bioinformatics.ca

Canadian Bioinformatics Workshops

www.bioinformatics.ca

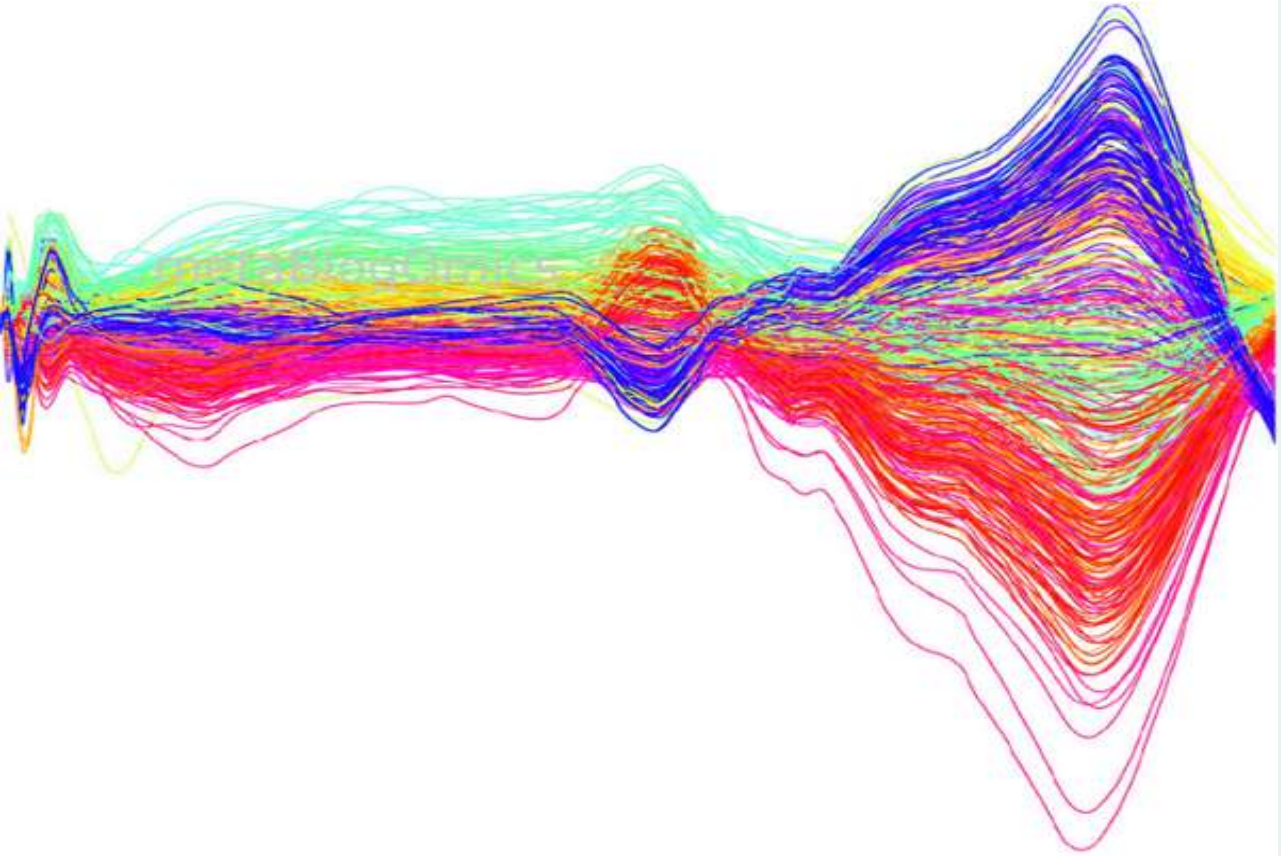
Module 5
Processing Raw LC-MS spectra using R and XCMS

