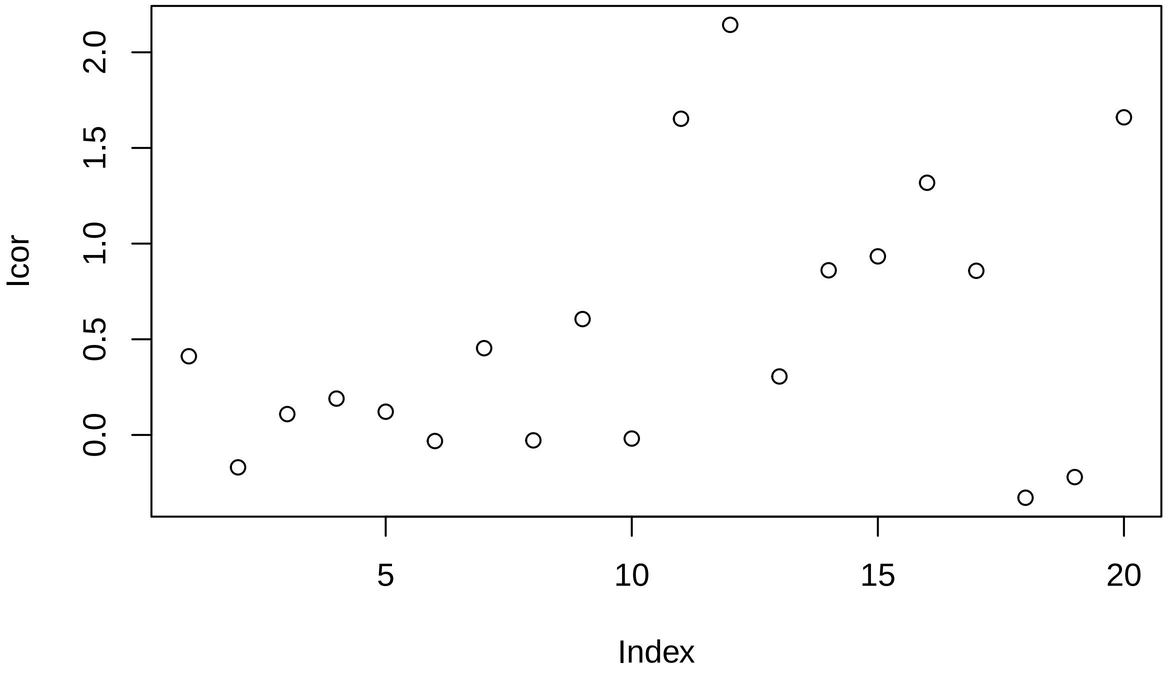
This method calibrates samples by the ratio between qc samples in all samples and in certain batch.For peak p of sample s in certain batch b, the corrected abundance I is:

$$\hat{I}_{p,s,b} = \frac{I_{p,s,b} * median(I_{p,qc})}{mean_{p,qc,b}}$$

```
set.seed(42)
# raw data
I = c(rnorm(10,mean = 0, sd = 0.3),rnorm(10,mean = 1, sd = 0.5))
# batch
B = c(rep(0,10),rep(1,10))
# qc
Iqc = c(rnorm(1,mean = 0, sd = 0.3),rnorm(1,mean = 1, sd = 0.5))
# corrected data
Icor = I * median(c(rep(Iqc[1],10),rep(Iqc[2],10)))/mean(c(rep(Iqc[1],10),rep(Iqc[2],10)))
# plot the result
plot(I)
```



```
plot(Icor)
```



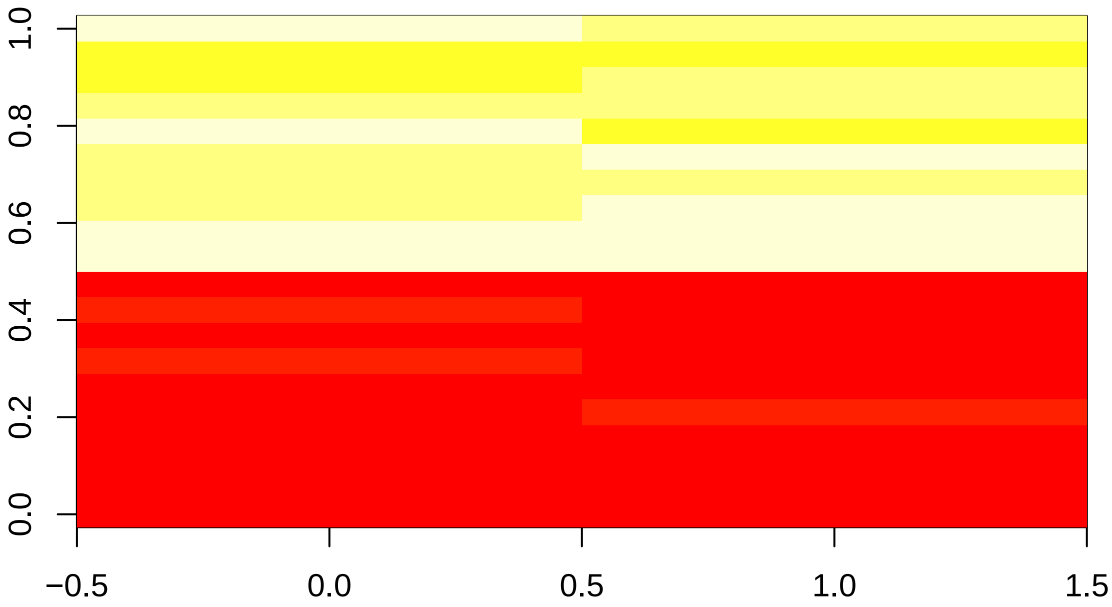
4.6.1.6 Linear Normalizer

This method initially scales each sample so that the sum of all peak abundances equals one. In this study, by multiplying the median sum of all peak abundances across all samples,we got the corrected data.

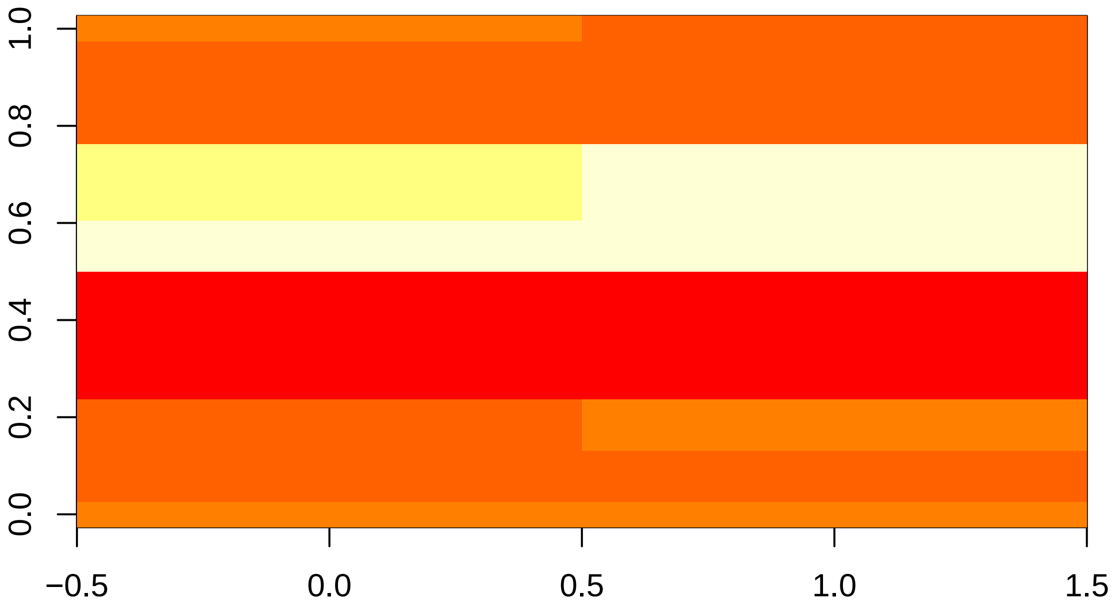
```
set.seed(42)
# raw data
peaksa <- c(rnorm(10,mean = 10, sd = 0.3),rnorm(10,mean = 20, sd = 0.5))
peaksb <- c(rnorm(10,mean = 10, sd = 0.3),rnorm(10,mean = 20, sd = 0.5))

df <- rbind(peaksa,peaksb)
dfcor <- df/apply(df,2,sum)* sum(apply(df,2,median))

image(df)
```



```
image(dfcor)
```



