应用简报

代谢组学/天然产物鉴定

Agilent

Trusted Answers

基于 XCMS 的天然产物分析流程：数据的提取、对齐、特征化、归组及化合物鉴定

作者

于擎

安捷伦科技（中国）有限公司

摘要

本应用简报介绍了如何使用 XCMS 及其他一系列基于 R 的开源包处理 Agilent 6500 系列 LC-Q-TOF 通过 auto MS/MS 模式集采到的数据。本工作使用 XCMS 和 MsFeatures 进行了色谱峰提取、色谱峰优化、保留时间对齐、形成特征以及特征归组；使用 Spectral 进行了二级谱图的提取和聚合；使用 MsBackendMsp 和 MetaboAnnotation 进行了第三方谱库的导入和基于二级谱图匹配的化合物鉴定，展示了完整的工作流程供读者参考。最终的数据处理结果，既可以导出为 csv 格式表格进而通过 MPP 进行统计学分析，也可以生成其他特定格式供第三方软件进行分析。

前言

随着液质联用 (LC/MS) 技术的高速发展和普及，当前研究人员广泛使用 LC/MS 技术对天然产物进行分析和鉴定。与常规的 MS + target MS/MS 相比，auto MS/MS 采集能够同时采集一级和二级质谱数据，有效提高数据采集效率。利用 auto MS/MS 采集到庞大的数据后，需要借助功能强大的软件来实施特征提取和化合物鉴定。XCMS[1-3] 是目前比较受欢迎的代谢组学分析软件，具有特征提取算法先进和占用系统资源少等特点，已经广泛用于液质和气质联用数据分析。目前除在线版 XCMS (Pro) 以外，Bioconductor 上提供了基于 R 语言的版本供研究人员使用（免费且无任何限制）。XCMS 自带官方教程可供学习。但由于 XCMS 本质上只能完成特征提取相关工作，因此在整个分析流程中还需要采用其他开源包。

本研究将以四个不同产地的某植物提取物（样品提供方要求对详情保密）为实验对象，使用 auto MS/MS 技术对其进行数据采集，然后利用 XCMS 及相关开源包对数据进行色谱峰提取、色谱峰优化、保留时间对齐、形成特征、特征归组和聚合、谱图提取、谱图聚合、化合物鉴定和定量数据生成等，为相关研究人员展示完整的工作流程。

实验部分

试剂和样品

乙腈为液质级，购于 Merck；甲酸为液质级，购于 Sigma；所用实验用水为 Millipore Milli-Q 超纯水系统现制备的高纯去离子水；四个不同产地的植物提取物由合作用户提供。

仪器和设备

采用 Agilent 1290 Infinity II UHPLC 与 Agilent 6545 LC/Q-TOF 联用系统，其中 LC/Q-TOF 配备安捷伦双喷射流离子源。

液相色谱条件

色谱柱：	Waters ACQUITY UPLC HSS T3 (2.7 µm, 2.1 × 100 mm)
流动相 A：	水（含 10 mmol/L 乙酸铵和 0.05% 甲酸）
流动相 B：	90% 乙腈/10% 水（含 10 mmol/L 乙酸铵和 0.05% 甲酸）
流速：	0.3 mL/min
柱温：	40 °C

梯度程序：

时间 (min)	B (%)
0	1
3	1
28	70
30	90
31	99
32	99

质谱条件

干燥气温度：	280 °C
干燥气流速：	7 L/min
雾化器压力：	35 psi
鞘流气温度：	325 °C
鞘流气流速：	12 L/min
喷嘴电压：	0 V
毛细管电压：	3500 V
碎裂电压：	125 V
MS 采集速率：	6 幅谱图/秒
MS2 采集速率：	4 幅谱图/秒
质量范围：	m/z 100–1000 (MS1)；m/z 50–1000 (MS2)
分离峰窗口：	窄，约 1.3 m/z
碰撞能量：	25 eV
每个循环最大母离子数：	6
MS/MS 阈值：	5000 响应值和 0.001%
基于母离子丰度的扫描速度：	40000 响应值/质谱图
剔除未达目标 TIC 的母离子：	否
启用主动排除：	是，重复一次，然后排除 0.2 min
纯度严格性：	70%，截留 0%
同位素模型：	常见有机分子
母离子排序：	1 价
静态排除范围：	m/z 50–106，900–1000

开源包准备工作

在分析流程中用到以下包：XCMS、Spectra[4] 、MsFeatures[5]、MsBackendMsp[6]、MetaboAnnotation[7]、QFeatures[8] 以及用于数据操作的 tidyverse[9]；另外，可以使用 RColorBrewer 包[10] 中的各种调色盘进行配色。

安装来自 Bioconductor 的开源包：

```
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install(c("xcms", "Spectra", "MsFeatures", "MsBackendMsp",
"MetaboAnnotation", "QFeatures"))
```

安装 tidyverse 包：

```
install.packages("tidyverse")
```

加载必要的包：

```
library(xcms)
library(MsFeatures)
library(QFeatures)
library(Spectra)
library(tidyverse)
```

为提高分析速度，Bioconductor 上的开源包默认使用并行运算进行数据分析。但是这在 Windows 系统中可能面临一些问题，导致某些函数无法正常工作。在优化色谱峰和特征归组过程中，并行运算可能导致报错。因此，可以使用 `register(SerialParam())` 取消并行运算以免报错。恢复并行运算的方法如下：检测 CPU 核心数（不包含逻辑核心）：`parallel::detectCores(logical = FALSE)`，添加用于并行运算的核心（例如，本机的物理核心为 4，可以预留一个核心用于其他工作）：`register(bpstart(SnowParam(3)))`。

```
register(SerialParam())
```

或者，可以由 R 根据操作系统自动选择如何进行并行运算（如果仍然出现错误，则可以使用上述方法取消并行运算）：

```
if (.Platform$OS.type == "unix") {
    register(bpstart(MulticoreParam(3)))
} else {
    register(bpstart(SnowParam(3)))
}
```

结果与讨论

使用 XCMS 对 auto MSMS 数据进行提取和分析

auto MSMS 数据的导入和读取

本文将以来自四个不同产地的某植物提取物的 auto MS/MS 数据为例，详细介绍如何使用 XCMS 进行数据导入。

首先创建数据（已经使用 MSConvert 将数据格式转换为 mzML），主要包括以下操作：

1. 创建数据的原始路径以及数据的分组信息
2. 将数据读入

```
raw_file <- fs::dir_ls(path = "raw_data", recurse = T, glob = "*.mzML")
# 获取原始数据的路径
sample_name <- str_extract(basename(raw_file), pattern = ".*(?=\\.)")
sample_group <- c(rep("QC", 3),
                  rep("OriginA", 4),
                  rep("OriginB", 4),
                  rep("OriginC", 4),
                  rep("OriginD", 4))
# 用于展示的数据中包含五组信息，3 个 QC 样品，另外四组样品数据，每组样品数据中包含 4 个生物学重复
pd <- data.frame(sample_name, sample_group)
raw_data <- readMSData(raw_file,
                       pdata = new("AnnotatedDataFrame", pd),
                       mode = "onDisk")
raw_data

## MSn experiment data ("OnDiskMSnExp")
## Object size in memory: 78.37 Mb
## - - - Spectra data - - -
##  MS level(s): 1 2
##  Number of spectra: 235883
##  MSn retention times: 0:02 - 32:00 minutes
## - - - Processing information - - -
## Data loaded [Tue Sep  5 14:38:36 2023]
##  MSnbase version: 2.27.1
## - - - Meta data  - - -
## phenoData
##   rowNames: 1 2 ... 19 (19 total)
##   varLabels: sample_name sample_group
##   varMetadata: labelDescription
## Loaded from:
##   [1] QC1.mzML...  [19] Sam4-4.mzML
##   Use 'fileNames(.)' to see all files.
## protocolData: none
## featureData
##   featureNames: F01.S00001 F01.S00002 ... F19.S12393 (235883 total)
##   fvarLabels: fileIdx spIdx ... spectrum (35 total)
##   fvarMetadata: labelDescription
## experimentData: use 'experimentData(object)'
```

可以利用 `fData()` 查看原始数据中的内容（见表 1）。限于篇幅，表 1 仅展示了部分信息。

表 1. 原始数据信息

	msLevel	polarity	original Peaks Count	totIon Current	base Peak Intensity
F01.S00001	1	1	186	175811.344	16355.487
F01.S00002	1	1	188	186688.516	18671.512
F01.S00003	1	1	173	169482.125	19210.144
F01.S00004	2	1	5	2911.012	1884.461
F01.S00005	2	1	8	8779.152	7847.838
F01.S00006	2	1	6	6406.985	3044.687

