XCMS 包用法

1. 前期准备

数据类型：NetCDF， mzXML， mzData 。 所以首先需要把自己的文件通过相应的软件转化成这类文件。

数据位置：因为 xcms 会记录下数据所在的位置，并且在数据分析过程中会来回从文件夹中读取数据。所以不要再随意改变数据所在的文件的位置。

数据来源：来源不同的数据应当分开进行数据前处理。比如正离子和负离子模式下得到的数据，不同的洗脱梯度下得到的数据。

数据存储：xcms 可以根据数据所在的文件夹的来识别不同的数据组。比如你想研究某一药物在两组病人间的纵向效应（longitudinaleffect），然后你可以把你的两组数据分别存入命名为 group A 和 groupB 的文件夹，这样 xcms 可以识别这两个组。每个文件夹内你还可以继续细分，比如按时间‘day1’ ， ‘day2’ 等等。xcms 会自动给这些数据命名为 groupA/day1， group A/day 2， 等等。 （所以这里你要根据你的数据组来把它们存入不同的文件夹）。*如果这里描述的不清楚的话，我会在后面的例子里进一步说明。*

下面将会以一个详细的例子来分布解说 xcms 是如何处理 LC-MS 数据的。

2. 数据分析

2.1 文件读取

```
cdfpath<-system.file("cdf", package = "faahKO")
```

system.file 作用是寻找包里面文件的路径和全名。这里指的是在叫‘faahKO’的包中找到文件类型为‘cdf’的文件的全名。

```
list.files(cdfpath,recursive = TRUE)
```

[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"

list.files 是读取该路径下的文件夹或文件名，并以字符型向量存储。

当然，这两条代码对我们都不重要，它们只是单纯的为了从这个包里面读取数据。

2.2 过滤及谱峰检测（filtration and peak identification）

Library（xcms）*每次使用这个包之前先要加载。*

cdffiles<- list.files(cdfpath, recursive = TRUE, full.names = TRUE)

读取 cdf 文件，这里如果你要读取你自己的文件，你只要把 cdfpath 换成你的文件夹所在的位置就行，比如你的数据在文件 D:/data,那你把这里的代码换成 cdffiles<- list.files('D:/data, recursive = TRUE, full.names = TRUE) 就行（当然还有更方便的方法，不过这个就够用了）。Recursive=TURE 的话，它会递归读取到你文件夹里，如果是 False 的话就只会读取到文件夹。Full.names=TURE 的话，你会得到一个包含路径的文件名，False 的话只会得到文件名。

[1] "C:/Softwares/R 3.0.1/R-3.0.1/library/faahKO/cdf/KO/ko15.CDF"

[2] "C:/Softwares/R 3.0.1/R-3.0.1/library/faahKO/cdf/KO/ko16.CDF"

[3] "C:/Softwares/R 3.0.1/R-3.0.1/library/faahKO/cdf/KO/ko18.CDF"

[4] "C:/Softwares/R 3.0.1/R-3.0.1/library/faahKO/cdf/KO/ko19.CDF"

[5] "C:/Softwares/R 3.0.1/R-3.0.1/library/faahKO/cdf/KO/ko21.CDF"

[6] "C:/Softwares/R 3.0.1/R-3.0.1/library/faahKO/cdf/KO/ko22.CDF"

[7] "C:/Softwares/R 3.0.1/R-3.0.1/library/faahKO/cdf/WT/wt15.CDF"

[8] "C:/Softwares/R 3.0.1/R-3.0.1/library/faahKO/cdf/WT/wt16.CDF"

[9] "C:/Softwares/R 3.0.1/R-3.0.1/library/faahKO/cdf/WT/wt18.CDF"

[10] "C:/Softwares/R 3.0.1/R-3.0.1/library/faahKO/cdf/WT/wt19.CDF"

[11] "C:/Softwares/R 3.0.1/R-3.0.1/library/faahKO/cdf/WT/wt21.CDF"

[12] "C:/Softwares/R 3.0.1/R-3.0.1/library/faahKO/cdf/WT/wt22.CDF"

xset<- xcmsSet(cdffiles)

这条语句主要作用是谱峰鉴定（peak identification），其结果存储在了一个‘xcmsset’

类型的数据（不用管这个类型是啥意思）。

250:38 300:103 350:226 400:338 450:431 500:529 550:674 600:847 250:43 300:128

350:275 400:394 450:500 500:637 550:835 600:1027 250:25 300:93 350:227 400:337

450:411 500:498 550:640 600:758 250:19 300:67 350:169 400:258 450:301 500:373

550:488 600:580 250:24 300:60 350:166 400:254 450:315 500:391 550:501

600:582 250:31 300:71 350:183 400:280 450:338 500:422 550:532 600:604 250:41

300:105 350:212 400:319 450:416 500:533 550:684 600:838 250:27 300:107 350:232

400:347 450:440 500:549 550:712 600:905 250:24 300:87 350:200 400:293 450:351

500:426 550:548 600:661 250:22 300:65 350:161 400:243 450:293 500:358 550:483

600:561 250:28 300:69 350:157 400:229 450:282 500:364 550:493 600:592 250:30

300:81 350:188 400:280 450:356 500:473 550:618 600:765

每一行就是一个文件，并且谱峰鉴定结果用成对的数据形式来给出。分号前面的数字表示正

在处理的质荷比（m/z），后面的数字表示到该质荷比时已经找出的峰数目（peaknumber）

这个语句看似简单，里面包含的参数却很多：

xcmsSet(files = NULL, snames = NULL, sclass = NULL, phenoData = NULL,

profmethod = "bin", profparam = list(), polarity = NULL, lockMassFreq=FALSE,

mslevel=NULL, nSlaves=0, progressCallback=NULL, scanrange = NULL, ...) 多数情况下 ,我们用这些默认值就够了。但对于特定的分析仪器或者数据 ,我们也需要优化一些参数。

Findpeaks 利用两种不同的算法来进行峰值检测（peak detection）。其中默认法是findPeaks.matchedFilter，另一种方法是findPeaks.centWave。

2.2.1 findPeaks.matchedFilter

该法有几个参数需要考虑：

峰宽（peak width）可用标准方差(sigma) 或者半峰全宽(fwhm)来表示，默认值是FWHM=30 s。根据色谱类型，我们应当选择合适的峰宽。

步长（step）默认值是 2，step=2.