4.6.1.7 Internal standards

$$\hat{I}_{p,s} = \frac{I_{p,s} * median(I_{IS})}{I_{IS,s}}$$

Some methods also use pooled calibration samples and multiple internal standard strategy to correct the data(**?**). Also some methods only use QC samples to handle the data(Kuligowski et al., 2015).

4.6.2 Supervised methods

4.6.2.1 Regression calibration

Considering the batch effect of injection order, regress the data by a linear model to get the calibration.

4.6.2.2 Batch Normalizer

Use the total abundance scale and then fit with the regression line(Wang et al., 2013).

4.6.2.3 Surrogate Variable Analysis(SVA)

We have a data matrix(M*N) with M stands for indentity peaks from one sample and N stand for individual samples. For one sample, $X = (x_{i1}, ..., x_{in})^T$ stands for the normalized intensities of peaks. We use $Y = (y_i, ..., y_m)^T$ stands for the group infomation of our data. Then we could build such modles:

$$x_{ij} = \mu_i + f_i(y_i) + e_{ij}$$

μ_i stands for the baseline of the peak intensities in a normal state. Then we have:

$$f_i(y_i) = E(x_{ij}|y_j) - \mu_i$$

stands for the biological variations caused by the our group, for example, whether treated by pollutions or not.

However, considering the batch effects, the real model could be:

$$x_{ij} = \mu_i + f_i(y_i) + \sum_{l=1}^{L} \gamma_{li} p_{lj} + e^*_{ij}$$

γ_{li} stands for the peak-specific coefficient for potentical factor l. p_{lj} stands for the potential factors across the samples. Actually, the error item e_{ij} in real sample could always be decomposed as $e_{ij} = \sum_{l=1}^{L} \gamma_{li} p_{lj} + e^*_{ij}$ with e^*_{ij} standing for the real random error in certain sample for certain peak.

We could not get the potential factors directly. Since we don't care the details of the unknown factors, we could estimate orthogonal vectors h_k standing for such potential factors. Thus we have:

$$x_{ij} = \mu_i + f_i(y_i) + \sum_{l=1}^{L} \gamma_{li} p_{lj} + e^*_{ij} = \mu_i + f_i(y_i) + \sum_{k=1}^{K} \lambda_{ki} h_{kj} + e_{ij}$$

Here is the details of the algorithm:

> The algorithm is decomposed into two parts: detection of unmodeled factors and construction of surrogate variables

4.6.2.3.1 Detection of unmodeled factors

- Estimate $\hat{\mu}_i$ and f_i by fitting the model $x_{ij} = \mu_i + f_i(y_i) + e_{ij}$ and get the residual $r_{ij} = x_{ij} - \hat{\mu}_i - \hat{f}_i(y_i)$. Then we have the residual matrix R.
- Perform the singular value decompositon(SVD) of the residual matrix $R = UDV^T$
- Let d_l be the lth eigenvalue of the diagonal matrix D for $l = 1, ..., n$. Set df as the freedom of the model $\hat{\mu}_i + \hat{f}_i(y_i)$. We could build a statistic T_k as:

$$T_k = \frac{d_k^2}{\sum_{l=1}^{n-df} d_l^2}$$

to show the variance explained by the kth eigenvalue.

- Permute each row of R to remove the structure in the matrix and get R^*.
- Fit the model $r^*_{ij} = \mu^*_i + f^*_i(y_i) + e^*_{ij}$ and get $r^0_{ij} = r^*_{ij} - \hat{\mu}^*_i - \hat{f}^*_i(y_i)$ as a null matrix R_0
- Perform the singular value decompositon(SVD) of the residual matrix $R_0 = U_0 D_0 V_0^T$
- Compute the null statistic:

$$T_k^0 = \frac{d_{0k}^2}{\sum_{l=1}^{n-df} d_{0l}^2}$$

- Repeat permuting the row B times to get the null statistics T_k^{0b}
- Get the p-value for eigengene:

$$p_k = \frac{\# T_k^{0b} \geq T_k; b = 1, ..., B}{B}$$

- For a significance level α, treat k as a significant signature of residual R if $p_k \leq \alpha$

4.6.2.3.2 Construction of surrogate variables

- Estimate $\hat{\mu}_i$ and f_i by fitting the model $x_{ij} = \mu_i + f_i(y_i) + e_{ij}$ and get the residual $r_{ij} = x_{ij} - \hat{\mu}_i - \hat{f}_i(y_i)$. Then we have the residual matrix R.
- Perform the singular value decompositon(SVD) of the residual matrix $R = UDV^T$. Let $e_k = (e_{k1}, ..., e_{kn})^T$ be the kth column of V
- Set $\hat{K}$ as the significant eigenvalues found by the first step.
- Regress each e_k on x_i, get the p-value for the association.
- Set π_0 as the proportion of the peak intensity x_i not associate with e_k and find the numbers $\hat{m} = [1 - \hat{\pi}_0 \times m]$ and the indices of the peaks associated with the eigenvalues
- Form the matrix $\hat{m}_1 \times N$, this matrixX_r stand for the potiential variables. As was done for R, get the eigengents of X_r and denote these by e_j^r
- Let $j^* = argmax_{1 \leq j \leq n} cor(e_k, e_j^r)$ and set $\hat{h}_k = e_j^r$. Set the estimate of the surrogate variable to be the eigenvalue of the reduced matrix most correlated with the corresponding residual eigenvalue. Since the reduced matrix is enriched for peaks associated with this residual eigenvalue, this is a principled choice for the estimated surrogate variable that allows for correlation with the primary variable.
- Employ the $\mu_i + f_i(y_i) + \sum_{k=1}^{K} \gamma_{ki} \hat{h}_{kj} + e_{ij}$ as te estimate of the ideal model $\mu_i + f_i(y_i) + \sum_{k=1}^{K} \gamma_{ki} h_{kj} + e_{ij}$

This method could found the potentical unwanted variables for the data. SVA were introduced by Jeff Leek(Leek et al., 2012; Leek and Storey, 2007, 2008) and EigenMS package implement SVA with modifications including analysis of data with missing values that are typical in LC-MS experiments(Karpievitch et al., 2014).

4.6.2.4 RUV (Remove Unwanted Variation)

This method's performance is similar to SVA. Instead find surrogate variable from the whole dataset. RUA use control or pool QC to find the unwanted variances and remove them to find the peaks related to experimental design. RUA-random(Livera et al., 2015) furthor use linear mixed model to estimate the variances of random error. This method could be used with suitable control, which is commen in metabolomics DoE.

4.6.2.5 RRmix

RRmix also use a latent factor models correct the data(Jr et al., 2017). This method could be treated as linear mixed model version SVA. No control samples are required and the unwanted variances could be removed by factor analysis. This method might be the best choise to remove the unwanted variables with commen experiment design.