MS Level：质谱级别
Polarity：极性
Original Peaks Count：原始峰计数
TotIon Current：总离子流
Base Peak Intensity：基峰强度

从行名可以看出，原始数据以扫描数为行名进行储存（例如，F01 指第一个文件，S00001 指第一次扫描）；列名则代表液质数据中受到关注的各种项目，例如 TIC 强度、母离子选择的窗口宽度和 MS 级别等。

用已知化合物或基质添加化合物对数据进行简单评估（在 MassHunter 上更直观）

查看一种结构已知的化合物（酪氨酸，tyrosine）的色谱峰（见图 1）：

```
library(RColorBrewer)
group_colors <- brewer.pal(5, "Spectral")
names(group_colors) <- c("QC", "OriginA", "OriginB", "OriginC", "OriginD")

raw_data %>%
  filterRt(rt = c(96, 114)) %>%
  filterMz(mz = c(182.07, 182.09)) %>%
  chromatogram(aggregationFun = "max", include = "none") %>%
  plot(col = group_colors[app_data$sample_group], lwd = 2)
```

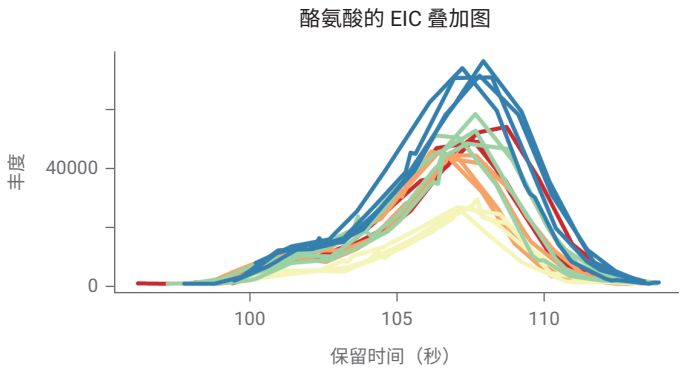

图 1. 酪氨酸的 EIC 叠加图

图 1 中显示了所有样品中的酪氨酸 EIC 叠加图。还可以分别提取数据中的此化合物的色谱图以及有关 m/z 随保留时间的变化的信息（结果如图 2 所示）：

```
raw_data %>%
  filterRt(rt = c(96, 114)) %>%
  filterMz(mz = c(182.07, 182.09)) %>%
  plot(type = "XIC")
```

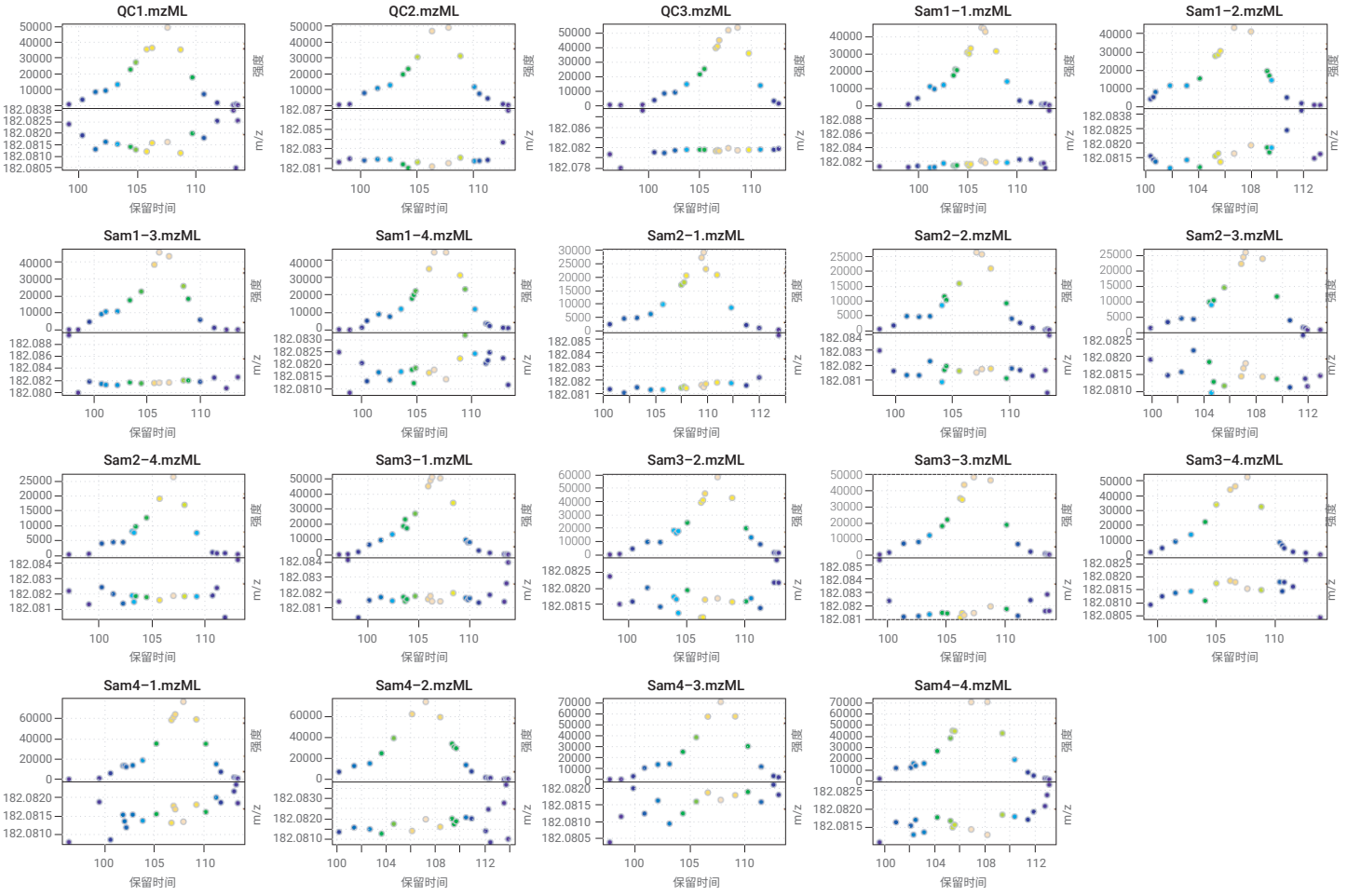

图 2. 酪氨酸在每个数据中的 EIC 及 m/z 随保留时间的变化

图 2 上半部分显示了酪氨酸色谱峰的强度随保留时间的变化，下半部分显示了 m/z 随保留时间的变化。这些图形都可以用于评估仪器状态和优化分析参数。

查找色谱峰

读取原始数据后，XCMS 会从查找色谱峰开始进行数据预处理。这个过程很类似于对单个数据文件进行 MFE 的过程。可利用 `findChromPeaks()` 函数识别色谱峰，进行最初的数据提取。`findChromPeaks()` 函数使用的算法为 `centWave`，因此首先需要对 `centWave` 参数进行设置：

```
cwp <- CentWaveParam(ppm = 10, peakwidth = c(10, 40), noise = 5000)
```

`centWave` 参数的含义及初始值如表 2 所示。如需进一步了解这些参数，请参考 `centWave` 参数。

表 2. 色谱峰提取参数

参数	描述	初始值
ppm	允许的 m/z 偏差	25
peakwidth	色谱峰的宽度范围（以秒为单位）	20–50 s
snthresh	信噪比阈值	10
prefilter	峰值定义：超过特定强度阈值 (I) 的数据点数量 (k)	k = 3，I = 100
mzdiff	可接受的两个相邻 m/z 的质量差	−0.001
noise	强度截止值，低于此值被视为仪器噪声	0
mzCenterFun	计算色谱峰 m/z 中心的函数	加权平均 (wMean)
ntegrate	峰定量积分方法：1：墨西哥帽小波过滤数据，2：真实数据	1
fitgauss	是否使用高斯分布使峰值参数化	FALSE
scanrange	选择扫描范围间隔内的色谱峰	numeric(0)

```
app_data <- findChromPeaks(raw_data, param = cwp)
app_data

## MSn experiment data ("XCMSnExp")
## Object size in memory: 78.38 Mb
## - - - Spectra data - - -
##  MS level(s): 1 2
##  Number of spectra: 235883
##  MSn retention times: 0:02 - 32:00 minutes
## - - - Processing information - - -
## Data loaded [Tue Sep  5 14:38:36 2023]
##  MSnbase version: 2.27.1
## - - - Meta data  - - -
## phenoData
##   rowNames: 1 2 ... 19 (19 total)
##   varLabels: sample_name sample_group
##   varMetadata: labelDescription
## Loaded from:
##   [1] QC1.mzML...  [19] Sam4-4.mzML
##   Use 'fileNames(.)' to see all files.
## protocolData: none
## featureData
##   featureNames: F01.S00001 F01.S00002 ... F19.S12393 (235883 total)
##   fvarLabels: fileIdx spIdx ... spectrum (35 total)
##   fvarMetadata: labelDescription
## experimentData: use 'experimentData(object)'
## - - - xcms preprocessing - - -
## Chromatographic peak detection:
##  method: centWave
##  22175 peaks identified in 19 samples.
##  On average 1167 chromatographic peaks per sample.
```