David Wishart & Jeff Xia

Informatics and Statistics for Metabolomics

June 16-17, 2014

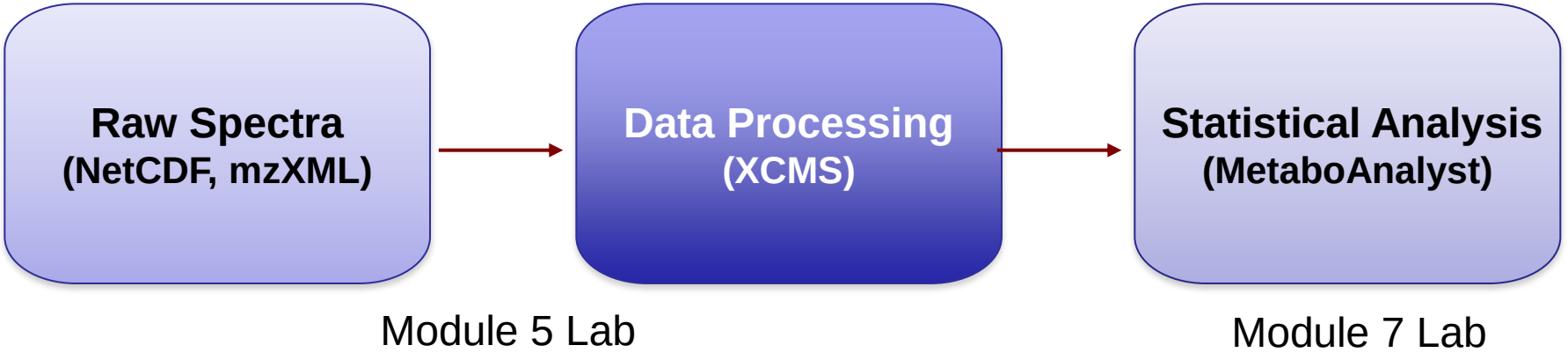
bioinformatics.ca

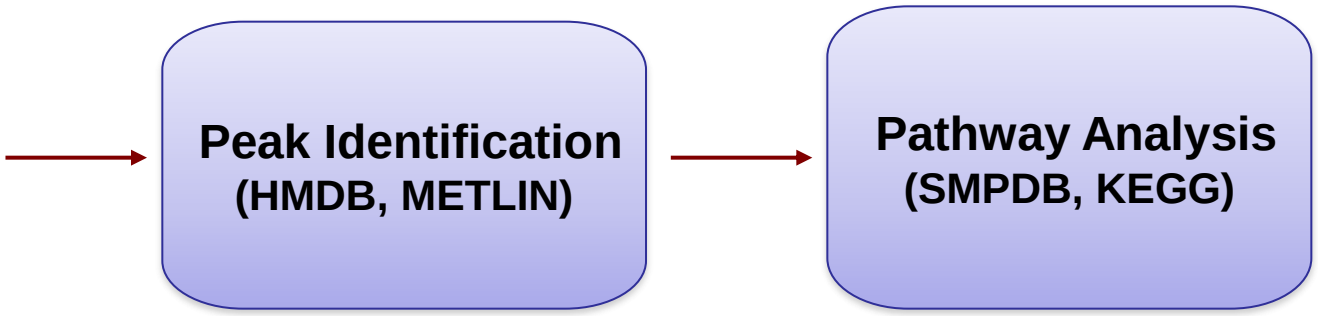


Learning Objectives

- Learn the basic steps in untargeted metabolomics
- Learn about XCMS and various input data formats
- Learn how to process raw LC-MS spectra using XCMS for MetaboAnalyst

Untargeted Metabolomics Overview



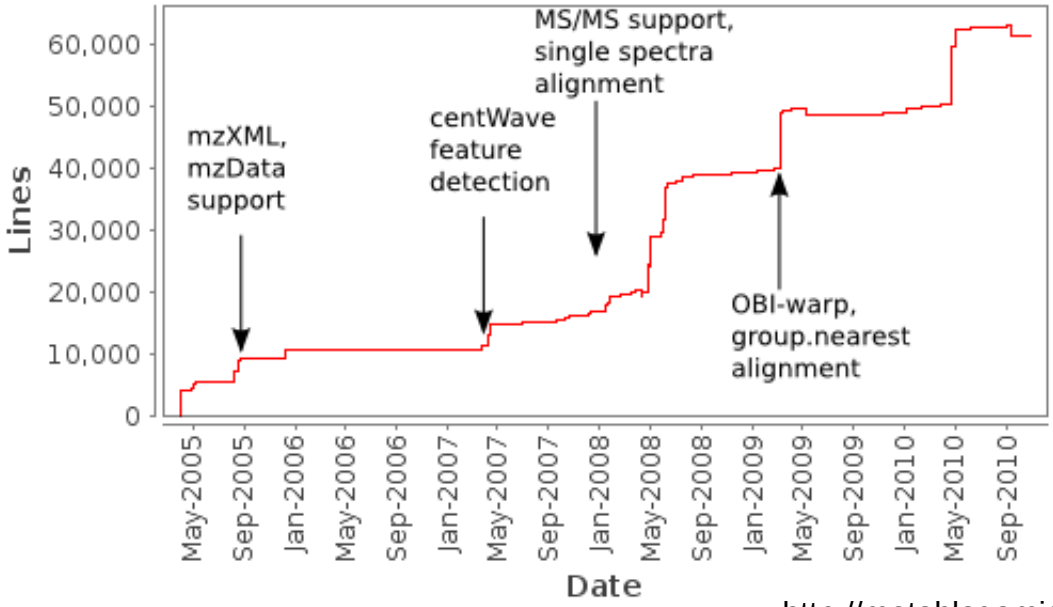


R & Bioconductor

- **R is a statistical programming environment**
- **Open source, cross platform**
- **Bioconductor project - open source software packages written in R for high-throughput "omics" data analysis**
  - Genomics, microarray, RNA-seq, proteomics, metabolomics
- **Limitations**
  - Requires substantial effort to learn statistics and programming skills before one can do a meaningful data analysis
  - Mainly command driven

XCMS

- **Bioconductor package for processing LC/GC-MS spectra**
- **Widely used for untargeted metabolomic studies**

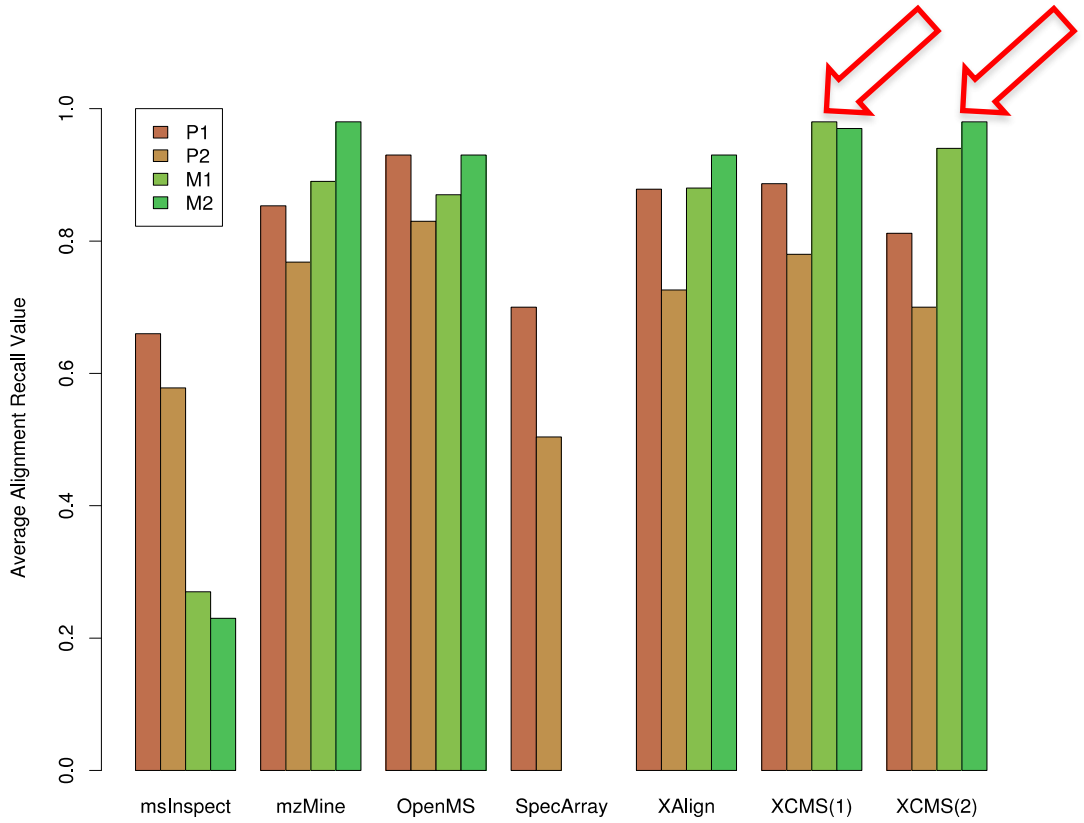


http://metablogomics.blogspot.ca/

Why XCMS (1)