Chapter 5

Peaks selection

After we get corrected peaks across samples, the next step is finding the differences between two groups. Actually, you could perform ANOVA or Kruskal-Wallis Test for comparison among more than two groups. The basic idea behind statistic analysis is to find the meaningful differences between groups and extract such ions or peak groups.

So how to find the differences? In most metabolomics software, such task is completed by a t-test and report p-value and fold changes. If you only compare two groups on one peaks, that's OK. However, if you compare two groups on thousands of peaks, statistic textbook would tell you to notice the false positive. For one comparasion, the confidence level is 0.05, which means 5% chances to get false positive result. For two comparasions, such chances would be $1 - 0.95^2$. For 10 comparasions, such chances would be $1 - 0.95^{10} = 0.4012631$. For 100 comparasions, such chances would be $1 - 0.95^{100} = 0.9940795$. You would almost certainly to make mistakes for your results.

In statistics, the false discovery rate(FDR) control is always mentioned in omics studies for mutiple tests. I suggested using q-values to control FDR. If q-value is less than 0.05, we should expect a lower than 5% chances we make the wrong selections for all of the comparisions showed lower q-values in the whole dataset. Also we could use local false discovery rate, which showed the FDR for certain peaks. However, such values are hard to be estimated accurately.

Karin Ortmayr thought fold change might be better than p-values to find the differences(**?**).

5.1 Peak misidentification

- Isomer

Use seperation methods such as chromatography, ion mobility MS, MS/MS. Reversed-phase ion-pairing chromatography and HILIC is useful and chemical derivatization is another options.

- Interfering compounds

20ppm is the least resolution and accuracy

- In-source degradation products

Chapter 6

Annotation

When you get the peaks table or features table, annotation of the peaks would help you. Obviously, peaks with similar retention time might come from the same compound such as isotopic peaks, addictions from the analysis process. **CAMERA** (Kuhl et al., 2012) or **Ramcluster**(Broeckling et al., 2014) package could be used to make such annotation. Also this package could be used to group the peaks together for further analysis.

There are two major annotation ideas: multi-stage MS analysis and similarity analysis. The former requires certain instruments and database for MS/MS data. The later needs algorithms to predict the structures. This is a nice review for annotation(Allard et al., 2017).

6.1 Annotation v.s. identification

According to the defination from the Chemical Analysis Working Group of the Metabolomics Standards Intitvative(Sumner et al., 2007; Viant et al., 2017). Four levels of confidence could be assigned to identification:

- Level 1 'identified metabolites'
- Level 2 'Putatively annotated compounds'
- Level 3 'Putatively characterised compound classes'
- Level 4 'Unknown'

In practice, data analysis based annotation could reach level 2. For level 1, we need at extra methods such as MS/MS, retention time, accurate mass, 2D NMR spectra, and so on to confirm the compounds. However, standards are always required for solid proof.

6.2 Network analysis

Network analysis is based on the inner correlationship in the data and make network analysis at peaks' level instand of annotated compounds. Usually such analysis could also be used to annotated peaks with supports from mass spectrum database. GNPS use such ananlysis to annotate the data, as well as Metscape based on Debiased Sparse Partial Correlation (DSPC) algorithm(Basu et al., 2017).

6.3 Tools

- Plantmat: excel library based pridiction for plant metabolites(Qiu et al., 2016).
- MetFamily: Shiny app for MS and MS/MS data annotation(Treutler et al., 2016).
- xMSannotator: R pacakges, from Emory University(Uppal et al., 2017).
- Lipidmatch: in silico lipid mass spectrum search (Koelmel et al., 2017)

Chapter 7

Omics analysis

When you get the filtered ions, the next step is making annotations for them. Such annotations would be helpful for omics studies.

Since we have got the annotations, Omics analysis could be performed.Upload the data obtained from the **xcms** to other tools or databases.

You will get an updated database list here

Right now, it is hard to connect different omics databases such as gene, protein and metabolites together for a whole scope of certain biological process. However, you might select few metabolites across those databases and find something interesting.

7.1 Pathway analysis

Pathwat analysis maps annotated data into known pathway and make statistical analysis to find the influenced pathway or the compounds with high inflences on certain pathway.

7.2 Network analysis

Mummichog could make pathway and network analysis without annotation.

MSS: sequential feature screening procedure to select important sub-network and identify the optimal matching for metabolimics data (Cai et al., 2017)

7.3 Omics integration

Chapter 8

Common analysis methods for metabolomics

8.1 PCA

In most cases, PCA is used as an exploratory data analysis(EDA) method. In most of those most cases, PCA is just served as visualization method. I mean, when I need to visualize some high-dimension data, I would use PCA.

So, the basic idea behind PCA is compression. When you have 100 samples with concentrations of certain compound, you could plot the concentrations with samples' ID. However, if you have 100 compounds to be analyzed, it would by hard to show the relationship between the samples. Actually, you need to show a matrix with sample and compounds (100 * 100 with the concentrations filled into the matrix) in an informal way.

The PCA would say: OK, guys, I could convert your data into only 100 * 2 matrix with the loss of information minimized. Yeah, that is what the mathematical guys or computer programmer do. You just run the command of PCA. The new two "compounds" might have the cor-relationship between the original 100 compounds and retain the variances between them. After such projection, you would see the compressed relationship between the 100 samples. If some samples' data are similar, they would be projected together in new two "compounds" plot. That is why PCA could be used for cluster and the new "compounds" could be referred as principal components(PCs).

However, you might ask why only two new compounds could finished such task. I have to say, two PCs are just good for visualization. In most cases, we need to collect PCs standing for more than 80% variances in our data if you want to recovery the data with PCs. If each compound have no relationship between each other, the PCs are still those 100 compounds. So you have found a property of the PCs: PCs are orthogonal between each other.

Another issue is how to find the relationship between the compounds. We could use PCA to find the relationship between samples. However, we could also extract the influences of the compounds on certain PCs. You might find many compounds showed the same loading on the first PC. That means the concentrations pattern between the compounds are looked similar. So PCA could also be used to explore the relationship between the compounds.

OK, next time you might recall PCA when you need it instead of other paper showed them.

Besides, there are some other usage of PCA. Loadings are actually correlation coefficients between peaks and their PC scores. Yamamoto et.al.(Yamamoto et al., 2014) used t-test on this correlation coefficient and thought the peaks with statistically significant correlation to the PC score have biological meanings for further study such as annotation. However, such analysis works better when few PCs could explain most of the variances in the datasets.

8.2 Cluster Analysis

After we got a lot of samples and analyzed the concentrations of many compounds in them, we may ask about the relationship between the samples. You might have the sampling information such as the date and the position and you could use boxplot or violin plot to explore the relationships among those categorical variables. However, you could also use the data to find some potential relationship.

But how? if two samples' data were almost the same, we might think those samples were from the same potential group. On the other hand, how do we define the "same" in the data?

Cluster analysis told us that just define a "distances" to measure the similarity between samples. Mathematically, such distances would be shown in many different manners such as the sum of the absolute values of the differences between samples.

For example, we analyzed the amounts of compound A, B and C in two samples and get the results:

Compounds(ng)	A	B	C
Sample 1	10	13	21
Sample 2	54	23	16

The distance could be:

$$distance = |10 - 54| + |13 - 23| + |21 - 16| = 59$$

Also you could use the sum of squares or other way to stand for the similarity. After you defined a "distance", you could get the distances between all of pairs for your samples. If two samples' distance was the smallest, put them together as one group. Then calculate the distances again to combine the small group into big group until all of the samples were include in one group. Then draw a dendrogram for those process.