从表 3 中可以看到数据的前六行（除非另有说明，否则下文所有表格都只展示前六行），其中行名代表色谱峰，列名则指示液质数据所包含的一些常见信息。

表 3. 色谱峰提取后的数据

	mz	mzmin	mzmax	rt	rtmin	rtmax	into	intb	maxo	sn
CP00001	110.0089	110.0086	110.0091	45.245	42.676	55.154	91416.10	91300.16	30215.99	352
CP00002	124.0246	124.0240	124.0248	45.413	42.509	55.154	250158.92	249089.98	87715.78	386
CP00003	125.0260	125.0257	125.0264	45.413	43.178	48.060	35301.96	35280.48	13580.74	337
CP00004	126.0223	126.0218	126.0227	45.413	43.178	48.992	27125.26	27103.83	12938.46	391
CP00005	127.0242	127.0239	127.0244	45.413	43.345	48.060	36489.50	36485.26	13128.84	13128
CP00006	141.0397	141.0393	141.0400	45.245	42.676	50.427	90428.01	90396.17	36911.79	746

另外，也可以利用某个 QC 数据来查看色谱峰的分布，结果如图 3 所示。根据图 3，可以比较直观地考察色谱峰在保留时间和 m/z 二维平面上的分布，从而评估色谱方法是否合适。

```
plotChromPeaks(app_data, file = 3)
```

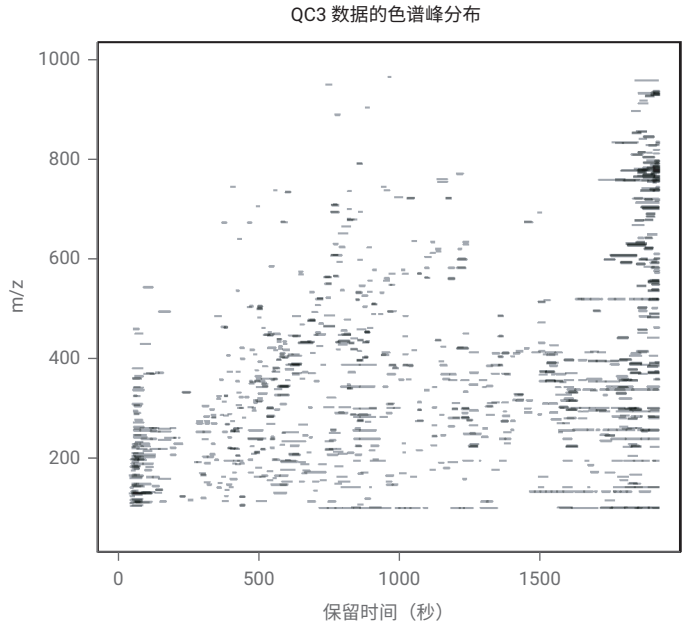

图 3. 数据 QC3 的 m/z 和保留时间的二维平面图

初步探索数据（建议使用带图形用户界面的 MassHunter 软件进行查看）

auto MS/MS 的数据不太适合直接用于观察 TIC 谱图的全貌，因此对根据一级数据绘制的原始 TIC 色谱图进行了重新筛选，所选保留时间范围为 500–1000 s（结果如图 4 所示）：

```
app_data %>%
  filterMsLevel(1L) %>%
  chromatogram(rt = c(500, 1000),
               aggregationFun = "sum",
               include = "none") %>%
  plot(col = group_colors[app_data$sample_group])
```

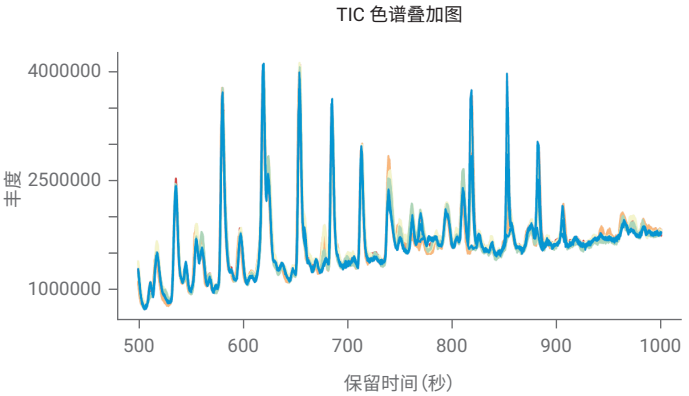

图 4. 未经保留时间对齐的 TIC 叠加图

色谱峰优化

在提取数据的过程中，经常会发现有些化合物分裂成两个色谱峰。除同分异构体的因素影响以外，还有一些化合物会在色谱柱上发生峰裂分；也有可能是色谱峰提取时出错。在 XCMS 中，可以利用 `refineChromPeaks()` 函数，按照一定的规则将这些色谱峰合并。有关如何设定峰合并的参数，最好根据先验的知识或通过在基质中添加标准品的方式来进行评估。

```
mpp <- MergeNeighboringPeaksParam(expandRt = 4, minProp = 0.75)
# 两峰连接处如果低于较小峰的 75% 则不合并。规则可以根据实际数据进行更改。
app_data <- refineChromPeaks(app_data, mpp) # 对裂分的色谱峰进行特征合并。
```

对齐

对齐在组学数据分析中非常重要，可避免产生大量缺失值。一般选用 QC 样本作为对齐的标准。对齐分两个步骤进行：首先，对数据进行分组（分为 QC 和研究 (study) 两组）；然后，按照 QC 组的保留时间对其他数据（研究组）进行保留时间对齐。

```
app_data$sample_type <- "study" # 先将所有的样品类型设置为"study"
app_data$sample_type[c(1, 2, 3)] <- "QC" # 设置 QC 组
pdp_subs <- PeakDensityParam(sampleGroups = app_data$sample_type,
                             minFraction = 0.9)

# 根据峰密度进行初始分组。使用 sample_type 作为分组变量。minFraction = 0.9 指至少在本组中出现 90% 以上的峰才能用于初始分组
app_data <- groupChromPeaks(app_data, param = pdp_subs) # 先进行特征的初始分组
pgd_subs <- PeakGroupsParam(minFraction = 0.85,
                            subset = which(app_data$sample_type == "QC"),
                            subsetAdjust = "average", span = 0.4) # 定义子集对齐参数，以 QC 为参考子集，对其他组别中的数据进行对齐。subsetAdjust 参数一共有两个值可用，"previous"和"average"，它们的具体含义可以参考 XCMS 官方教程
app_data <- adjustRtime(app_data, param = pgd_subs) # 使用设置好的子集对齐参数对保留时间进行对齐
```

粗略浏览对齐后的 TIC 色谱图，结果如图 5 所示。

```
app_data %>%
  filterMsLevel(1L) %>%
  chromatogram(rt = c(500, 1000),
               aggregationFun = "sum",
               include = "none") %>%
  plot(col = group_colors[app_data$sample_group])
```

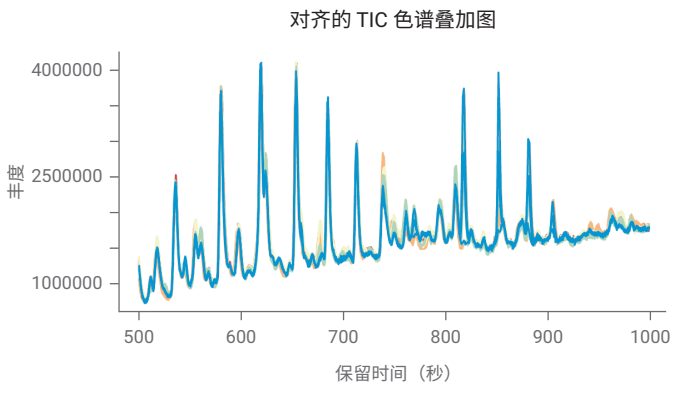

图 5. 保留时间对齐后的 TIC 叠加图

由于 Agilent 1290 Infinity II UHPLC 具有优异的稳定性，因此从图 5 中较难直观地看出保留时间的调整效果。但是可以通过展示调整保留时间的过程，来具体查看调整前后的变化（结果如图 6 所示）。

```
clrs <- rep("grey", 19)
clrs[app_data$sample_type == "QC"] = "#478058"
plotAdjustedRtime(app_data, col = clrs, peakGroupsPch = 1,
                  peakGroupsCol = "#F77E66")
```