- **Free, open source, powerful and flexible**
- **High-throughput (batch processing)**
- **Cutting-edge algorithms**
  - peak detection
  - peak deconvolution
  - peak alignment

Why XCMS (2)



Highest average precision and recall

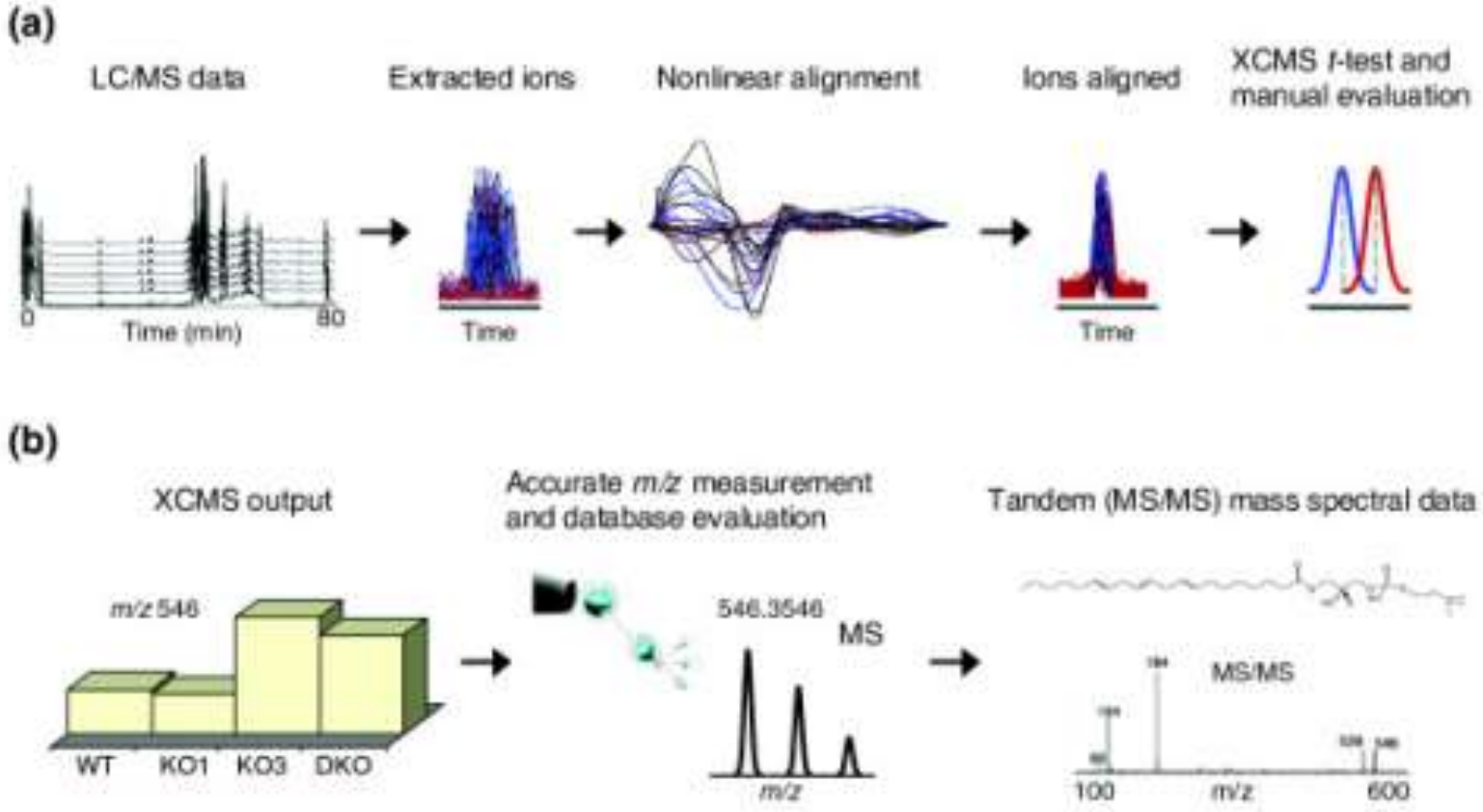
Table 7: Wall-clock runtime for the metabolomics data sets M1 and M2 in minutes

Data set	msInspect	MZmine	OpenMS	SpecArray	XAlign	XCMS	
						without retention time	with correction
M1	12	20	4.4	-	51	0.9	1.4
M2	24	44	8.7	-	35	5.5	5.8
Total	36	64	13.1	-	86	6.4	7.2

Superfast

BMC Bioinformatics. 2008; 9: 375.