The following issue is that how to cluster samples? You might set a cut-off and directly get the group from the dendrogram. However, sometimes you were ordered to cluster the samples into certain numbers of groups such as three. In such situation, you need K means cluster analysis.

The basic idea behind the K means is that generate three virtual samples and calculate the distances between those three virtual samples and all of the other samples. There would be three values for each samples. Choose the smallest values and class that sample into this group. Then your samples were classified into three groups. You need to calculate the center of those three groups and get three new virtual samples. Repeat such process until the group members unchanged and you get your samples classified.

OK, the basic idea behind the cluster analysis could be summarized as define the distances, set your cut-off and find the group. By this way, you might show potential relationships among samples.

8.3 PLSDA

PLS-DA, OPLS-DA and HPSO-OPLS-DA(Yang et al., 2017) could be used.

8.4 Self-organizing map

Chapter 9

Demo

9.1 Project Setup

I suggest building your data analysis projects in RStudio(Click File - New project - New dictionary - Empty project). Then assign a name for your project. I also recommend the following tips if you are familiar with it.

- Use git/github to make version control of your code and sync your project online.
- NOT use your name for your project because other peoples might cooperate with you and someone might check your data when you publish your papers. Each project should be a work for one paper or one chapter in your thesis.
- Use **workflow** document(txt or doc) in your project to record all of the steps and code you performed for this project. Treat this document as digital version of your experiment notebook
- Use **data** folder in your project folder for the raw data and the results you get in data analysis
- Use **figure** folder in your project folder for the figure
- Use **munuscript** folder in your project folder for the manuscript (you could write paper in rstudio with the help of template in Rmarkdown)
- Just double click [**yourprojectname**].**Rproj** to start your project

9.2 Data input

xcms does not support all of the Raw files from every mass spectrometry manufacturers. You need to convert your Raw data into some open-source data format such as mzData, mzXML or CDF files. The tool is **MScovert** from **ProteoWizard**.

Here is a demo:

```r
# install the packages for data analysis and
# source("https://bioconductor.org/biocLite.R")
# biocLite(c("multtest","faahKO","xcms","qvalue","CAMERA"))
# load the functions and dataset for demo

library(multtest)
library(xcms)
library(faahKO)
library(BiocParallel)
# get the demo data in faahKO packages
cdfpath <- system.file("cdf",package = "faahKO")
# show the name of demo data
list.files(cdfpath,recursive = T)
```

```
##  [1] "KO/ko15.CDF" "KO/ko16.CDF" "KO/ko18.CDF" "KO/ko19.CDF" "KO/ko21.CDF"
##  [6] "KO/ko22.CDF" "WT/wt15.CDF" "WT/wt16.CDF" "WT/wt18.CDF" "WT/wt19.CDF"
## [11] "WT/wt21.CDF" "WT/wt22.CDF"
```

Here is a demo for *xcmsSet*:

```r
cdffiles <- list.files(cdfpath, recursive = TRUE, full.names = TRUE)
xset <- xcmsSet(cdffiles)
xset
```

```
## An "xcmsSet" object with 12 samples
##
## Time range: 2506.1-4147.7 seconds (41.8-69.1 minutes)
## Mass range: 200.1-599.3338 m/z
## Peaks: 4721 (about 393 per sample)
## Peak Groups: 0
## Sample classes: KO, WT
##
## Feature detection:
##  o Peak picking performed on MS1.
## Profile settings: method = bin
##                   step = 0.1
##
## Memory usage: 0.741 MB
```

9.3 Find the peaks

The first step to process the MS data is that find the peaks against the noises. In **xcms**, all of related staffs are handled by *xcmsSet* function.

For any functions in **xcms** or **R**, you could get their documents by type `?` before certain function. Another geek way is input the name of the function in the console of Rstudio and press F1 for help.

```r
?xcmsSet
```

In the document of *xcmsset*, we could set the sample classes, profmethod, profparam, polarity,etc. In the online version, such configurations are shown in certain windows. In the local analysis environment, such parameters are setup by yourselves. However, I think the default configurations could satisfied most of the analysis because related information should have been recorded in your Raw data and **xcms** could find them. All you need to do is that show the data dictionary for *xcmsSet*.

If your data have many groups such as control and treated group, just put them in separate subfolder of the data folder and *xcmsSet* would read them as separated groups.

The output was an object with class of *xcmsSet*. You could see a summary by type the name. In this cases, *xcmsSet* found 4721 peaks with time range 41.8-69.1 min and mass range 200.1-599.3338 m/z in the 12 samples.

Another function which might be useful is `group`. This function will add additional information about the same analytes for `xcmsSet` objects.

```
xset <- group(xset)
xset
```

```
## An "xcmsSet" object with 12 samples
##
## Time range: 2506.1-4147.7 seconds (41.8-69.1 minutes)
## Mass range: 200.1-599.3338 m/z
## Peaks: 4721 (about 393 per sample)
## Peak Groups: 403
## Sample classes: KO, WT
##
## Feature detection:
##  o Peak picking performed on MS1.
## Profile settings: method = bin
##                   step = 0.1
##
## Memory usage: 0.805 MB
```

Now you see there are 403 groups in the demo data, which meant 403 analytes are found across 4721 peaks.

9.4 Data correction

Reasons of data correction might come from many aspects such as the unstable instrument and pollution on column. In **xcms**, the most important correction is retention time correction.

Remember the original retention time might changed and use another object to save the new object:

```
xset2 <- retcor(xset, method = "obiwarp")
```

```
## center sample:  ko16
## Processing: ko15  ko18  ko19  ko21  ko22  wt15  wt16  wt18  wt19  wt21  wt22
```

```
xset2
```

```
## An "xcmsSet" object with 12 samples
##
## Time range: 2506.3-4162.2 seconds (41.8-69.4 minutes)
## Mass range: 200.1-599.3338 m/z
## Peaks: 4721 (about 393 per sample)
## Peak Groups: 0
## Sample classes: KO, WT
##
## Feature detection:
##  o Peak picking performed on MS1.
## Profile settings: method = bin
##                     step = 0.1
##
## Memory usage: 0.741 MB
```

```
# you need group the peaks again for this corrected data
xset2 <- group(xset2)
xset2
```

```
## An "xcmsSet" object with 12 samples
##
## Time range: 2506.3-4162.2 seconds (41.8-69.4 minutes)
## Mass range: 200.1-599.3338 m/z
## Peaks: 4721 (about 393 per sample)
## Peak Groups: 404
## Sample classes: KO, WT
##
## Feature detection:
##  o Peak picking performed on MS1.
## Profile settings: method = bin
##                     step = 0.1
##
## Memory usage: 0.805 MB
```

You see one more peak groups after the correction. After the retention time correction, we also need to correct the peak groups by filling the missing peaks. Such function calls *fillpeaks*:

```
xset3 <- fillPeaks(xset2,BPPARAM=SnowParam())
xset3
```

```
## An "xcmsSet" object with 12 samples
##
## Time range: 2502.9-4162.2 seconds (41.7-69.4 minutes)
## Mass range: 200.1-599.3338 m/z
## Peaks: 6054 (about 504 per sample)
## Peak Groups: 404
## Sample classes: KO, WT
##
## Feature detection:
##  o Peak picking performed on MS1.
## Profile settings: method = bin
##                     step = 0.1
##
## Memory usage: 0.946 MB
```

You see more peaks found.