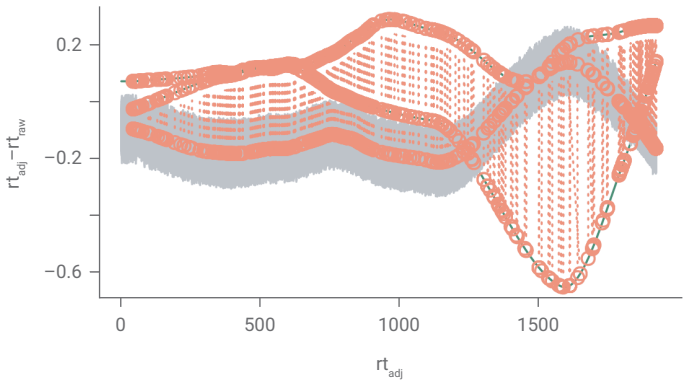

图 6. 保留时间调整曲线

从图 6 中可以清晰地看出，在整个保留时间范围内，调整后的保留时间与原始保留时间之间的差值始终较小。

一致化（形成特征）

将保留时间对齐后，需要对不同数据中代表同一离子的色谱峰进行整合，使其能够在后续分析中保持一致（即，无论它们出现在哪个样品中，均对应同一离子）。这也是一个从色谱峰到特征的过程，即，这些色谱峰要来源于同一离子。另外，需要注意的是每个样品数据中的色谱峰数量必须相等。这时存在一个问题，即，在有些样品中确实未检出这些化合物（尤其在 auto MS/MS 的采集中），此时相应化合物在这些样品中将呈现为缺失值。因此，最后需要填充上这些缺失值。

```
pdp <- PeakDensityParam(sampleGroups = app_data$sample_group,
                        minFraction = 0.4,
                        bw = 1.8)
```

其中 bw 是一个重要参数。通过查看酪氨酸色谱峰之间的相互关系，评估 bw 参数设置是否合理，如图 7 所示。

```
sample_colors <- group_colors[app_data$sample_group]

app_data %>%
  filterRt(rt = c(96, 114)) %>%
  filterMz(mz = c(182.07, 182.09)) %>%
  chromatogram(aggregationFun = "max") %>%
  plotChromPeakDensity(col = sample_colors,
                       param = pdp,
                       peakBg = sample_colors,
                       peakCol = sample_colors,
                       peakPch = 16)
```

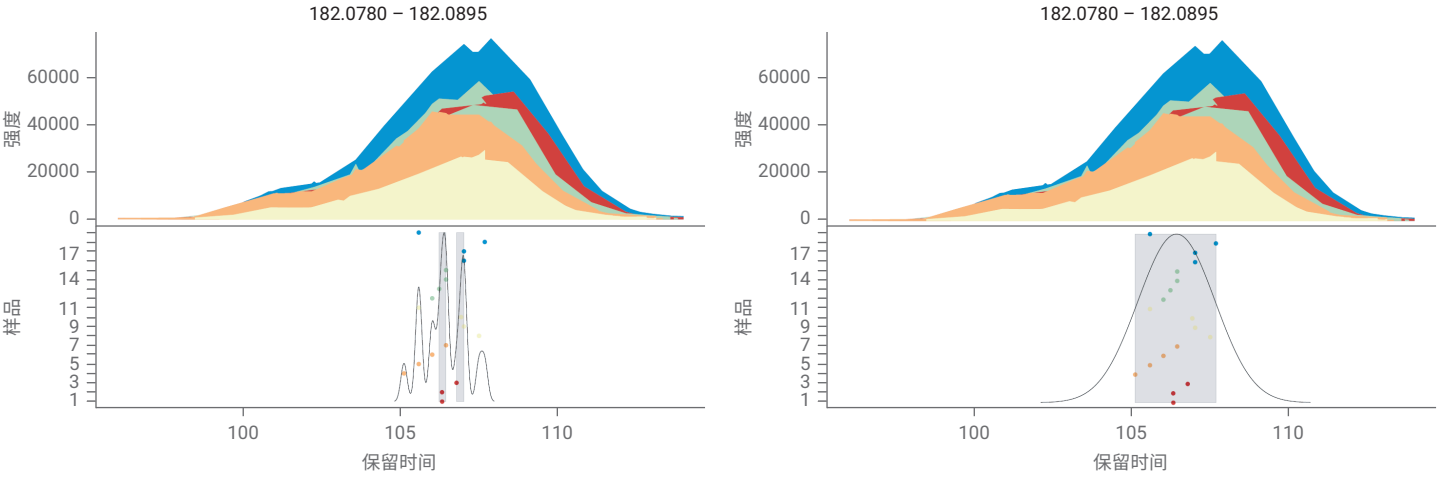

图 7. 酪氨酸的特征形成过程

从图 7 中可以看出，19 个峰对应于同一化合物（酪氨酸），但是 bw 过小（例如 0.1）时可能会出现不正常的峰裂分，如果过大则可能会将不相关的峰纳入特征中。可以使用更多化合物（例如脱氧海草素 (Deoxykhivorin)）来评估 bw 设置是否合理，但需要注意的是脱氧海草素在样品 Sam1-1 中缺失，结果如图 8 所示。

```
app_data %>%
  filterRt(rt = c(1775, 1833)) %>%
  filterMz(mz = c(593.26, 593.28)) %>%
  chromatogram(aggregationFun = "max") %>%
  plotChromPeakDensity(col = sample_colors[-4],
                       param = pdp,
                       peakBg = sample_colors[-4],
                       peakCol = sample_colors[-4],
                       peakPch = 16)
```

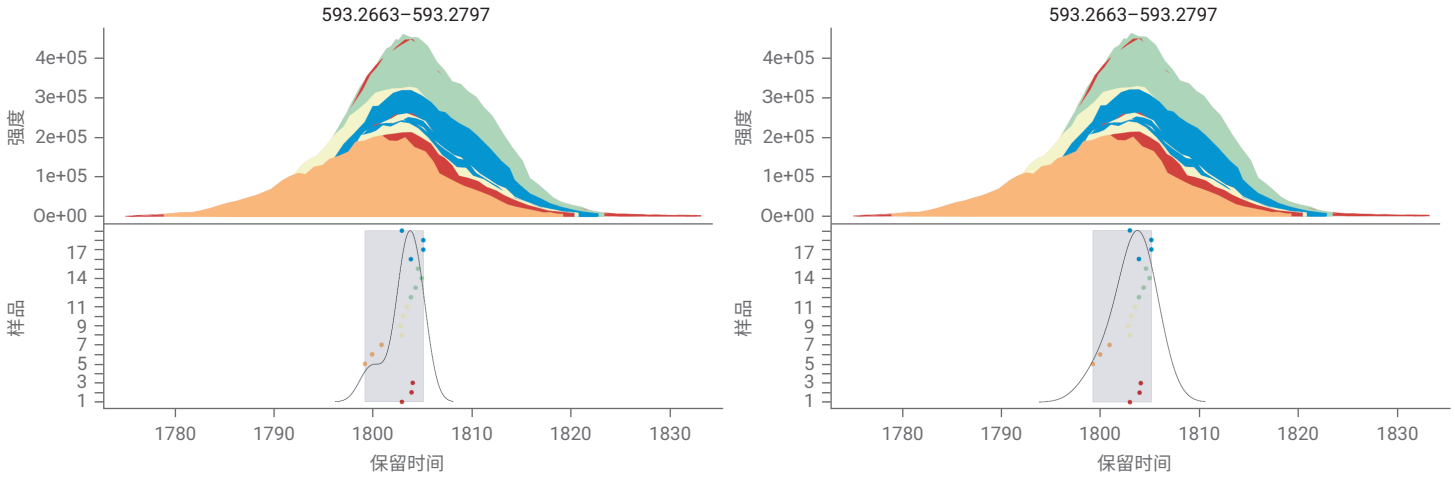

图 8. 脱氧海草素的特征形成过程

从图 8 中可以看出，bw = 1 对酪氨酸来说是合理的，但是对于脱氧海草素来说过小，因此将其对应的 bw 设置为 1.8（见图 8）。该化合物的峰出现分裂，因此需要使用多种化合物来评估 bw 的合理性。