LC-MS and XCMS



Some Notes

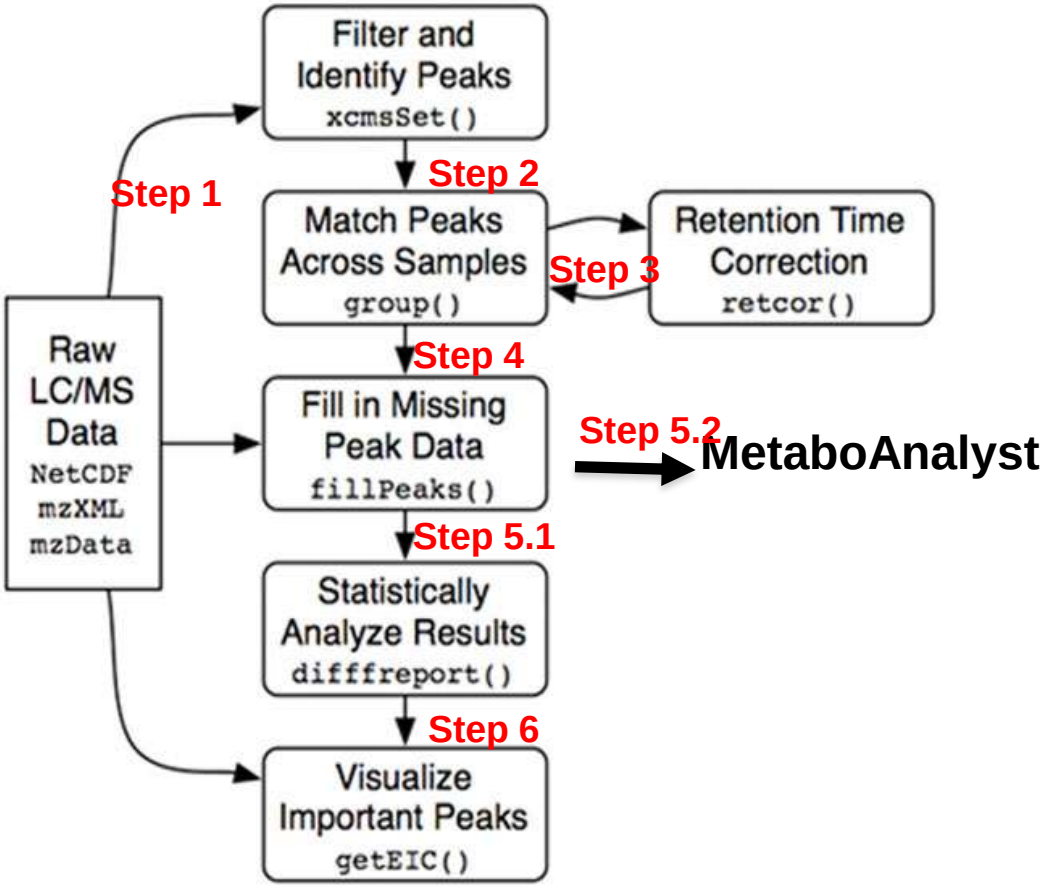
```
# start of comments
> start of R command



# R command format
> output.data <- function(input.data)
> ?function # getting help
```

Prerequisites

- **Latest R installed**
- **XCMS package installed**
- **Test data “faahKO” installed**

```
# use biocLite to install a Biocondcutor package
> source("http://bioconductor.org/biocLite.R")
# Install the xcms package
> biocLite("xcms")
# Install dataset package used in this session
> biocLite("faahKO")
# Install multtest package for diffreport function
> biocLite("multtest")
```

Basic XCMS Flowchart



R commands overview

```
> library(xcms)
> cdfpath <- system.file("cdf", package = "faahKO")
> cdffiles <- list.files(cdfpath, recursive = TRUE, full=T) # input files (step 1)
> xset <- xcmsSet(cdffiles)  # peak picking (step 2)
> xsg <- group(xset)    # peak alignment (step 3.1)
> xsg <- retcor(xsg)    # retention time correction (step 3.2)
> xsg <- group(xsg)     # re-align (step 3.3)
> xsg <- fillPeaks(xsg)  # filling in missing peak data (step 4)
> dat <- groupval(xsg, "medret", "into")  # get peak intensity matrix (step 5)
> dat <- rbind(group = as.character(phenoData(xsg)$class), dat)  # add group label
> write.csv(dat, file= ‘MyPeakTable.csv’)  # save the data to CSV file
>
```

Prepare Input (step 1)

- **Supported formats: NetCDF, mzXML, mzData**
- **Software for most instruments can export to NetCDF (often referred to as CDF or AIA)**

```
# put all .cdf files inside a folder named ‘myspectra’, save the
# folder under your current working directory
> cdffiles <- list.files(‘./myspectra’, recursive = TRUE, full=T)
# cdffiles now contain absolute path to all raw spectra
> cdffiles
```

Prepare input (step 1)