9.5 Statistic analysis

Right now we get peaks across samples, the next step is finding the differences between two groups. You will find the P values of t-test for pairwise comparison:

```
reporttab <- diffreport(xset3, "WT", "KO", "example")
reporttab[1:3,]
```

```
##        name     fold    tstat       pvalue    mzmed    mzmin    mzmax
## 1 M300T3391 5.693594 14.44368 5.026336e-08 300.1898 300.1706 300.2000
## 2 M301T3391 6.283030 15.52501 5.385022e-08 301.1879 301.1659 301.1949
## 3 M298T3185 3.984984 11.88773 3.615841e-07 298.1508 298.1054 298.1592
##      rtmed    rtmin    rtmax npeaks KO WT      ko15      ko16      ko18
## 1 3390.699 3374.142 3398.743     12  6  6 4534353.6 4980914.5 5290739.1
## 2 3391.126 3385.366 3394.937      7  6  1  962353.4 1047934.1 1109303.0
## 3 3185.221 3182.083 3190.163      4  4  0  180780.8  204134.9  191015.9
##        ko19      ko21      ko22      wt15      wt16      wt18      wt19
## 1 4564262.9 4733236.1 3931592.6 349660.89 491793.18 645526.70 634108.85
## 2  946943.4  984787.2  806171.5  80639.28 118940.90 134531.39 102784.65
## 3  190626.8  155276.9  220288.6  16448.42  41050.04  50082.55  76704.81
##         wt21       wt22
## 1 1438254.45 1364627.84
## 2  203982.76  291392.97
## 3   53957.78   48363.33
```

Now you have got the ions that varies a lot between groups. Such ions are things we should take care of. In a ideal case, this is the endpoint of your study and the left work is making a report of your finding.

However,we need q-values to control FDR. To get the q-values, you need input p-values and use the function from **qvalue** package.

```
library(qvalue)
# extract the p-value to caculate q-value
qvalue <- qvalue(p=reporttab$pvalue)
# add qvalue to reporttab
reporttab$qvalue <- qvalue$qvalues
# reporttab[1:3,]
```

For further information about q-value, check here.

After the FDR control, the following steps depend on your study.

9.6 Annotation

I suggest **CAMERA** package to handle this task. You need to prepare an object of class *xcmsSet*, for example, *xset3*(remember to use *fillpeaks* to get the ions group).

```
library(CAMERA)
# Create an xsAnnotate object
xsa <- xsAnnotate(xset3)
# Group after RT value of the xcms grouped peak
xsaF <- groupFWHM(xsa, perfwhm=0.6)
```

```
## Start grouping after retention time.
## Created 132 pseudospectra.
```

```
# Verify grouping
xsaC <- groupCorr(xsaF)
```

```
## Start grouping after correlation.
## Generating EIC's ..
##
## Calculating peak correlations in 132 Groups...
##  % finished: 10  20  30  40  50  60  70  80  90  100
##
## Calculating graph cross linking in 132 Groups...
##  % finished: 10  20  30  40  50  60  70  80  90  100
## New number of ps-groups:  202
## xsAnnotate has now 202 groups, instead of 132
```

```
# Annotate isotopes, could be done before groupCorr
xsaFI <- findIsotopes(xsaC)
```

```
## Generating peak matrix!
## Run isotope peak annotation
##  % finished: 10  20  30  40  50  60  70  80  90  100
## Found isotopes: 57
```

```
# Annotate adducts
xsaFA <- findAdducts(xsaFI, polarity="positive")
```

```
## Generating peak matrix for peak annotation!
##
## Calculating possible adducts in 202 Groups...
##  % finished: 10  20  30  40  50  60  70  80  90  100
```

```
# See the results
getPeaklist(xsaFA)[1:3,]
```

```
##         mz    mzmin    mzmax       rt    rtmin    rtmax npeaks KO WT
## 1 200.1000 200.1000 200.1000 2924.027 2876.967 2939.450      9  4  5
## 2 205.0000 205.0000 205.0000 2788.377 2782.719 2795.550     12  6  6
## 3 205.9927 205.9786 206.0023 2789.144 2782.719 2793.925     12  6  6
##        ko15      ko16       ko18       ko19       ko21      ko22      wt15
## 1  147887.5  451600.7   65290.38   52834.57   70042.53  162012.4  175177.1
## 2 1778568.9 1567038.1 1482796.38 1039129.82 1223132.35 1072037.7 1950287.5
## 3  237993.6  269714.0  201393.42  150107.31  176989.65  156797.0  276541.8
##         wt16       wt18       wt19      wt21       wt22  isotopes
## 1   82619.48   46255.03   69198.22  153273.5   98144.28
## 2 1466780.60 1572679.16 1275312.76 1356014.3 1231442.16   [1][M]+
## 3  222366.15  211717.71  186850.88  188285.9  172348.76 [1][M+1]+
##             adduct pcgroup
## 1                      165
## 2 [M+Na]+ 182.007        5
## 3                        5
```

```
# Get final peaktable and store on harddrive
# write.csv(getPeaklist(xsaFA),file="data/result_CAMERA.csv")
```

Any steps after the *annotation* could be operated solo and you may not need the isotopes or adducts. You could also use *annotateDiffreport* to show the results as *diffreport* in **xcms**.

```
# make a diffreport with CAMERA result and extract the fold change higher than 3
dreport <- annotateDiffreport(xset3, fc_th = 3)
```

```
## Start grouping after retention time.
## Created 132 pseudospectra.
## Generating peak matrix!
## Run isotope peak annotation
##  % finished: 10  20  30  40  50  60  70  80  90  100
## Found isotopes: 68
## Start grouping after correlation.
## Generating EIC's ..
##
## Calculating peak correlations in 34 Groups...
##  % finished: 10  20  30  40  50  60  70  80  90  100
##
## Calculating graph cross linking in 34 Groups...
##  % finished: 10  20  30  40  50  60  70  80  90  100
## New number of ps-groups:  156
## xsAnnotate has now 156 groups, instead of 132
## Generating peak matrix for peak annotation!
##
## Calculating possible adducts in 58 Groups...
##  % finished: 10  20  30
```

```
# extract the p-value to caculate q-value
qvalue <- qvalue(p=dreport$pvalue)
# add qvalue to reporttab
dreport$qvalue <- qvalue$qvalues
# See the results
# dreport[1:3,]
# save on harddrive
# write.csv(dreport,file='data/diffreport.csv')
```