接下来形成特征并进行缺失值填充：

```
app_data <- groupChromPeaks(app_data, param = pdp) # 形成特征
app_data <- fillChromPeaks(app_data, param = ChromPeakAreaParam()) # 缺失值填充
```

可以从特征的角度查看数据的提取情况，结果如图 9 所示。

```
app_data_feature <- featureChromatograms(app_data, features = c(203, 230,
1128, 1421))
plot(app_data_feature, col = sample_colors,
     peakBg = sample_colors[app_data_peaks[, "sample"]])
```

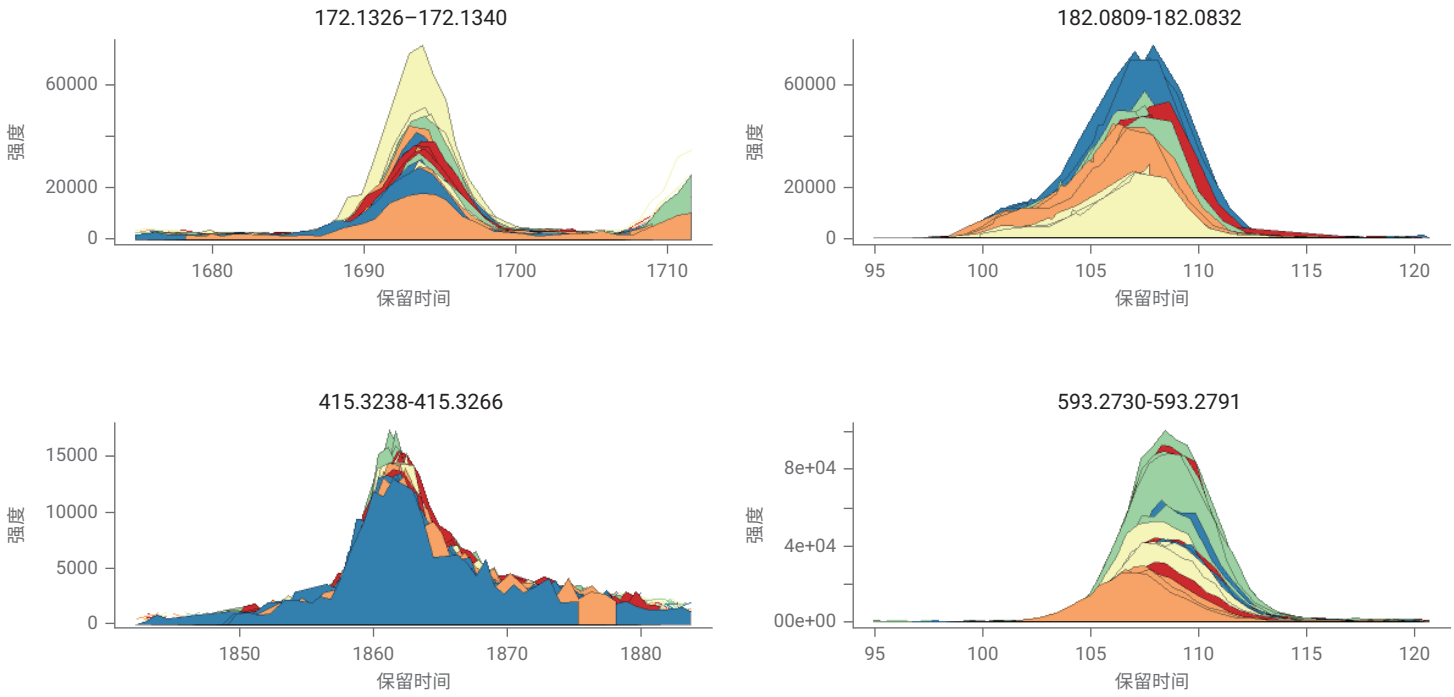

图 9. 查看部分特征

特征归组

所谓“特征归组”，是指将描述相同化合物的不同特征归为一个特征组的过程。这些不同的特征可能是由不同的加合物、中性丢失和同分异构体等因素造成的。特征归组功能由 `MsFeatures` 包完成。一般分成以下三步：

1. 按照保留时间相似度归组
2. 按照丰度相似度归组
3. 按照 EIC 相似度归组，由于这一步速度最慢，占用 CPU 和内存资源最多，所以应放在最后执行，其能够将误归组的特征剔除

```
app_data <- groupFeatures(app_data, param = SimilarRtimeParam(5)) # 保留时间差值 5 秒以内
app_data <- groupFeatures(app_data,
                          param = AbundanceSimilarityParam(threshold = 0.7,
                                                           transform = log2))
 # 相关系数阈值 0.7，丰度值已经经过 log2 转换
```

前两步相对比较粗略，可能将不同的特征错误归组，因此需要通过最后一步来细化：

```
app_data <- groupFeatures(app_data,
                                  EicSimilarityParam(threshold = 0.7, n = 3)) # 当 n = 3 时，将首先为每个特征组识别该组中所有特征的信号最高的 3 个样本，然后在每个样本中执行成对相似度计算。
```

特征归组的结果如表 4 所示（仅显示前六个）。

表 4. 特征归组结果

featureGroups.app_data.	Freq
FG.0001.001.001	2
FG.0001.002.001	1
FG.0002.001.001	1
FG.0002.002.001	1
FG.0003.001.001	1
FG.0003.002.001	1

featureGroups.app_data.：app_data 中的特征组名称

Freq：出现频次，即特征组包含的特征数

接下来可以进一步查看归组过程如何对原始特征进行进一步的归组，结果如图 9 所示。

```
features <- grep("FG.0088.001.001", featureGroups(app_data))
eics <- featureChromatograms(app_data,
                             features = features,
                             filled = T, n = 1)
cols <- c("#6CA6CD", "#B95756")
names(cols) <- unique(featureGroups(app_data)[features])
par(mfrow = c(1, 2))
plotChromatogramsOverlay(eics, col = cols[featureGroups(app_data)[features]],
                         lwd = 2, peakType = "none")
plotChromatogramsOverlay(normalize(eics),
                         col = cols[featureGroups(app_data)[features]],
                         lwd = 2, peakType = "none")
```

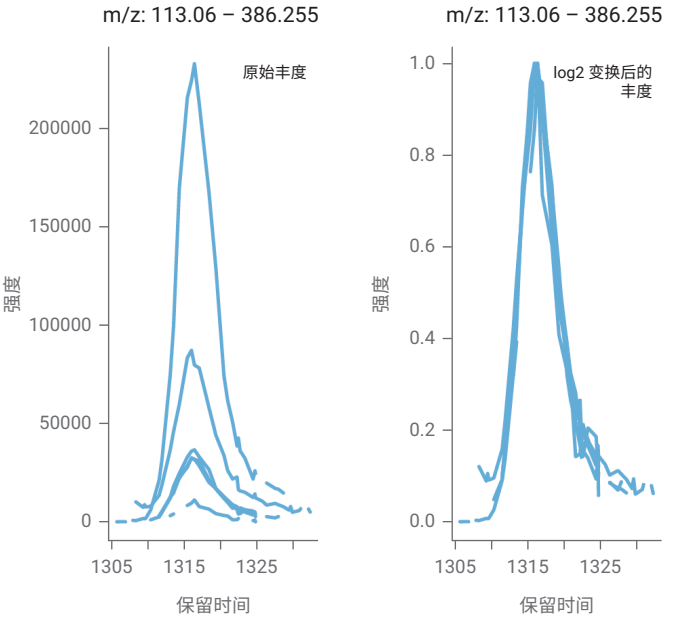

图 10. 特征归组过程：左图所示为原始数据，右图所示为经过 log2 变换的数据

从图 10 中可以看出，有五个特征被归为一个特征组。这些特征代表同一化合物，但是由于加合物或源内裂解等原因而成为五个。另外，从颜色可以看出，对于当前的特征组，不存在错误归组的特征。

完成上述步骤后，即可生成定量数据，其中行名为特征 (feature)，列名为分组信息，内容为强度信息。

```
res_xcms <- quantify(app_data)
```

接下来，将定量数据按照特征归组进行聚合，使其强度相加。再结合注释信息后，即可用于后续分析。

```
res_agg <- aggregateFeatures(res_xcms,
                             i = rownames(res_xcms),
                             fcol = "feature_group",
                             name = "agg_data")
```