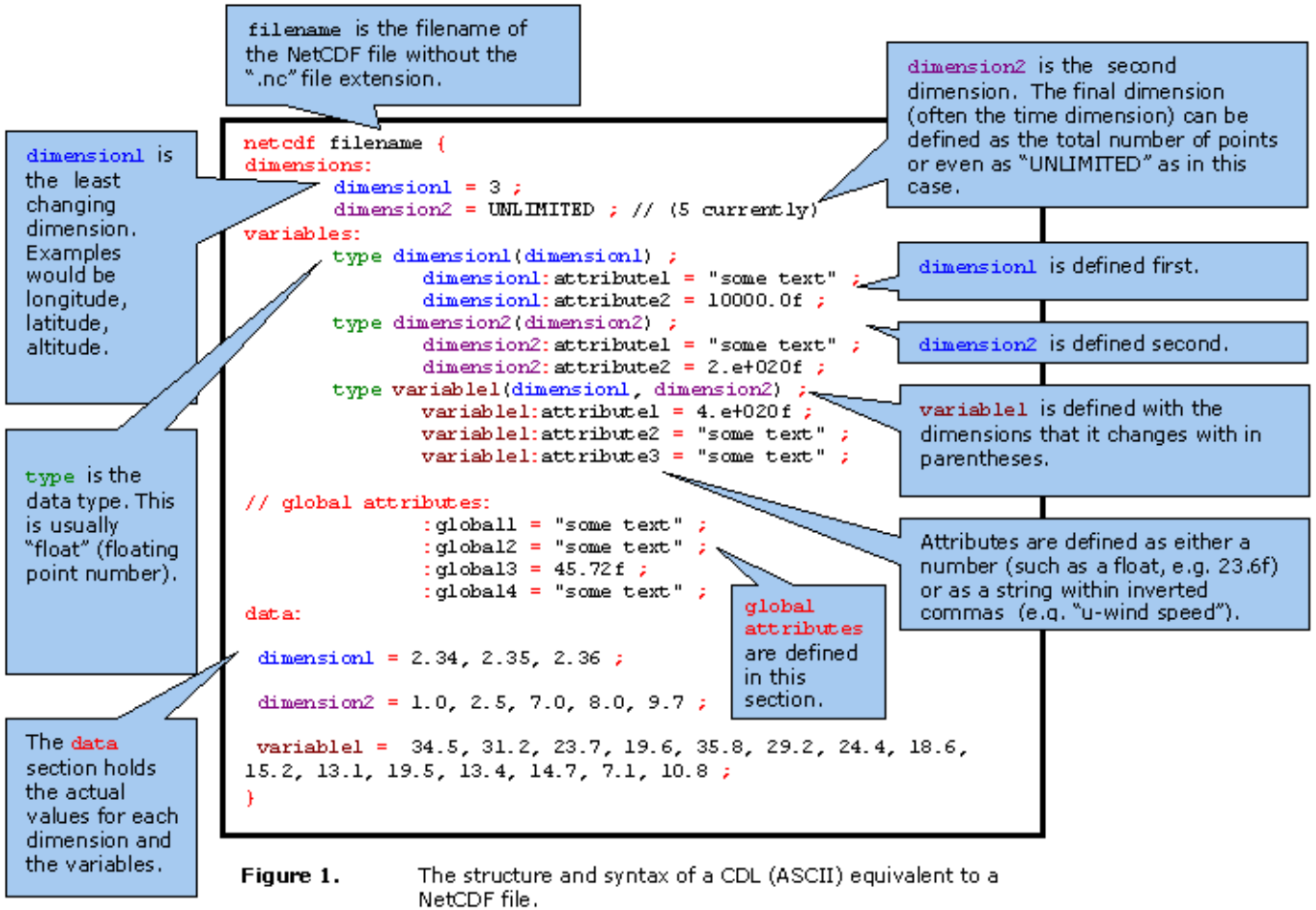
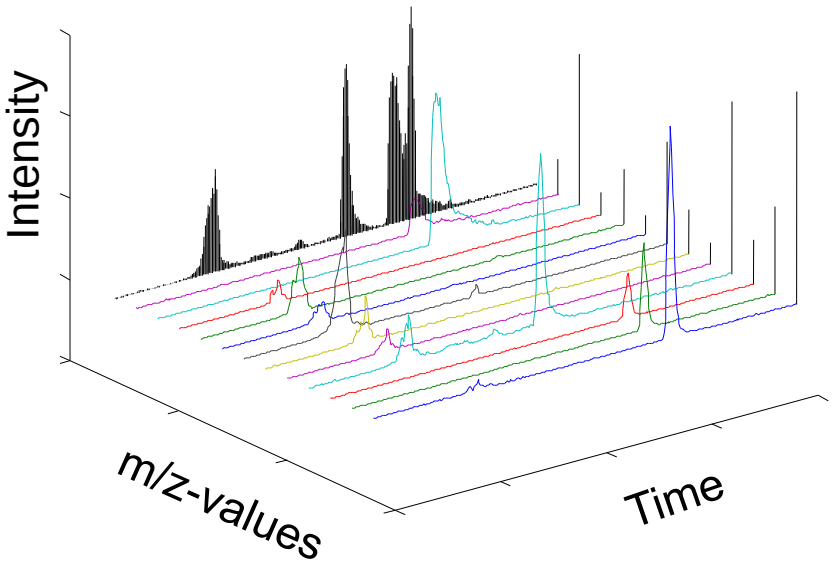
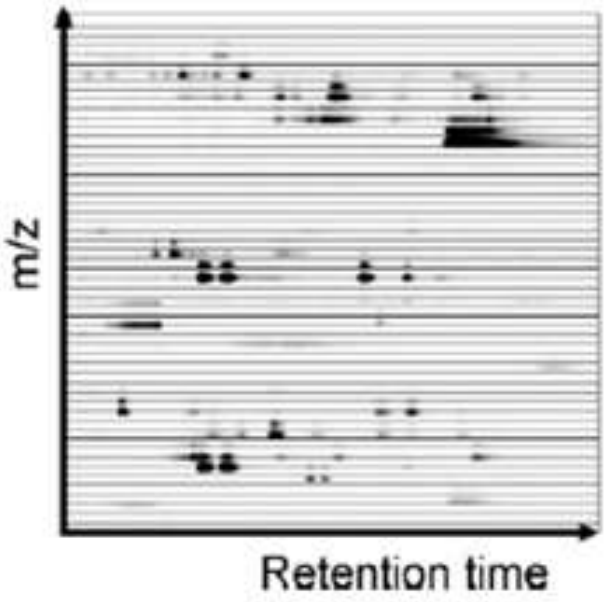


Figure 1. The structure and syntax of a CDL (ASCII) equivalent to a NetCDF file.

Peaking Detection (step 2)





Peaking Detection (step 2)

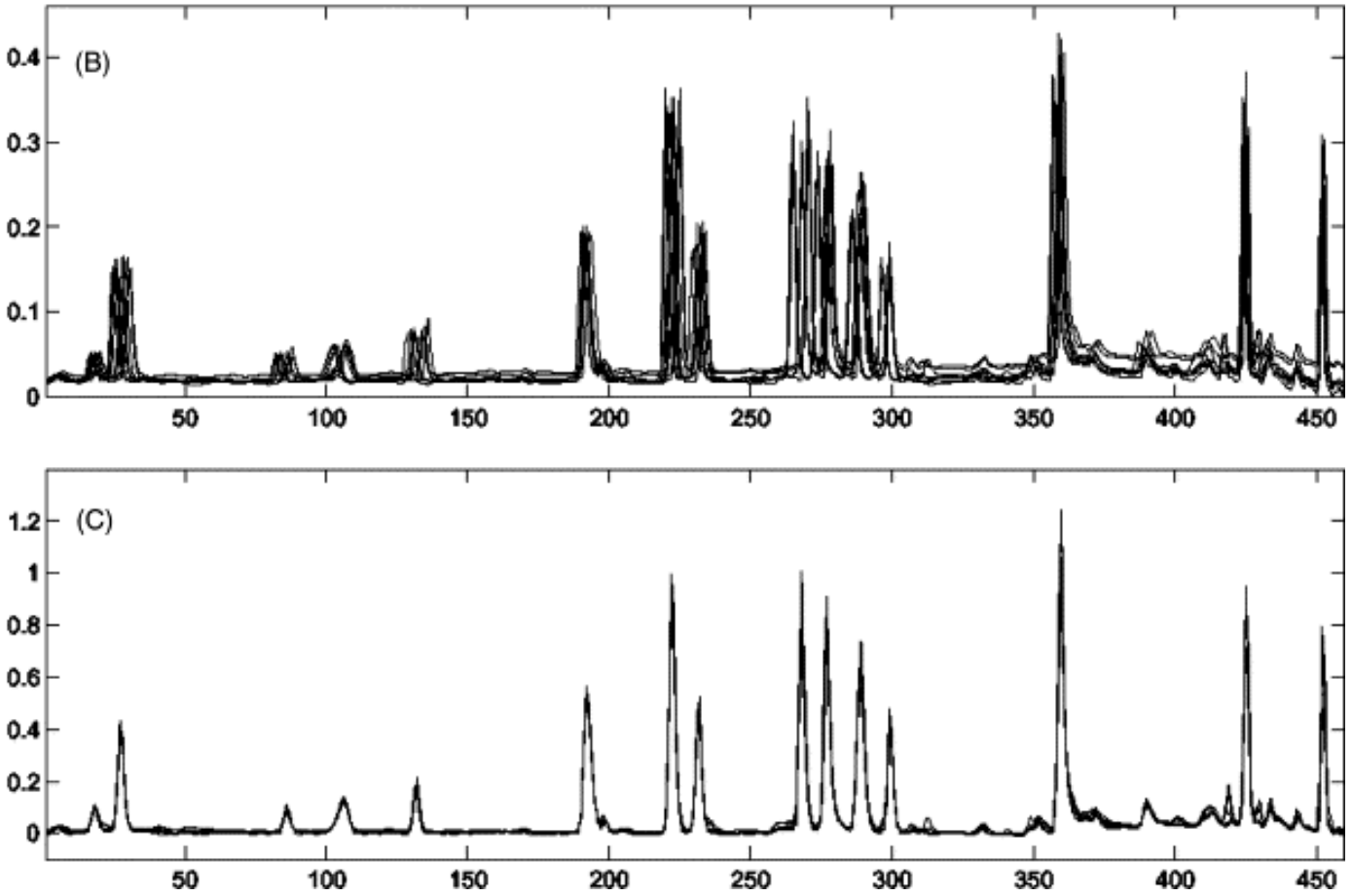
```
> cdfpath <- system.file("cdf", package = "faahKO")
> cdffiles <- list.files(cdfpath, recursive = TRUE, full=T) # input files (step 1)
> xset <- xcmsSet(cdffiles)
250:38 300:103 350:226 400:338 450:431 500:529 550:674 600:847
250:43 300:128 350:275 400:394 450:500 500:637 550:835 600:1027
250:25 300:93 350:227 400:337 450:411 500:498 550:640 600:758
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
>
```