9.7 Omics analysis

Since we have got the annotations, Omics analysis could be performed. In **xcms**, the default database is **metlin**. You could directly get the link to certain compounds when you generate the differences report.

```
# make a diffreport with CAMERA result and extract the fold change higher than 3, add the metlin links
dreport <- annotateDiffreport(xset3, fc_th = 3, metlin = T)
```

```
## Start grouping after retention time.
## Created 132 pseudospectra.
## Generating peak matrix!
## Run isotope peak annotation
##  % finished: 10  20  30  40  50  60  70  80  90  100
## Found isotopes: 68
## Start grouping after correlation.
## Generating EIC's ..
##
## Calculating peak correlations in 34 Groups...
##  % finished: 10  20  30  40  50  60  70  80  90  100
##
## Calculating graph cross linking in 34 Groups...
##  % finished: 10  20  30  40  50  60  70  80  90  100
## New number of ps-groups:  156
## xsAnnotate has now 156 groups, instead of 132
## Generating peak matrix for peak annotation!
##
## Calculating possible adducts in 58 Groups...
##  % finished: 10  20  30
```

```
# extract the p-value to caculate q-value
qvalue <- qvalue(p=dreport$pvalue)
# add qvalue to reporttab
dreport$qvalue <- qvalue$qvalues
# See the results
dreport[1:3,]
```

```
##                   name     fold     tstat       pvalue    mzmed    mzmin
## 300.2/3391 M300T3391 5.693594 -14.44368 5.026336e-08 300.1898 300.1706
## 301.2/3391 M301T3391 6.283030 -15.52501 5.385022e-08 301.1879 301.1659
## 298.2/3185 M298T3185 3.984984 -11.88773 3.615841e-07 298.1508 298.1054
##                mzmax    rtmed    rtmin    rtmax npeaks KO WT
## 300.2/3391 300.2000 3390.699 3374.142 3398.743     12  6  6
## 301.2/3391 301.1949 3391.126 3385.366 3394.937      7  6  1
## 298.2/3185 298.1592 3185.221 3182.083 3190.163      4  4  0
##                                                                           metlin
## 300.2/3391 http://metlin.scripps.edu/metabo_list.php?mass_min=298.2&mass_max=300.2
## 301.2/3391 http://metlin.scripps.edu/metabo_list.php?mass_min=299.2&mass_max=301.2
## 298.2/3185 http://metlin.scripps.edu/metabo_list.php?mass_min=296.2&mass_max=298.2
##                        ko15             ko16             ko18
## 300.2/3391 4534353.62273683 4980914.48421051 5290739.13866664
## 301.2/3391 962353.429578945 1047934.14136842 1109303.04472222
## 298.2/3185 180780.817277777 204134.864631578 191015.910842105
##                        ko19             ko21             ko22
## 300.2/3391 4564262.89684209 4733236.07999997      3931592.586
## 301.2/3391 946943.392842103 984787.204999993 806171.472899999
## 298.2/3185  190626.84952381 155276.902163857      220288.6218
##                        wt15             wt16             wt18
## 300.2/3391  349660.88536842 491793.181333331 645526.704947367
## 301.2/3391 80639.2842881944 118940.899088542  134531.38671875
## 298.2/3185 16448.4191894531 41050.0418122944 50082.5494449013
##                        wt19             wt21             wt22  isotopes
## 300.2/3391 634108.848947367 1438254.44559999      1364627.844   [4][M]+
## 301.2/3391 102784.647854275 203982.760512408 291392.971409092 [4][M+1]+
## 298.2/3185 76704.8068359375 53957.7833573191  48363.333932977
##            adduct pcgroup       qvalue
## 300.2/3391             17 1.087774e-05
## 301.2/3391             17 1.087774e-05
## 298.2/3185            103 4.869333e-05
```

```
# save on harddrive
# write.csv(dreport,file='data/diffreport.csv')
```

9.8 MetaboAnalyst

Actully, after you perform data correction, you have got the data matrix for statistic analysis. You might choose **MetaboAnalyst** online or offline to make furthor analysis, which supplied more statistical choices than xcms.

The input data format for **MetaboAnalyst** should be rows for peaks and colomns for samples. You could also add groups infomation if possible. Use the following code to get the data for analysis.

```
MAdata <- groupval(xset3,method = "medret", intensity = "into")
MAdata <- rbind(group = as.character(phenoData(xset)$class),MAdata)
# output the data for MetaboAnalyst
# write.csv(MAdata, file = "data/MAdata.csv")
```

9.9 Visulizing Peaks

If you find some significant peaks, the best way to check them is data visulization. **xcms** supplies such functions. All you need are the retention time and ions' range.

```
eic <- groups(xset3)
index <- which(eic[,"rtmed"] > 2500 & eic[,"rtmed"<2600])[1]
```

9.10 Optimation of XCMS

IPO package could be used to optimaze the parameters for XCMS. Try the following code.

```
mzdatapath <- system.file("cdf",package = "faahKO")
mzdatafiles <- list.files(mzdatapath, recursive = TRUE, full.names=TRUE)
library(IPO)
peakpickingParameters <- getDefaultXcmsSetStartingParams('matchedFilter')
#setting levels for min_peakwidth to 10 and 20 (hence 15 is the center point)
peakpickingParameters$min_peakwidth <- c(10,20)
peakpickingParameters$max_peakwidth <- c(26,42)
#setting only one value for ppm therefore this parameter is not optimized
peakpickingParameters$ppm <- 20
resultPeakpicking <-
  optimizeXcmsSet(files = mzdatafiles[6:9],
                  params = peakpickingParameters,
                  nSlaves = 4,
                  subdir = 'rsmDirectory')
```

```
optimizedXcmsSetObject <- resultPeakpicking$best_settings$xset

retcorGroupParameters <- getDefaultRetGroupStartingParams()
retcorGroupParameters$profStep <- 1
resultRetcorGroup <-
  optimizeRetGroup(xset = optimizedXcmsSetObject,
                   params = retcorGroupParameters,
                   nSlaves = 4,
                   subdir = "rsmDirectory")


writeRScript(resultPeakpicking$best_settings$parameters,
             resultRetcorGroup$best_settings,
             nSlaves=12)
# https://github.com/rietho/IPO/blob/master/vignettes/IPO.Rmd
```

9.11 Summary

This is the offline metaboliomics data process workflow. For each study, details would be different and F1 is always your best friend.

Enjoy yourself in data mining!

Chapter 10

Case Study(selected)

10.1 Cancer

- LC-QToF, Urine, Prostate (**?**)
- Muti, tobacco, Senescence (Li et al., 2016)
- Serum (Itoi et al., 2017)

10.2 Mental Health

- Internet gaming disorder (IGD) (Cho et al., 2017)

10.3 Interestion papers

- Ionmoics(Konz et al., 2017)

References

Bibliography

Allard, P.-M., Genta-Jouve, G., and Wolfender, J.-L. (2017). Deep metabolome annotation in natural products research: Towards a virtuous cycle in metabolite identification. *Current Opinion in Chemical Biology*, 36:40–49.

Alonso, A., Marsal, S., and Julià, A. (2015). Analytical Methods in Untargeted Metabolomics: State of the Art in 2015. *Front Bioeng Biotechnol*, 3.

Barnes, S., Benton, H. P., Casazza, K., Cooper, S. J., Cui, X., Du, X., Engler, J., Kabarowski, J. H., Li, S., Pathmasiri, W., Prasain, J. K., Renfrow, M. B., and Tiwari, H. K. (2016a). Training in metabolomics research. I. Designing the experiment, collecting and extracting samples and generating metabolomics data. *J. Mass Spectrom.*, 51(7):461–475.

Barnes, S., Benton, H. P., Casazza, K., Cooper, S. J., Cui, X., Du, X., Engler, J., Kabarowski, J. H., Li, S., Pathmasiri, W., Prasain, J. K., Renfrow, M. B., and Tiwari, H. K. (2016b). Training in metabolomics research. II. Processing and statistical analysis of metabolomics data, metabolite identification, pathway analysis, applications of metabolomics and its future. *J. Mass Spectrom.*, 51(8):535–548.

Basu, S., Duren, W., Evans, C. R., Burant, C. F., Michailidis, G., and Karnovsky, A. (2017). Sparse network modeling and metscape-based visualization methods for the analysis of large-scale metabolomics data. *Bioinformatics*, 33(10):1545–1553.

Bennett, B. D., Kimball, E. H., Gao, M., Osterhout, R., Van Dien, S. J., and Rabinowitz, J. D. (2009). Absolute metabolite concentrations and implied enzyme active site occupancy in Escherichia coli. *Nat Chem Biol*, 5(8):593–599.

Blaise, B. J., Correia, G., Tin, A., Young, J. H., Vergnaud, A.-C., Lewis, M., Pearce, J. T. M., Elliott, P., Nicholson, J. K., Holmes, E., and Ebbels, T. M. D. (2016). Power Analysis and Sample Size Determination in Metabolic Phenotyping. *Anal. Chem.*, 88(10):5179–5188.

Broeckling, C. D., Afsar, F. A., Neumann, S., Ben-Hur, A., and Prenni, J. E. (2014). RAMClust: A Novel Feature Clustering Method Enables Spectral-Matching-Based Annotation for Metabolomics Data. *Anal. Chem.*, 86(14):6812–6817.