使用 Spectra 包提取质谱信息

XCMS 本质上是一种特征提取工具，其本身无法用于操作质谱数据，需要配合其他包一起工作。在 Bioconductor 上，一般使用 Spectra 包对质谱信息进行提取和分析：

```
app_spectra_feature <- featureSpectra(app_data,
                                      msLevel = 2L,
                                      return.type = "Spectra")
# 用 featureSpectra()，保留 feature 信息。msLevel = 2L 表示只提取二级质谱信息。
app_spectra_feature <- setBackend(app_spectra_feature, MsBackendDataFrame())
# 转换数据后端类型，将其转换为 MsBackendDataFrame，将数据保留在 DataFrame 中，
加快处理速度
```

接下来，需要将具有相同特征的不同谱图整合（例如，不同组别中的同一特征的质谱图，其可能由于母离子丰度不同或其他原因而导致谱图外观存在差别）。首先，自定义一个函数（此函数来自 `Spectra` 包的官方教程，可用于将同一特征中 TIC 最强的碎片组合成一张二级质谱图）：

```
maxTic <- function(x, ...) {
  tic = vapply(x, function(z) sum(z[, "intensity"], na.rm = TRUE),
               numeric(1))
  x[[which.max(tic)]]
}
```

然后，利用以上新函数和 `Spectra` 包中的 `combineSpectra()` 函数，按照特征对质谱图进行聚合：

```
combined_app_spectra_feature <- Spectra::combineSpectra(app_spectra_feature,
                                                         f = app_spectra_featu
re$feature_id,
                                                         FUN = maxTic)
# 需要注意的是这里使用的 combineSpectra() 函数来自 Spectra 包，而默认可能会调用
MSnbase 中的同名函数，因此可以使用 Spectra::以避免错误
```

再对新生成的对象应用 `applyProcessing()` 函数，即可完成聚合操作：

```
applyProcessing(combined_app_spectra_feature)
```

```
## MSn data (Spectra) with 1300 spectra in a MsBackendDataFrame backend:
##              msLevel     rtime scanIndex
##            <integer> <numeric> <integer>
## F02.S04834         2   816.594      4834
## F02.S05131         2   864.563      5131
## F02.S07142         2  1183.999      7142
## F02.S11451         2  1811.653     11451
## F02.S04244         2   725.336      4244
## ...              ...       ...       ...
## F09.S11179         2  1770.397     11179
## F02.S12416         2  1918.667     12416
## F02.S04539         2   771.450      4539
## F01.S11699         2  1838.192     11699
## F14.S04041         2   693.027      4041
##  ... 39 more variables/columns.
## Processing:
##  Switch backend from MsBackendMzR to MsBackendDataFrame
[Tue Sep  5 15:00:18 2023]
```

此时，可以使用一种已知的化合物（鸟苷，guanosine）来查看二级质谱图，结果如图 11 所示。

```
guanosine_mz <- 284.0993
guanosine_rt <- c(305, 325)
guanosine_id <- rownames(featureDefinitions(app_data,
                                            mz = guanosine_mz,
                                            rt = guanosine_rt,
                                            ppm = 5))
guanosine_spectra <- combined_app_spectra_feature[combined_app_spectra_
feature$feature_id == guanosine_id]
plotSpectra(guanosine_spectra)
```

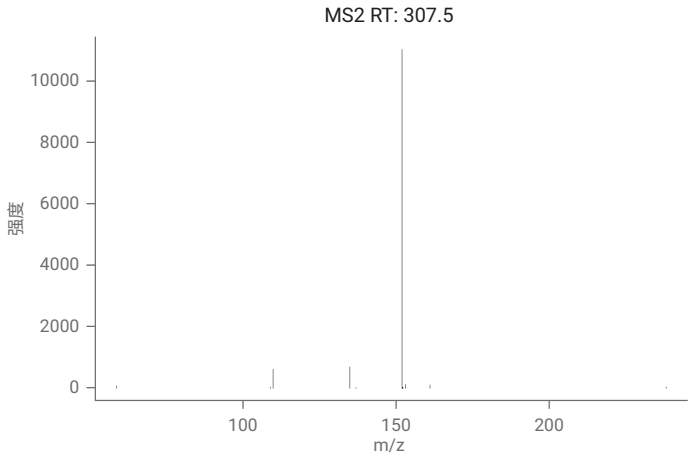

图 11. 鸟苷的整合二级质谱图

从图 11 中可以看出，所有样品中的特征所对应的 19 张二级谱图已经被整合为一张谱图，大幅提高了二级谱图鉴定的便利性和可靠性。

至此，已经按照特征提取到数据中的二级质谱信息，并对其进行了整合。如需更深入地了解 `Spectra` 包，请参考 `Spectra` 包的官方教程。

如果需要将结果以 mgf 格式文件导出并利用其他平台进行结构解析，则可以利用 `MsBackendMgf` 包[11] 来实现，具体操作步骤如下：

首先安装 `MsBackendMgf` 包：

```
BiocManager::install("MsBackendMgf")
```

然后执行如下操作，将数据结果导出：

```
library(MsBackendMgf)
export(combined_app_spectra_feature,
       backend = MsBackendMgf(),
       file = str_c(getwd(), "/export.mgf"))
```

然后即可将数据 (export.mgf) 上传至其他平台（例如 GNPS[12]），或者导入 SIRIUS[13-20] 中进行分析。

匹配谱库以进行化合物识别

从 MSP 文件生成谱库

MSP 格式的谱库比较容易获得（例如通过 MS-DIAL 官网），可基于此类谱库轻松创建能够为 `R` 所用的谱库。首先需要安装 `MsBackendMsp` 包（`Spectra` 包中的各种后端参见官方文档）；然后即可创建谱库。下面为了加快速度，仅创建一个较小的谱库（正离子模式，16481 个化合物，324191 张二级谱图）作为示例：

```
library(MsBackendMsp)
msdialPos <- "MSMS_Public_EXP_Pos_VS17.msp"
msdialPosSp <- Spectra(msdialPos, source = MsBackendMsp()) # 创建了一个新的谱库
```

可以创建多个 `MsBackendMsp` 对象，然后利用 `concatenateSpectra()` 函数将它们合并为更大的谱库，以提高匹配度。

使用 MetaboAnnotation 包进行化合物匹配

```
library(MetaboAnnotation)
app_match <- matchSpectra(combined_app_spectra_feature,
                          msdialPosSp,
                          param = CompareSpectraParam(requirePrecursor = T,
                                                      ppm = 50,
                                                      THRESHFUN = function(x)
 which(x >= 0.7)))
```

接下来创建一个归一化函数（参考官方教程），将数据中质谱图的丰度值转化为相对丰度，以便于与谱库中的谱图进行镜像比对：

```
norm_int <- function(x) {
   x[, "intensity"] = x[, "intensity"] / max(x[, "intensity"], na.rm = T)
   x
}
app_matched = addProcessing(app_match, norm_int) # 将归一化函数应用于数据
```

`validateMatchedSpectra(app_matched[whichQuery(app_matched)])` 可使用图形用户界面查看谱图匹配情况，如图 12 所示。

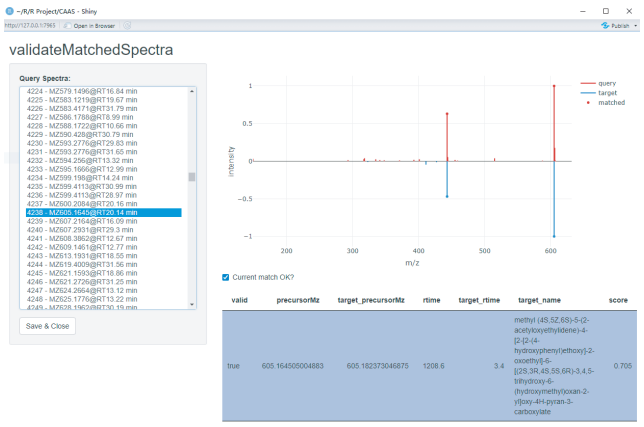

图 12. MetaboAnnotation::validateMatchedSpectra() 函数交互式查看谱图匹配

将匹配的结果返回到定量数据中

首先，需要提取关键数据，生成数据框，以便进行后续分析。此处需要提取的变量包括化合物名称、特征和匹配打分。

```
app_matched_df <- tibble(Compound = app_matched$target_name,
                         Feature = app_matched$feature_id,
                         Score = app_matched$score)
```

接下来，创建一个包含特征归组的数据框，其中涉及的变量包括特征和特征组：

```
ft_grp_mat <- tibble(Feature = rownames(featureDefinitions(app_data)),
                     Feature_Groups = featureGroups(app_data))
```