存储，有四种方法， ‘bin’ ， ‘binlin‘ ， ‘binlinbase’ ， ‘intlin’ ，其中 bin 是默认方法。四种方法的具体意思自己看文献吧，最后一个不推荐用，第三个具体怎么设置，我也没咋看懂。

2.2.2 findPeaks.centWave

该法更适合于高分辨率的仪器下的 centroid mode 的数据， 比如LC/{TOF,OrbiTrap,FTICR}-MS。binning 在这里是不必要的。这里面有两个参数需要考虑：

ppm，其选择和仪器精度有关

峰宽范围（peak width range）：比如 HPLC 里用 peakwidth=c(20,50) ，UPLC 里是peakwidth=c(5，12)， 单位是秒。

说了这个多了，举例子吧（英语说明里这部分一笔带过，而且 help 里的例子也不适合batchfiles，如何设置这些参数的确让人有些摸不着头脑）：

1. findPeaks.matchedFilter

xset<- xcmsSet(cdffiles, method=’ matchedFilter’ ,fwhm=60, step=3,

profmethod='binlin')

xset

An"xcmsSet" object with 12 samples

Timerange: 2507.6-4139.9 seconds (41.8-69 minutes)

Massrange: 200.1-597.0233 m/z

Peaks:2549 (about 212 per sample)

PeakGroups: 0

Sampleclasses: KO, WT

Profilesettings: method = binlin

step = 3

Memoryusage: 0.497 MB

最后结果如上，我用了 matchedFilter 方法，fwhm 是 60s，步长是 3，存储方法是 binlin

2. findPeaks.centWave

xset<-xcmsSet(cdffiles, method=’ centWave’ , ppm=5, peakwidth=c(10,20))

xset

An"xcmsSet" object with 12 samples

Timerange: 2502.9-4476.4 seconds (41.7-74.6 minutes)

Massrange: 200.1-600 m/z

Peaks:52907 (about 4409 per sample)

PeakGroups: 0

Sampleclasses: KO, WT

Profilesettings: method = bin

step = 0.1

Memoryusage: 4.68 MB

最后结果如上，我用了 centWave 方法，ppm=5，峰宽范围是 10-20s。

2.3 样本间峰匹配 （peak match across samples）

xset<- group(xset)

Group 有三种方法‘density’，‘mzClust’和‘nearest’。每种方法下都有一系列不同的参数。‘density’是默认方法。nearest’需要额外安装‘RANN’的包 才能实现。

Density：Group peaks together across samples using overlapping m/z bins andcalculation of smoothed peak distributions in chromatographic time。

mzClust：Runs high resolution alignment onsingle spectra samples stored in a given xcmsSet.

Nearest: Group peaks together across samples by creating a master peaklist and assigning corresponding peaks from all samples.

具体的方法和参数自己可以用 help。比如想查 density 方法，就用 help(‘group.density’)来查看。

举个例子：

Xset<-group(xset, mzppm = 10, mzabs= 0, minsamp = 1, minfrac=0)

这里我用了 mzCluster，它对应的各个参数跟在后面。

2.4 保留时间校正 （retention time correction）

样本间峰匹配分组后，xcms 便通过这些分组来确定和校正每次运行之间保留时间的漂移。

峰对其后 xcms 再进行一次分组，以提高分组的精确性。并非所有的分组都适合用来做保留时间的校正，比如有很多缺失峰的组和来自于同一样品，但却有多条峰的组。

xset2 <- retcor(xset,family = "symmetric", plottype = "mdevden")

这里面又是一大堆参数

Rector 也有三种方法‘loess’,’ obiwarp’,’ peakgroups’，其中‘loess’是默认方法。

Loess &peakgroups：Use smoothed deviations to alignretention times. 这两个方法竟然完全一样，参数也一样。

Obiwarp：Calculate retention time deviationsfor each sample. It is based on the code at http://obi-warp.sourceforge.net/. However, this function is ableto align multiple samples, by a center-star strategy.

每个方法后面又是跟了很多参数。具体还是通过 help 来查看。比如想看 loess，就用 help（’ retcor.loess’ ）

保留时间校正后，xcms 又进行了一次分组，方法同上，不再细讲。

xset2 <- group(xset2,bw = 10)

2.5Filling in Missing Peak Data

即使再次精确分组，还会存在有些组有缺失峰。

xset3 <-fillPeaks(xset2)

这里有两种方法，‘chrom’和‘MSW’，其中‘chrom’是默认法。

‘chrom’法：Integrateareas of missing peaks

它有一个参数‘nSlaves’ : number of slaves/cores to be used for parallel peak filling. MPI is used if installed,otherwise the snow package is employed for multicore support.

‘**MSW**’法：Integrateareas of missing peaks in FTICR-MS data