Proteomics

Cavan Reilly

December 6, 2019

Table of contents

Overview

2 dimensional gel electrophoresis

Mass spectrometry

Shotgun proteomics

Analysis of MS data in R

iTRAQ

Metabolomics

Biomarkers

Proteomics

Proteomics is the study of all proteins and their various forms much like genomics is the study of genes.

Standard assays exist for measuring individual proteins, so what makes proteomics distinct is that we use some method for studying all of the proteins in some biological context, or what is called the *proteome*.

What the various methods of proteomics research have in common is an attempt to separate a collection of proteins in a way so that one can quantify that which one has separated.

Proteomics

Proteins frequently undergo post-translation modifications and these changes can have a dramatic impact on the function of the protein.

For example, phosphorylation is the addition of a phosphate group (PO_4) to a protein and this can have a dramatic impact on the function of the protein.

Some proteins can undergo phosphorylation at multiple sites and the site at which they are phosphorylated can determine the activity of the protein (e.g. the RNA binding molecule CELF1).

Proteomics

Hence much like investigating if different isoforms of the same gene are present at different levels in a biological sample, proteomics researchers are interested in the extent to which proteins exhibit different post-translational modifications.

As there is at least one protein for each gene, there are at least 22,000 proteins in humans, but when one considers the number of different gene products produced via alternative splicing and if one distinguishes between the various post-translational modifications of a protein (as one should) the magnitude of the human proteome is enormous.

Proteomics

There is also a huge range of protein concentrations in human samples:

IL-6 levels in human serum samples have been reported at 2 pg/ml (Lai R, et al., 2002, *Cancer*, 95, 1071-1075),

while albumin has been found as high as 50 mg/ml (Ritchie RF, et al. (1999) *J Clin Lab Anal*, 13, 280-286) giving a more then 10 orders of magnitude range of concentrations.

2 dimensional gel electrophoresis

A number of different technologies have been developed for separating complex mixtures of proteins.

One method is 2 dimensional gel electrophoresis.

Typically, one first separates by isoelectric focusing (which depends on the charge state of the protein), then by mass.

To compare samples one needs to align the 2 images one obtains after separating then quantify the darkness of the spot.

2-D DIGE (fluorescence 2 dimensional differential gel electrophoresis) is an alternative that uses multiple dyes to label samples thus avoiding the need for aligning if one only compares 2 samples.

2 dimensional gel electrophoresis

To best make use of this approach for realistic clinical applications one should use an internal control as a reference and run all of one's samples with this internal control.

One can identify the proteins at a spot in a gel via Edman sequencing or mass spectrometry.

The use of gels has a long history and the use of multivariate statistical techniques that are now common for the analysis of microarrays dates back to the early 1980s in the gel electrophoresis literature.

Unfortunately, over the last 15 years it has become apparent that multiple proteins can reside at the same spot on a gel and not all proteins are labeled by the reagents used for identifying proteins in this system.

2 dimensional gel electrophoresis

There is an R package called digeR that can be used for analysis of these types of data sets.

This package expects data in a file with x and y coordinates of the spots and the spot volume for each sample (so it does not have warping functionality).

Most image analysis software (e.g. Progensis and Metamorph) is capable of producing files of this type from images.

This package is unlike the others we have used in class thus far: it is driven by a graphical user interface.

Its functionality doesn't go much beyond things we've already seen how to do using the command line in R and isn't nearly as flexible.

Protein microarrays

Some researchers have successfully bound proteins to substrates (typically highly engineered slides that have, for example, nanowells or microfluidic channels).

One can then screen proteins for various properties, such as the tendency to bind to other molecules-these are *functional protein microarrays*.

For example, in Zhu H, et al. (2001), “Global analysis of protein activities using proteome chips”, *Science*, 293, 2101-2105.

Protein microarrays were used to screen yeast proteins for binding activity to calmodulin (a protein that is part of many cellular processes due to its ability to bind to calcium ions).

Protein microarrays

Others have developed protein microarrays that are more like genomic microarrays by binding protein specific probes (e.g. antibodies) to a substrate then exposing that substrate to a sample of proteins.

See for example Sreekumar A, et al. (2001), "Profiling of cancer cells using protein microarrays: discovery of novel radiation-regulated proteins", *Cancer Research*, 61, 7585-7593.

The latter type of microarray is called a *analytical protein microarray*.

Cross-hybridization seems to be more of an issue than is the case with arrays with bound nucleic acid sequences.

Protein microarrays

These and other microarray applications in proteomics entail the same sorts of considerations we dealt with in the context of genome microarray, namely

- normalization: if we think that most spots shouldn't show a signal then there shouldn't be any systematic deviations in plots like an MA plot
- confounders: are there differences among the samples that should be accounted for when testing for differences between groups, e.g. age
- multiple hypothesis testing
- pathway analysis

Mass spectrometry

Most contemporary researchers use mass spectrometry in some form to do proteomics.