此时，只需按照特征列将两个数据框组合，即可得到包含特征、特征归组和化合物名称的数据框。但是，还应考虑到化合物的名称有可能重复。在本研究中，由于条件限制，无法通过标准品确认匹配的具有相同名称的化合物究竟是什么，因此可以为其添加标记。即，仅匹配到一次的化合物的显示方式保持不变；对于存在多个匹配结果的化合物，则在其名称中添加数字标签，例如 Phloretin 和 Phloretin_2。

```
mached_feature_groups <- left_join(ft_grp_mat, app_matched_df,
by = "Feature") %>%
  filter(!is.na(Compound)&Compound != "Pentaethylene glycol") %>% # 删除未匹
配到的和背景污染化合物
  group_by(Feature_Groups) %>%
  arrange(desc(Score)) %>%
  distinct(Feature_Groups, .keep_all = T) %>%
  ungroup()%>%
  group_by(Compound) %>%
  mutate(id = row_number()) %>%
  mutate(Compound_id = ifelse(id >= 2, str_c(Compound, id, sep = "_"),
Compound)) %>%
  ungroup() %>%
  select(1, 2, 6, 4) %>%
  rename(Compound = Compound_id)
```

表 5. 匹配的化合物及其对应的特征和特征归组

Feature	Feature_Groups	Compound	Score
FT0043	FG.0362.001.001	L-Proline	0.9779790
FT0537	FG.0007.001.001	Phloretin	0.9521480
FT1186	FG.0370.002.001	NCGC00380502-01_C19H28O10_beta-D-Glucopyranoside, 2-phenylethyl 6-O-beta-D-xylopyranosyl-	0.9504267
FT0047	FG.0278.001.001	Betaine; CE30; CE30; KWIUHFFTVRNATP-UHFFFAOYSA-N	0.9452283
FT0184	FG.0034.001.001	Phenylalanine	0.9433026
FT0156	FG.0455.002.001	Histidine	0.9427960

接下来，将定性结果与聚合的定量结果相结合，即可得到各组中识别出的化合物的峰面积信息（见表 6）：

```
app_agg_group <- assay(res_agg) %>%
  as.data.frame() %>%
  rownames_to_column(var = "Feature_Groups")

app_identified_quant <- left_join(mached_feature_groups,
                                  app_agg_group,
                                  by = "Feature_Groups") %>%
  select(-c(1, 2, 4: 7)) # 去除 Feature、Score 和 QC 的峰面积
```

表 6. 最终定性和定量分析结果

Compound	Sam1-1.mzML	Sam1-2.mzML	Sam1-3.mzML	Sam1-4.mzML	Sam2-1.mzML
L-Proline	1379357.11	1278952.66	1292965.84	1230300.9	1990998.07
Phloretin	1662647.72	1607765.06	1713684.57	1779211.4	1211873.65
NCGC00380502-01_C19H28O10_beta-D-Glucopyranoside, 2-phenylethyl 6-O-beta-D-xylopyranosyl-	171652.52	174667.51	302897.90	209806.6	165545.94
Betaine; CE30; CE30; KWIUHFFTVRNATP-UHFFFAOYSA-N	2085997.70	2134778.68	2107713.90	2128516.6	1790364.64
Phenylalanine	155590.42	160430.72	163438.45	157768.7	175355.91
Histidine	80855.96	93886.57	68481.24	104549.0	72006.29

虽然上面已经得到了预想的数据形式，但是这仅仅是一个表格。如需使用 Bioconductor 的更多资源进行分析，还需要将这些数据转换为 `SummarizedExperiment` 对象。以下使用 `readSummarizedExperiment()` 函数对数据进行转换。

```
app_summarized <- readSummarizedExperiment(app_identified_quant, ecol = 2:17,
fnames = "Compound")
# 将数据整理为 SummarizedExperiment 对象，2:17 指定量信息所在的列。
group <- c(rep("OriginA", 4),
           rep("OriginB", 4),
           rep("OriginC", 4),
           rep("OriginD", 4))
app_summarized$group <- group

app_summarized

## class: SummarizedExperiment
## dim: 98 16
## metadata(0):
## assays(1): ''
## rownames(98): L-Proline Phloretin ... Mono-iso-butyl phthalate
##   NCGC00385084-01_C23H32O15_alpha-D-Glucopyranoside,
##   3-O-[(2E)-3-(4-hydroxy-3,5-dimethoxyphenyl)-1-oxo-2-propen-1-yl]
-beta-D-fructofuranosyl_2
## rowData names(1): Compound
## colnames(16): Sam1-1.mzML Sam1-2.mzML ... Sam4-3.mzML Sam4-4.mzML
## colData names(1): group
```

至此，已经完成对质谱数据的处理。可以将数据导出为 csv 格式，通过 MPP 进行后续分析；也可以将此 `SummarizedExperiment` 对象，直接使用 Bioconductor 上的其他包进行分析。例如，可直接使用 `POMA` 包[21] 进行统计学分析（此处忽略了数据预处理），部分方差分析结果列于表 7 中。

```
library(POMA)
anova_result <- PomaUnivariate(app_summarized, method = "anova")
```

表 7. 部分方差分析结果

feature	pvalueAdj
L-Proline	0.0000002
Phloretin	0.0000004
NCGC00380502-01_C19H28O10_beta-D-Glucopyranoside, 2-phenylethyl 6-O-beta-D-xylopyranosyl-	0.0001272
Betaine; CE30; CE30; KWIUHFFTVRNATP-UHFFFAOYSA-N	0.0000010
Phenylalanine	0.0000000
Histidine	0.0000302