- **Some important parameters**
  - **xcmsSet(..., scanrange=c(lower, upper)) # to scan part of the spectra**
  - **xcmsSet(..., fwhm = seconds)**

    # specify full width at half maximum (default 30s) based on the type of chromatography
  - **xcmsSet(..., method = 'centWave')**

    # use wavelet algorithm for peak detection, suitable for high resolution spectra

Peak Alignment & Retention Time Correction (step 3)

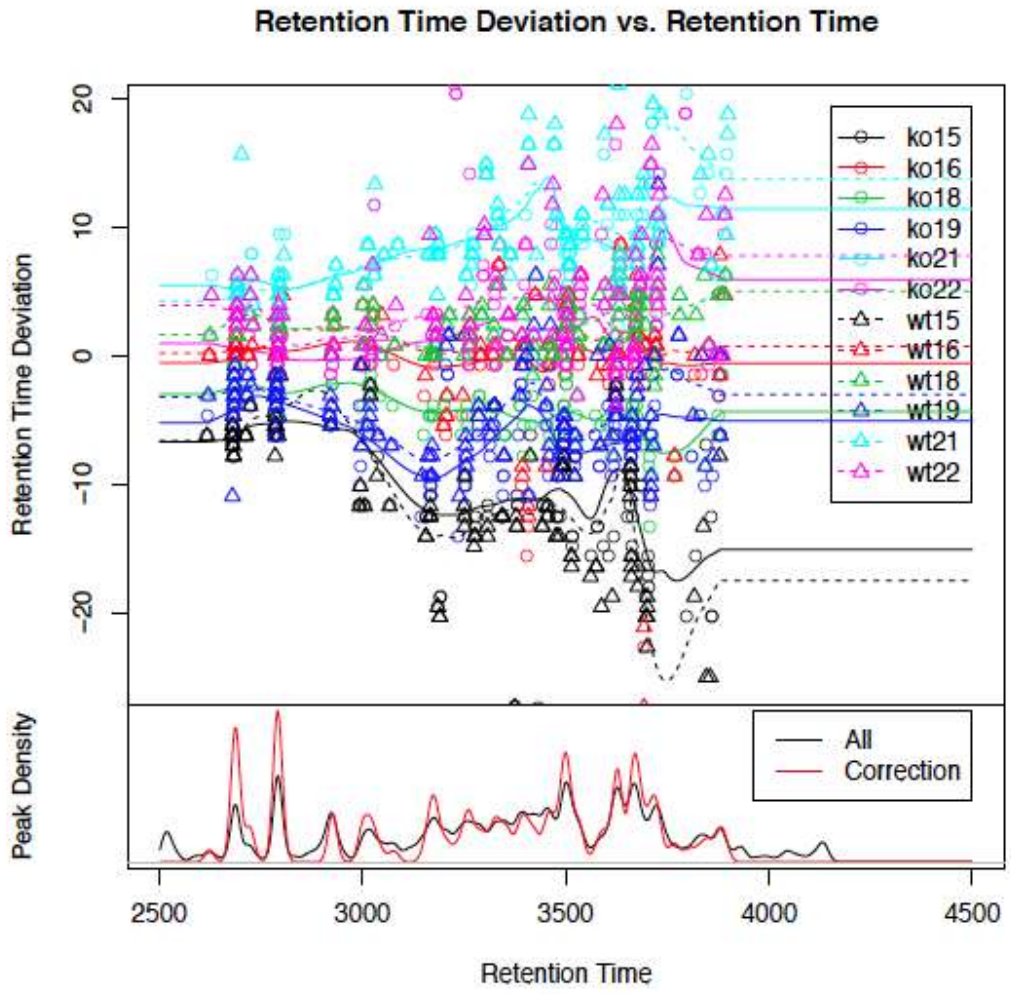


Peak Alignment & Retention Time Correction with XCMS

- **Matching peaks across samples**
- **Using the peak groups to correct drift**
- **Re-do the alignment**
- **Can be performed iteratively until no further change**

```
> xsg <- group(xset)    # peak alignment
> xsg <- retcor(xsg)    # retention time correction
# xsg <- retcor(xsg, plottype = 'mdevden') # also plot the result
> xsg <- group(xsg, bw=10)    # re-align with tighter range
```

Retention Time Deviation Profile



Filling in Missing Peaks (step 4)

- A significant number of potential peaks can be missed during peak detection
- Missing values are problematic for robust statistical analysis
- We now have a better idea about where to expect real peaks and their boundaries
- Re-scan the raw spectra and integrate peaks in the regions of the missing peaks