A mass spectrometer consists of 3 basic components:

1. an ion source
2. a mass analyzer
3. a detector

There are many choices available for these components leading to a host of different approaches.

Mass spectrometry

Most researchers purchase the machines from one of a small number of manufacturers, so there are frequently issues with proprietary methods and software.

Nonetheless there are some standardized data formats such as the mzXML and netCDF file formats (the Bioconductor package mzR has functions for reading these files into R and manipulating such files).

MS versus tandem MS (i.e. MS/MS): latter is for sequencing ions.

The development of 2 ionization methods, matrix assisted laser desorption ionization (MALDI) and surface enhanced laser desorption/ionization (SELDI), allowed for the use of mass spectrometry for the analysis of complex biological samples.

Mass spectrometry

These are typically used in conjunction with time of flight (TOF) mass analyzer.

These determine the mass by measuring the amount of time it takes for the ion to travel through a tube with known length.

Once the ion hits the detector the charge is determined and the mass over charge is computed for that ion.

Shotgun proteomics

In shotgun proteomics one first digests the sample containing proteins so as to generate a set of fragments.

Typically trypsin is used for this, and upon digesting a protein with trypsin generally 30-50 different peptides will be created.

One then uses a mass spectrometer to examine the spectrum of a sample.

Many have explored the use MALDI-TOF and SELDI-TOF for analysis of the trypsin digested samples.

Others have pursued the use of LC-MS/MS for analysis of the resulting mixture.

Shotgun proteomics

The use of MALDI-TOF and SELDI-TOF produces tens of thousands of data points which represent the intensity corresponding to each ion mass to charge ratio.

As these methods mostly produce singly charged ions we can roughly think of the data as quantity of each mass.

One problem is that the intensities have not proved as reproducible as many chemists would expect.

Shotgun proteomics

This has been explained by

- the exact chemical composition of the sample
- differences in the substrate used
- other poorly understood factors

For this reason some think of shotgun proteomics as not really a quantitative approach, I tend to think of it as signal corrupted by noise (like all the data I see).

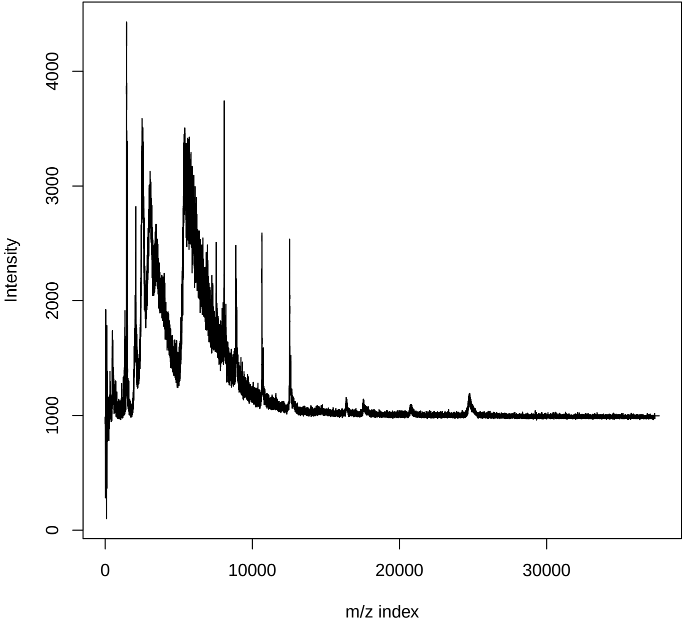
Analysis of MS data in R

There are several packages available for the analysis of MALDI-TOF and SELDI-TOF data.

These include IPPD, PROcess and MassSpecWavelet.

We will examine the MassSpecWavelet package as an example of what is involved.

Here is an example of a SELDI-TOF spectrum from a patient sample.


Intensity
0
1000
2000
3000
4000
m/z index
0
10000
20000
30000

Analysis of MS data in R

The continuous wavelet transform of a signal $s(t)$ depending on value t is defined for a pair of scales and positions a and b respectively by

$$C(a, b) = \int s(t)\psi_{a,b}(t)\, dt$$

where

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right).$$

and $\psi(t)$ is the *mother wavelet*.

We will use the Mexican hat wavelet as the mother wavelet: it is proportional to the second derivative of the standard normal density.

Analysis of MS data in R

$C(a, b)$ measures the extent to which your signal is similar to the mother wavelet near position b at scale a.

The idea behind the algorithm used in the MassSpecWavelet package is that if a portion of the spectrum is similar to our mother wavelet at some scale then there is a peak at that location.

Wavelets are used throughout all areas of science for this sort of signal filtering property.