pvalueAdj 为调整 p 值

结论

本文介绍了使用 XCMS 对 auto MS/MS 数据进行特征提取、特征处理、谱图提取以及生成定量数据的完整工作流程。由于整个流程使用代码控制，因此可以首先使用带图形用户界面的 MassHunter 预览原始数据以及有代表性的基质添加物，以辅助确定 XCMS 的参数设置；完成前置工作后，即可通过脚本全自动运行 XCMS 的全部流程。最终定量数据既可以导出以通过 MPP 进行统计学分析，也可以直接用其他开源软件进行后续分析。

```
sessionInfo()

## R version 4.3.1 (2023-06-16 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=Chinese (Simplified)_China.utf8
## [2] LC_CTYPE=Chinese (Simplified)_China.utf8
## [3] LC_MONETARY=Chinese (Simplified)_China.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=Chinese (Simplified)_China.utf8
##
## time zone: Asia/Shanghai
## tzcode source: internal
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] POMA_1.10.0                 MetaboAnnotation_1.5.4
##  [3] MsBackendMsp_1.5.0          RColorBrewer_1.1-3
##  [5] lubridate_1.9.2             forcats_1.0.0
##  [7] stringr_1.5.0               dplyr_1.1.3
##  [9] purrr_1.0.2                 readr_2.1.4
## [11] tidyr_1.3.0                 tibble_3.2.1
## [13] ggplot2_3.4.3               tidyverse_2.0.0
## [15] Spectra_1.11.9              QFeatures_1.11.2
## [17] MultiAssayExperiment_1.27.5 SummarizedExperiment_1.31.1
## [19] GenomicRanges_1.53.1        GenomeInfoDb_1.37.3
## [21] IRanges_2.35.2              MatrixGenerics_1.13.1
## [23] matrixStats_1.0.0           MsFeatures_1.9.0
## [25] xcms_3.99.2                 MSnbase_2.27.1
## [27] ProtGenerics_1.33.1         S4Vectors_0.39.1
## [29] mzR_2.35.1                  Rcpp_1.0.11
## [31] Biobase_2.61.0              BiocGenerics_0.47.0
## [33] BiocParallel_1.35.4
##
## loaded via a namespace (and not attached):
##   [1] jsonlite_1.8.7                rstudioapi_0.15.0
##   [3] magrittr_2.0.3                MALDIquant_1.22.1
##   [5] rmarkdown_2.24                fs_1.6.3
##   [7] zlibbioc_1.47.0               vctrs_0.6.3
##   [9] multtest_2.57.0               memoise_2.0.1
##  [11] RCurl_1.98-1.12               base64enc_0.1-3
##  [13] htmltools_0.5.6               S4Arrays_1.1.6
```

```
##  [15] progress_1.2.2                curl_5.0.2
##  [17] AnnotationHub_3.9.2           SparseArray_1.1.12
##  [19] mzID_1.39.0                   htmlwidgets_1.6.2
##  [21] plyr_1.8.8                    cachem_1.0.8
##  [23] impute_1.75.1                 igraph_1.5.1
##  [25] mime_0.12                     lifecycle_1.0.3
##  [27] iterators_1.0.14              pkgconfig_2.0.3
##  [29] Matrix_1.6-1                  R6_2.5.1
##  [31] fastmap_1.1.1                 shiny_1.7.5
##  [33] GenomeInfoDbData_1.2.10       clue_0.3-64
##  [35] digest_0.6.33                 rsvg_2.4.0
##  [37] pcaMethods_1.93.0             colorspace_2.1-0
##  [39] AnnotationDbi_1.63.2          RSQLite_2.3.1
##  [41] filelock_1.0.2                fansi_1.0.4
##  [43] timechange_0.2.0              httr_1.4.7
##  [45] abind_1.4-5                   compiler_4.3.1
##  [47] bit64_4.0.5                   withr_2.5.0
##  [49] doParallel_1.0.17             DBI_1.1.3
##  [51] highr_0.10                    MASS_7.3-60
##  [53] MsExperiment_1.3.0            ChemmineR_3.53.2
##  [55] rappdirs_0.3.3                DelayedArray_0.27.10
##  [57] rjson_0.2.21                  tools_4.3.1
##  [59] interactiveDisplayBase_1.39.0 httpuv_1.6.11
##  [61] CompoundDb_1.5.0              glue_1.6.2
##  [63] promises_1.2.1                grid_4.3.1
##  [65] cluster_2.1.4                 generics_0.1.3
##  [67] gtable_0.3.4                  tzdb_0.4.0
##  [69] preprocessCore_1.63.1         hms_1.1.3
##  [71] MetaboCoreUtils_1.9.2         xml2_1.3.5
##  [73] utf8_1.2.3                    XVector_0.41.1
##  [75] BiocVersion_3.18.0            RANN_2.6.1
##  [77] foreach_1.5.2                 pillar_1.9.0
##  [79] limma_3.57.7                  later_1.3.1
##  [81] robustbase_0.99-0             splines_4.3.1
##  [83] BiocFileCache_2.9.1           lattice_0.21-8
##  [85] bit_4.0.5                     survival_3.5-7
##  [87] tidyselect_1.2.0              Biostrings_2.69.2
##  [89] knitr_1.43                    gridExtra_2.3
##  [91] xfun_0.40                     statmod_1.5.0
##  [93] DEoptimR_1.1-2                DT_0.29
##  [95] stringi_1.7.12                lazyeval_0.2.2
##  [97] yaml_2.3.7                    evaluate_0.21
##  [99] codetools_0.2-19              MsCoreUtils_1.13.1
## [101] BiocManager_1.30.22           cli_3.6.1
## [103] affyio_1.71.0                 xtable_1.8-4
## [105] munsell_0.5.0                 MassSpecWavelet_1.67.0
## [107] dbplyr_2.3.3                  png_0.1-8
## [109] XML_3.99-0.14                 parallel_4.3.1
## [111] ellipsis_0.3.2                blob_1.2.4
## [113] prettyunits_1.1.1             AnnotationFilter_1.25.0
## [115] bitops_1.0-7                  scales_1.2.1
## [117] affy_1.79.3                   ncdf4_1.21
## [119] crayon_1.5.2                  rlang_1.1.1
## [121] KEGGREST_1.41.0               vsn_3.69.0
```

参考文献

1. C.A. Smith, E.J. Want, G. O’ Maille, et al. XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. Analytical Chemistry, 78 (2006) 779–787. https://doi.org/10.1021/ac051437y

2. R. Tautenhahn, C. Boettcher, S. Neumann. Highly sensitive feature detection for high resolution LC/MS. BMC Bioinformatics, 9 (2008) 504. https://doi.org/10.1186/1471-2105-9-504

3. H.P. Benton, E.J. Want, T.M.D. Ebbels. Correction of mass calibration gaps in liquid chromatography-mass spectrometry metabolomics data. BIOINFORMATICS, 26 (2010) 2488. https://doi.org/10.1093/bioinformatics/btq441

4. J. Rainer, A. Vicini, L. Salzer, et al. A modular and expandable ecosystem for metabolomics data annotation in r. Metabolites, 12 (2022) 173. https://doi.org/10.3390/metabo12020173

5. J. Rainer. MsFeatures: Functionality for mass spectrometry features, 2022. https://doi.org/10.18129/B9.bioc.MsFeatures

6. J. Rainer, A. Vicini, L. Salzer, et al. A modular and expandable ecosystem for metabolomics data annotation in r. Metabolites, 12 (2022) 173. https://doi.org/10.3390/metabo12020173

7. J. Rainer, A. Vicini, L. Salzer, et al. A modular and expandable ecosystem for metabolomics data annotation in r. Metabolites, 12 (2022) 173. https://doi.org/10.3390/metabo12020173

8. L. Gatto, C. Vanderaa. QFeatures: Quantitative features for mass spectrometry data, 2022. https://github.com/RforMassSpectrometry/QFeatures

9. H. Wickham, M. Averick, J. Bryan, et al. Welcome to the tidyverse. Journal of Open Source Software, 4 (2019) 1686. https://doi.org/10.21105/joss.01686

10. E. Neuwirth. RColorBrewer: ColorBrewer palettes, 2022. https://CRAN.R-project.org/package=RColorBrewer

11. L. Gatto, J. Rainer, S. Gibb. MsBackendMgf: Mass spectrometry data backend for mascot generic format (mgf) files, 2023. https://doi.org/10.18129/B9.bioc.MsBackendMgf

12. M. Wang, J.J. Carver, V.V. Phelan, et al. Sharing and community curation of mass spectrometry data with global natural products social molecular networking. Nature Biotechnology, 34 (2016) 828–837. https://doi.org/10.1038/nbt.3597

13. Z.L. Sebastian Böcker Matthias Letzel, A. Pervukhin. SIRIUS: Decomposing isotope patterns for metabolite identification. Bioinformatics, 25 (2009) 218–224. https://doi.org/10.1093/bioinformatics/btn603

14. S. Böcker, K. Dührkop. Fragmentation trees reloaded. J Cheminform, 8 (2016) 5. https://doi.org/10.1186/s13321-016-0116-8

15. M.L. Kai Dührkop Markus Fleischauer, S. Böcker. SIRIUS: A rapid tool for turning tandem mass spectra into metabolite structure information. Nat Methods, 16 (2019) 299–302. https://doi.org/10.1038/s41592-019-0344-8

16. M.F. Kai Dührkop Louis-Félix Nothias, S. Böcker. Systematic classification of unknown metabolites using high-resolution fragmentation mass spectra. Nature Biotechnology, (2020). https://doi.org/10.1038/s41587-020-0740-8

17. K.D. Marcus Ludwig Louis-Félix Nothias. ZODIAC: Database-independent molecular formula annotation using Gibbs sampling reveals unknown small molecules. bioRxiv, (2019). https://doi.org/10.1101/842740

18. C.K. Yannick Djoumbou Feunang Roman Eisner. ClassyFire: Automated chemical classification with a comprehensive, computable taxonomy. J Cheminf, 8 (2016) 61. https://doi.org/10.1186/s13321-016-0174-y

19. M.M. Kai Dührkop Huibin Shen, S. Böcker. Searching molecular structure databases with tandem mass spectra using CSI:FingerID. Proc Natl Acad Sci USA, 112 (2015) 12580–12585. https://doi.org/10.1073/pnas.1509788112

20. H.W. Kim, M. Wang, C.A. Leber, L.-F. Nothias, R. Reher, K.B. Kang, J.J.J. van der Hooft, P.C. Dorrestein, W.H. Gerwick, G.W. Cottrell. NPClassifier: A deep neural network-based structural classification tool for natural products. Journal of Natural Products, 84 (2021) 2795–2807. https://doi.org/10.1021/acs.jnatprod.1c00399

21. R.A.C.-P. Castellano-Escuder Pol AND González-Domínguez. POMAShiny: A user-friendly web-based workflow for metabolomics and proteomics data analysis. PLOS Computational Biology, 17 (2021) 1–15. https://doi.org/10.1371/journal.pcbi.1009148

查找当地的安捷伦客户中心：

www.agilent.com/chem/contactus-cn

免费专线：

800-820-3278，400-820-3278（手机用户）

联系我们：

LSCA-China_800@agilent.com

在线询价：

www.agilent.com/chem/erfq-cn

www.agilent.com

DE88964235

2023 年 12 月 6 日，中国出版
5994-6891ZHCN

Agilent
Trusted Answers