Filling in missing peaks (step 4)

```
> xsg <- fillPeaks(xsg)  # filling in missing peak data
```

- **Some warnings may show up when the expected peak (as indicated by many other files) are beyond the "end" of the file. There is no raw data available for fillPeaks(). These warnings can be ignored.**

Results of Peak Detection

- **Long list of peaks with**
  - mz, mzmin, mzmax
  - rt, rtmin, rtmax
  - peak intensities/areas (raw data)

```
> peaks(xsg)[1:10, ]
           mz mzmin mzmax       rt    rtmin    rtmax       into  maxo sample
 [1,] 200.1000 200.1 200.1 2934.069 2918.299 2948.200  147887.53  9687      1
 [2,] 201.0638 201.0 201.1 2537.824 2522.175 2556.604  204572.42  7726      1
 [3,] 205.0000 205.0 205.0 2789.840 2775.835 2805.384 1778568.94 84280      1
 [4,] 205.9819 205.9 206.0 2791.393 2777.394 2805.384  237993.62 10681      1
 [5,] 207.0821 207.0 207.1 2718.340 2704.553 2732.193  380873.05 18800      1
 [6,] 208.0671 208.0 208.1 2647.286 2631.687 2662.867   96070.72  4112      1
 [7,] 208.1201 208.1 208.2 2716.810 2704.553 2732.193   67967.10  2878      1
 [8,] 219.0848 219.0 219.1 2525.305 2511.220 2540.954  235544.92 11588      1
 [9,] 229.1000 229.1 229.1 2522.175 2509.655 2536.259   87236.08  6649      1
[10,] 233.0390 233.0 233.1 3026.699 3012.052 3041.379  399145.34 19752      1
>
```

Statistical Analysis with XCMS (step 5.1)

- XCMS‘diffreport’computes Welch's two-sample t-statistic for each analyte and ranks them by p-value.
- It returns a summary report

```
>
> reporttab <- diffreport(xsg, "WT", "KO")
> reporttab[1:3, 1:13]
      name     fold    tstat       pvalue    mzmed    mzmin    mzmax    rtmed
1 M300T3392 5.693594 14.44368 5.026336e-08 300.1898 300.1706 300.2000 3391.922
2 M301T3389 6.098560 15.41220 5.630163e-08 301.1879 301.1659 301.1949 3389.377
3 M298T3186 3.915807 12.18165 2.544233e-07 298.1508 298.1054 298.1592 3186.221
     rtmin    rtmax npeaks KO WT
1 3382.165 3396.330     12  6  6
2 3382.165 3396.330      7  6  1
3 3180.805 3192.017      4  4  0
>
```

- Multivariate analysis and visualization can be performed using MetaboAnalyst

Statistical Analysis with MetaboAnalyst (step 5.2)

- **The general format required by MetaboAnalyst and most other statistical tools is a data matrix:**
  - Features (peaks) in rows;
  - Samples in columns;
  - Group labels;

```
> dat <- groupval(xsg, "medret", "into")  # get peak intensity matrix
# add group labels (KO, WT)
> dat <- rbind(group = as.character(phenoData(xsg)$class), dat)
> write.csv(dat, file="MyPeakTable.csv")  # save the data to CSV file
```

Peak Intensity Table

	ko15	ko16	ko18	ko19	ko21	ko22	wt15	wt16	wt18	wt19	wt21	wt22
group	KO	KO	KO	KO	KO	KO	WT	WT	WT	WT	WT	WT
200.1/2926	147887.526	451600.713	65290.3849	56546.1373	85147.0429	162012.439	175177.079	82619.48	51942.9448	69198.2221	153273.469	98144.28
205/2791	1778568.94	1567038.14	1482796.38	1039129.82	1223132.35	1072037.7	1950287.49	1466780.6	1572679.16	1275312.76	1356014.33	1231442.16
206/2791	237993.621	269713.984	201393.416	150107.31	176989.653	156797.035	276541.849	222366.155	211717.713	186850.878	188285.941	172348.755
207.1/2719	380873.049	460629.738	351750.138	219287.968	286848.561	235022.626	417169.585	324892.463	277990.701	220972.352	252874.011	236728.16
219.1/2524	235544.923	173623.378	82364.5911	79480.3984	185792.429	174458.768	244584.471	161184.045	72029.3783	75096.9891	238194.379	173829.951
231/2516	0	70796.2076	222609.068	286232.146	435094.492	100076.188	0	73142.0489	165382.595	240261.212	201316.154	179437.72
233/3023	399145.34	356951.308	410550.655	198416.547	363381.665	317805.788	397107.783	271252.06	334459.927	181901.317	456900.469	294270.588
234/3024	76880.8652	99479.8431	97427.9085	53440.0968	88227.7903	81072.2347	65215.6404	55914.7466	73781.0086	45136.3262	83693.3853	57524.7078
235.1/2694	171995.219	128945.162	155442.477	159288.56	99668.458	112541.211	199981.494	148699.757	156968.295	174559.048	142974.789	106634.911
236.1/2524	252282.035	206031.927	71763.7915	67643.1587	186660.976	198804.282	253791.071	187225.645	53773.0561	90012.6427	256263.261	206486.981
240.2/3681	112440.555	153375.51	193768.652	170641.492	88800.0299	146563.043	122672.854	270872.933	176425.151	187062.973	75235.9352	124070.147
241.1/3679	1465988.67	1318746.57	1215368.74	632037.425	579968.192	561964.932	1468102.64	1594705.06	1006929.31	805533.2	731777.647	522916.137
242.1/3679	280767.61	248792.381	224467.69	109019.428	103855.804	101092.934	280260.239	299453.212	188328.195	139748.3	139968.457	95347.8537
244.1/2832	612169.852	256316.47	90539.8608	35364.5996	54603.1821	115505.42	627834.975	56875.2604	168524.487	27262.6173	110170.73	1352929.86
246.1/2517	27932.12	26061.0198	0	0	56145.3946	47901.5546	105508.752	70508.8223	24386.5748	0	77205.3608	43561.775
249.1/3668	1435000.72	1228148.16	1193346.7	641881.756	536808.256	574005.708	1297985.82	1566269.11	1076654.09	747969.405	688496.327	485427.604
250.1/3668	347794.775	238893.441	248245.61	126623.762	107844.148	118626.972	281745.027	336058.763	215534.93	149872.756	139317.286	99814.4919
254.1/3231	79006.9631	564680.786	89307.0016	127186.81	42745.7444	28747.1784	70817.6569	124200.881	108479.022	42715.7163	78615.5039	44408.536
255.2/3678	1420043.19	1187751.92	1264221.96	673816.219	581177.903	651560.258	1211727.06	1550070.66	1024087.2	875296.292	759075.901	640885.69
256.2/3678	307708.367	263117.303	275104.925	135782.849	121369.162	128540.429	258452.243	338390.94	213618.13	180384.486	164675.68	91822.7078
256.2/3453	229192.802	2899288.97	245869.325	114129.761	65505.2262	62148.0395	249301.23	293543.931	185415.276	98358.9931	170656.944	159465.518