Cai, Q., Alvarez, J. A., Kang, J., and Yu, T. (2017). Network Marker Selection for Untargeted LC–MS Metabolomics Data. *J. Proteome Res.*, 16(3):1261–1269.

Cajka, T. and Fiehn, O. (2016). Toward Merging Untargeted and Targeted Methods in Mass Spectrometry-Based Metabolomics and Lipidomics. *Anal. Chem.*, 88(1):524–545.

Cho, Y. U., Lee, D., Lee, J.-E., Kim, K. H., Lee, D. Y., and Jung, Y.-C. (2017). Exploratory metabolomics of biomarker identification for the internet gaming disorder in young Korean males. *Journal of Chromatography B*, 1057:24–31.

De Livera, A. M., Dias, D. A., De Souza, D., Rupasinghe, T., Pyke, J., Tull, D., Roessner, U., McConville, M., and Speed, T. P. (2012). Normalizing and Integrating Metabolomics Data. *Anal. Chem.*, 84(24):10768–10776.

Domingo-Almenara, X., Brezmes, J., Vinaixa, M., Samino, S., Ramirez, N., Ramon-Krauel, M., Lerin, C., Díaz, M., Ibáñez, L., Correig, X., Perera-Lluna, A., and Yanes, O. (2016). eRah: A Computational Tool Integrating Spectral Deconvolution and Alignment with Quantification and Identification of Metabolites in GC/MS-Based Metabolomics. *Anal. Chem.*, 88(19):9821–9829.

Du, X. and Zeisel, S. H. (2013). SPECTRAL DECONVOLUTION FOR GAS CHROMATOGRAPHY MASS SPECTROMETRY-BASED METABOLOMICS: CURRENT STATUS AND FUTURE PERSPECTIVES. *Computational and Structural Biotechnology Journal*, 4(5):1–10.

Franceschi, P., Masuero, D., Vrhovsek, U., Mattivi, F., and Wehrens, R. (2012). A benchmark spike-in data set for biomarker identification in metabolomics. *J. Chemometrics*, 26(1-2):16–24.

Fu, H.-Y., Hu, O., Zhang, Y.-M., Zhang, L., Song, J.-J., Lu, P., Zheng, Q.-X., Liu, P.-P., Chen, Q.-S., Wang, B., Wang, X.-Y., Han, L., and Yu, Y.-J. (2017). Mass-spectra-based peak alignment for automatic nontargeted metabolic profiling analysis for biomarker screening in plant samples. *Journal of Chromatography A*, 1513(Supplement C):201–209.

Guitton, Y., Tremblay-Franco, M., Le Corguillé, G., Martin, J.-F., Pétéra, M., Roger-Mele, P., Delabrière, A., Goulitquer, S., Monsoor, M., Duperier, C., Canlet, C., Servien, R., Tardivel, P., Caron, C., Giacomoni, F., and Thévenot, E. A. (2017). Create, run, share, publish, and reference your LC–MS, FIA–MS, GC–MS, and NMR data analysis workflows with the Workflow4Metabolomics 3.0 Galaxy online infrastructure for metabolomics. *The International Journal of Biochemistry & Cell Biology*, 93(Supplement C):89–101.

Haug, K., Salek, R. M., and Steinbeck, C. (2017). Global open data management in metabolomics. *Current Opinion in Chemical Biology*, 36:58–63.

Itoi, T., Sugimoto, M., Umeda, J., Sofuni, A., Tsuchiya, T., Tsuji, S., Tanaka, R., Tonozuka, R., Honjo, M., Moriyasu, F., Kasuya, K., Nagakawa, Y., Abe, Y., Takano, K., Kawachi, S., Shimazu, M., Soga, T., Tomita, M., and Sunamura, M. (2017). Serum Metabolomic Profiles for Human Pancreatic Cancer Discrimination. *Int. J. Mol. Sci.*, 18(4):767.

Jorge, T. F., Mata, A. T., and António, C. (2016). Mass spectrometry as a quantitative tool in plant metabolomics. *Phil. Trans. R. Soc. A*, 374(2079):20150370.

Jr, S. S., Mehrmohamadi, M., Liberti, M. V., Wan, M., Wells, M. T., Booth, J. G., and Locasale, J. W. (2017). RRmix: A method for simultaneous batch effect correction and analysis of metabolomics data in the absence of internal standards. *PLOS ONE*, 12(6):e0179530.

Kapoore, R. V. and Vaidyanathan, S. (2016). Towards quantitative mass spectrometry-based metabolomics in microbial and mammalian systems. *Phil. Trans. R. Soc. A*, 374(2079):20150363.

Karpievitch, Y. V., Nikolic, S. B., Wilson, R., Sharman, J. E., and Edwards, L. M. (2014). Metabolomics Data Normalization with EigenMS. *PLOS ONE*, 9(12):e116221.

Koelmel, J. P., Kroeger, N. M., Ulmer, C. Z., Bowden, J. A., Patterson, R. E., Cochran, J. A., Beecher, C. W. W., Garrett, T. J., and Yost, R. A. (2017). LipidMatch: An automated workflow for rule-based lipid identification using untargeted high-resolution tandem mass spectrometry data. *BMC Bioinformatics*, 18:331.

Konz, T., Migliavacca, E., Dayon, L., Bowman, G., Oikonomidi, A., Popp, J., and Rezzi, S. (2017). ICP-MS/MS-Based Ionomics: A Validated Methodology to Investigate the Biological Variability of the Human Ionome. *J. Proteome Res.*, 16(5):2080–2090.

Kuhl, C., Tautenhahn, R., Böttcher, C., Larson, T. R., and Neumann, S. (2012). CAMERA: An Integrated Strategy for Compound Spectra Extraction and Annotation of Liquid Chromatography/Mass Spectrometry Data Sets. *Anal. Chem.*, 84(1):283–289.

Kuligowski, J., Sánchez-Illana, Á., Sanjuán-Herráez, D., Vento, M., and Quintás, G. (2015). Intra-batch effect correction in liquid chromatography-mass spectrometry using quality control samples and support vector regression (QC-SVRC). *Analyst*, 140(22):7810–7817.

Kusonmano, K., Vongsangnak, W., and Chumnanpuen, P. (2016). Informatics for Metabolomics. In *Translational Biomedical Informatics*, Advances in Experimental Medicine and Biology, pages 91–115. Springer, Singapore.

Leek, J. T., Johnson, W. E., Parker, H. S., Jaffe, A. E., and Storey, J. D. (2012). The sva package for removing batch effects and other unwanted variation in high-throughput experiments. *Bioinformatics*, 28(6):882–883.

Leek, J. T. and Storey, J. D. (2007). Capturing Heterogeneity in Gene Expression Studies by Surrogate Variable Analysis. *PLOS Genet*, 3(9):e161.

Leek, J. T. and Storey, J. D. (2008). A general framework for multiple testing dependence. *PNAS*, 105(48):18718–18723.

Li, B., Tang, J., Yang, Q., Li, S., Cui, X., Li, Y., Chen, Y., Xue, W., Li, X., and Zhu, F. (2017). NOREVA: Normalization and evaluation of MS-based metabolomics data. *Nucleic Acids Res*, 45(W1):W162–W170.

Li, L., Zhao, J., Zhao, Y., Lu, X., Zhou, Z., Zhao, C., and Xu, G. (2016). Comprehensive investigation of tobacco leaves during natural early senescence via multi-platform metabolomics analyses. *Sci Rep*, 6.

Lisec, J., Hoffmann, F., Schmitt, C., and Jaeger, C. (2016). Extending the Dynamic Range in Metabolomics Experiments by Automatic Correction of Peaks Exceeding the Detection Limit. *Anal. Chem.*, 88(15):7487–7492.

Livera, A. M. D., Sysi-Aho, M., Jacob, L., Gagnon-Bartsch, J. A., Castillo, S., Simpson, J. A., and Speed, T. P. (2015). Statistical Methods for Handling Unwanted Variation in Metabolomics Data. *Anal. Chem.*, 87(7):3606–3615.

Lu, W., Su, X., Klein, M. S., Lewis, I. A., Fiehn, O., and Rabinowitz, J. D. (2017). Metabolite Measurement: Pitfalls to Avoid and Practices to Follow. *Annu. Rev. Biochem.*, 86(1):277–304.

Lu, X. and Xu, G. (2008). LC-MS Metabonomics Methodology in Biomarker Discovery. In Wang, F., editor, *Biomarker Methods in Drug Discovery and Development*, Methods in Pharmacology and Toxicology™, pages 291–315. Humana Press.

Luo, X. and Li, L. (2017). Metabolomics of Small Numbers of Cells: Metabolomic Profiling of 100, 1000, and 10000 Human Breast Cancer Cells. *Anal. Chem.*, 89(21):11664–11671.

Matsuo, T., Tsugawa, H., Miyagawa, H., and Fukusaki, E. (2017). Integrated Strategy for Unknown EI–MS Identification Using Quality Control Calibration Curve, Multivariate Analysis, EI–MS Spectral Database, and Retention Index Prediction. *Anal. Chem.*, 89(12):6766–6773.

Misra, B. B. and van der Hooft, J. J. J. (2016). Updates in metabolomics tools and resources: 2014–2015. *ELECTROPHORESIS*, 37(1):86–110.

Najdekr, L., Friedecký, D., Tautenhahn, R., Pluskal, T., Wang, J., Huang, Y., and Adam, T. (2016). Influence of Mass Resolving Power in Orbital Ion-Trap Mass Spectrometry-Based Metabolomics. *Anal. Chem.*, 88(23):11429–11435.

Ni, Y., Su, M., Qiu, Y., Jia, W., and Du, X. (2016). ADAP-GC 3.0: Improved Peak Detection and Deconvolution of Co-eluting Metabolites from GC/TOF-MS Data for Metabolomics Studies. *Anal. Chem.*, 88(17):8802–8811.

Qiu, F., Fine, D. D., Wherritt, D. J., Lei, Z., and Sumner, L. W. (2016). PlantMAT: A Metabolomics Tool for Predicting the Specialized Metabolic Potential of a System and for Large-Scale Metabolite Identifications. *Anal. Chem.*, 88(23):11373–11383.

Robbat Jr., A., Kfoury, N., Baydakov, E., and Gankin, Y. (2017). Optimizing targeted/untargeted metabolomics by automating gas chromatography/mass spectrometry workflows. *Journal of Chromatography A*, 1505:96–105.

Schrimpe-Rutledge, A. C., Codreanu, S. G., Sherrod, S. D., and McLean, J. A. (2016). Untargeted Metabolomics Strategies—Challenges and Emerging Directions. *J. Am. Soc. Mass Spectrom.*, 27(12):1897–1905.

Sitnikov, D. G., Monnin, C. S., and Vuckovic, D. (2016). Systematic Assessment of Seven Solvent and Solid-Phase Extraction Methods for Metabolomics Analysis of Human Plasma by LC-MS. *Sci Rep*, 6.

Sumner, L. W., Amberg, A., Barrett, D., Beale, M. H., Beger, R., Daykin, C. A., Fan, T. W.-M., Fiehn, O., Goodacre, R., Griffin, J. L., Hankemeier, T., Hardy, N., Harnly, J., Higashi, R., Kopka, J., Lane, A. N., Lindon, J. C., Marriott, P., Nicholls, A. W., Reily, M. D., Thaden, J. J., and Viant, M. R. (2007). Proposed minimum reporting standards for chemical analysis Chemical Analysis Working Group (CAWG) Metabolomics Standards Initiative (MSI). *Metabolomics*, 3(3):211–221.

Tautenhahn, R., Böttcher, C., and Neumann, S. (2008). Highly sensitive feature detection for high resolution LC/MS. *BMC Bioinformatics*, 9:504.

Thonusin, C., IglayReger, H. B., Soni, T., Rothberg, A. E., Burant, C. F., and Evans, C. R. (2017). Evaluation of intensity drift correction strategies using MetaboDrift, a normalization tool for multi-batch metabolomics data. *Journal of Chromatography A*, 1523(Supplement C):265–274.

Tian, T.-F., Wang, S.-Y., Kuo, T.-C., Tan, C.-E., Chen, G.-Y., Kuo, C.-H., Chen, C.-H. S., Chan, C.-C., Lin, O. A., and Tseng, Y. J. (2016). Web Server for Peak Detection, Baseline Correction, and Alignment in Two-Dimensional Gas Chromatography Mass Spectrometry-Based Metabolomics Data. *Anal. Chem.*, 88(21):10395–10403.

Townsend, M. K., Aschard, H., De Vivo, I., Michels, K. B., and Kraft, P. (2016). Genomics, Telomere Length, Epigenetics, and Metabolomics in the Nurses' Health Studies. *Am J Public Health*, 106(9):1663–1668.

Treutler, H., Tsugawa, H., Porzel, A., Gorzolka, K., Tissier, A., Neumann, S., and Balcke, G. U. (2016). Discovering Regulated Metabolite Families in Untargeted Metabolomics Studies. *Anal. Chem.*, 88(16):8082–8090.

Uppal, K., Walker, D. I., and Jones, D. P. (2017). xMSannotator: An R Package for Network-Based Annotation of High-Resolution Metabolomics Data. *Anal. Chem.*, 89(2):1063–1067.

Viant, M. R., Kurland, I. J., Jones, M. R., and Dunn, W. B. (2017). How close are we to complete annotation of metabolomes? *Current Opinion in Chemical Biology*, 36:64–69.

Wang, S.-Y., Kuo, C.-H., and Tseng, Y. J. (2013). Batch Normalizer: A Fast Total Abundance Regression Calibration Method to Simultaneously Adjust Batch and Injection Order Effects in Liquid Chromatography/Time-of-Flight Mass Spectrometry-Based Metabolomics Data and Comparison with Current Calibration Methods. *Anal. Chem.*, 85(2):1037–1046.

Watrous, J. D., Henglin, M., Claggett, B., Lehmann, K. A., Larson, M. G., Cheng, S., and Jain, M. (2017). Visualization, Quantification, and Alignment of Spectral Drift in Population Scale Untargeted Metabolomics Data. *Anal. Chem.*, 89(3):1399–1404.

Wu, Y. and Li, L. (2016). Sample normalization methods in quantitative metabolomics. *Journal of Chromatography A*, 1430:80–95.

Yamamoto, H., Fujimori, T., Sato, H., Ishikawa, G., Kami, K., and Ohashi, Y. (2014). Statistical hypothesis testing of factor loading in principal component analysis and its application to metabolite set enrichment analysis. *BMC Bioinformatics*, 15:51.

Yang, Q., Lin, S.-S., Yang, J.-T., Tang, L.-J., and Yu, R.-Q. (2017). Detection of inborn errors of metabolism utilizing GC-MS urinary metabolomics coupled with a modified orthogonal partial least squares discriminant analysis. *Talanta*, 165:545–552.

Zampieri, M., Sekar, K., Zamboni, N., and Sauer, U. (2017). Frontiers of high-throughput metabolomics. *Current Opinion in Chemical Biology*, 36:15–23.