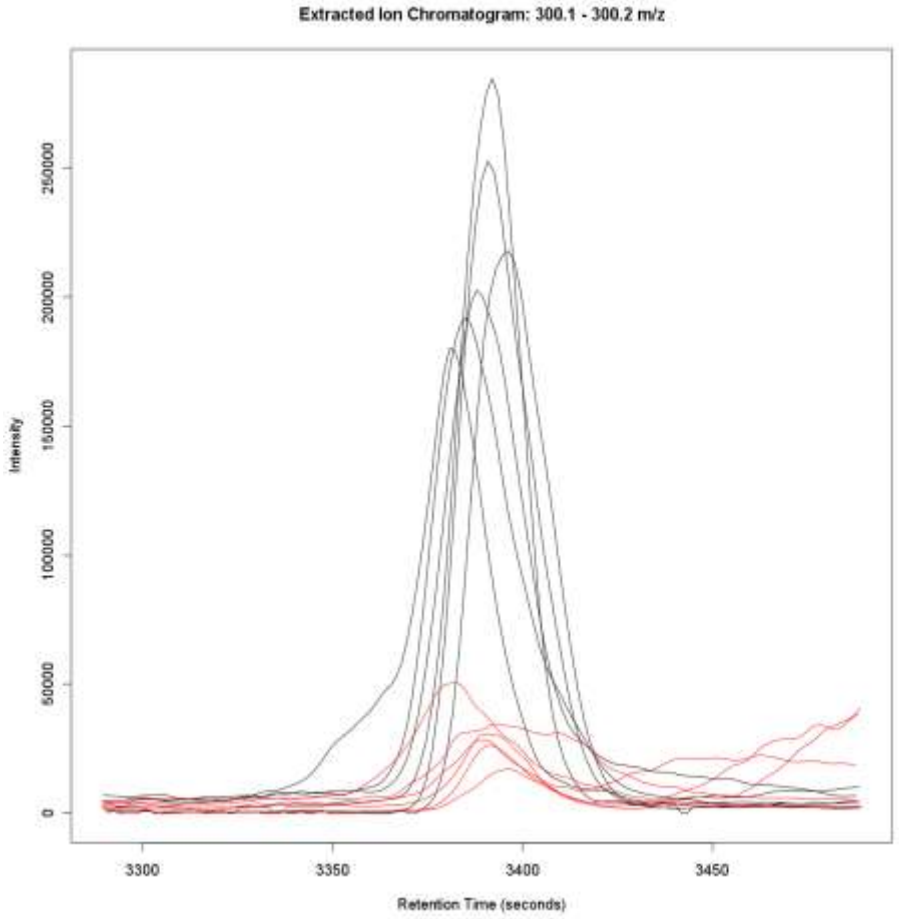
- Peaks are identified by “m/z / retention time”
- Can be directly uploaded to MetaboAnalyst

Visualizing Peaks (step 6)

- When significant peaks are identified, it is critical to visualize these peaks to assess quality
- This is done using the Extracted ion chromatogram (EIC)

```
>
> gt <- groups(xsg)
> gt[1:3,]
        mzmed    mzmin    mzmax    rtmed    rtmin    rtmax npeaks KO WT
[1,] 200.1000 200.1000 200.1000 2926.132 2876.260 2934.069      9  4  5
[2,] 205.0000 205.0000 205.0000 2790.613 2789.840 2791.023     12  6  6
[3,] 205.9927 205.9786 206.0023 2790.529 2789.155 2792.428     12  6  6
> # select peaks with median retention time [3385, 3400], detected in at least 8 samples
> grp_idx <- which(gt[,"rtmed"] > 2600 & gt[,"rtmed"] < 2700 & gt[,"npeaks"] == 12)[1]
> eiccor <- getEIC(xsg, groupidx = grp_idx);
ko15 ko16 ko18 ko19 ko21 ko22 wt15 wt16 wt18 wt19 wt21 wt22
> # plot with colors corresponding to group labels
> plot(eiccor, col=as.numeric(phenoData(xsg)$class));
>
```

Visualizing Important Peaks (step 6)



XCMS Online


Scripps Center For Metabolomics XCMS Online
Home
Public Shares
XCMS Institute
Publications
Contact
XCMS Online
Click here to view the new Interactive Cloud Plot!
Institute
Click here to go to XCMS Institute
METLIN
New to XCMS Online?
Simple mass spectrometry data processing.
Register ›
Current Users:
E-mail:
jeff@hancocklab.cor
(e.g. researcher@scripps.edu)
Password:
Sign In
Forgot your password?
Do you want to hear about all of the latest XCMS Online news? Click here to get on our newsletter to keep informed about what we're doing to help you with your metabolomics data.
DOWNLOAD USER MANUAL

File Format

- **File formats that can be directly uploaded to XCMS Online are:**
  - mzXML
  - mzData
  - .cdf NetCDF (AIA/ANDI)
  - .d folders (Agilent; Bruker)
  - .wiff files (AB SCIEX)

The Main Steps

1. Log in (need to register first)
2. Create a new job
3. Upload your data
4. Specify parameters
5. Submit the job
6. Wait for email notification
7. Reformat your data for MetaboAnalyst

Summary

- **We have only shown the most basic functions of XCMS. For more advanced tutorials and troubleshooting see:**
  - Tutorial:
    - http://bioconductor.org/packages/release/bioc/vignettes/xcms/inst/doc/xcmsPreprocess.pdf
  - Discussion Forum
    - http://www.metabolomics-forum.com/viewforum.php?f=8