It is an extension of Fourier based techniques: the primary difference is that we allow the frequency composition of a signal to vary over t, which for us is mass/charge value.

Analysis of MS data in R

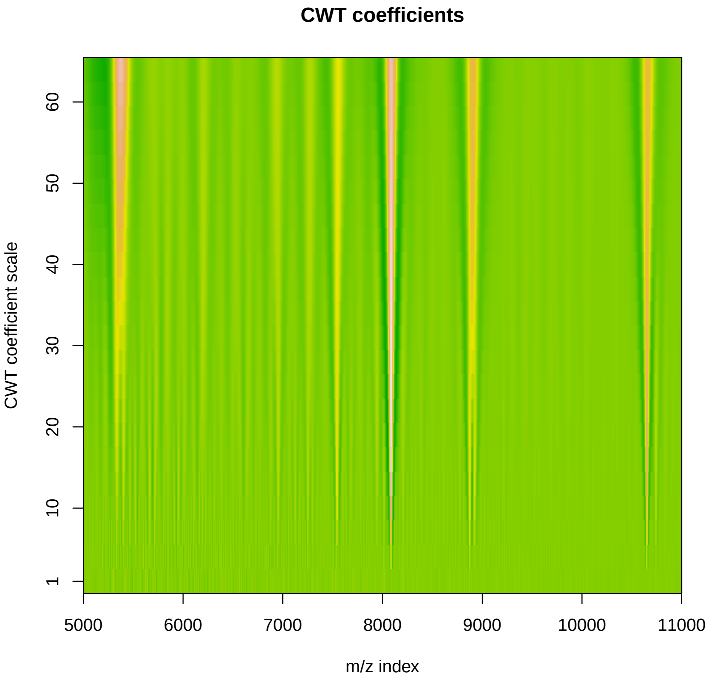
To use the MassSpecWavelet, first get the continuous wavelet transform using the cwt function.

```
> library(MassSpecWavelet)
> data(exampleMS)
> scales <- seq(1, 64, 3)
> wCoefs <- cwt(exampleMS, scales=scales,
+ wavelet="mexh")
> wCoefs <- cbind(as.vector(exampleMS), wCoefs)
```

Analysis of MS data in R

We can then make a plot to examine the coefficients.

```
> xTickInterval <- 1000
> plotRange <- c(5000, 11000)
> image(plotRange[1]:plotRange[2], scales,
+ wCoefs[plotRange[1]:plotRange[2],2:23],
+ col=terrain.colors(256), axes=FALSE,
+ xlab="m/z index", ylab="CWT coefficient scale",
+ main="CWT coefficients")
> axis(1, at=seq(plotRange[1], plotRange[2],
+ by=xTickInterval))
> axis(2, at=c(1, seq(10, 64, by=10)))
> box()
```

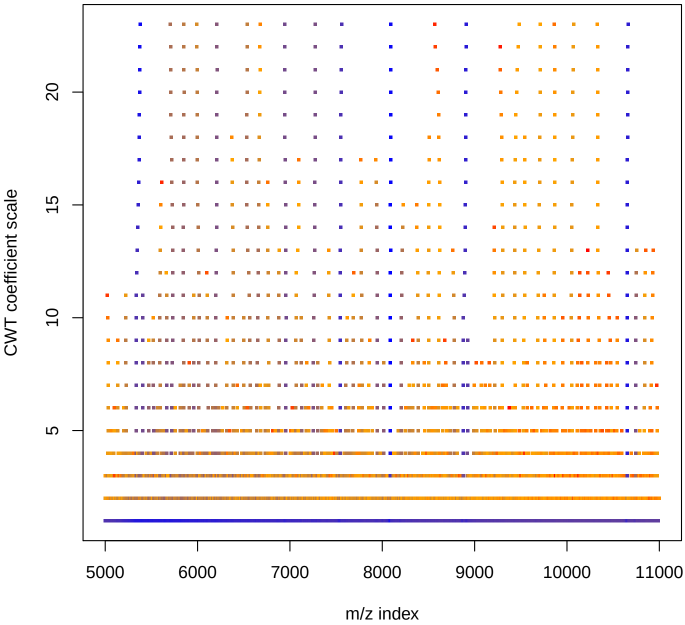

CWT coefficients
CWT coefficient scale
1
10
20
30
40
50
60
5000
6000
7000
8000
9000
10000
11000
m/z index

Analysis of MS data in R

The method is based on searching the collection of wavelet coefficients for local maxima in the position at each scale, then connecting these to identify ridges.

So we next get local maxima and ridges, then plot these local maxima (the set of ridges looks the same).

```
> colnames(wCoefs) <- c(0, scales)
> localMax <- getLocalMaximumCWT(wCoefs)
> ridgeList <- getRidge(localMax)
> plotLocalMax(localMax, wCoefs, range=plotRange)
```

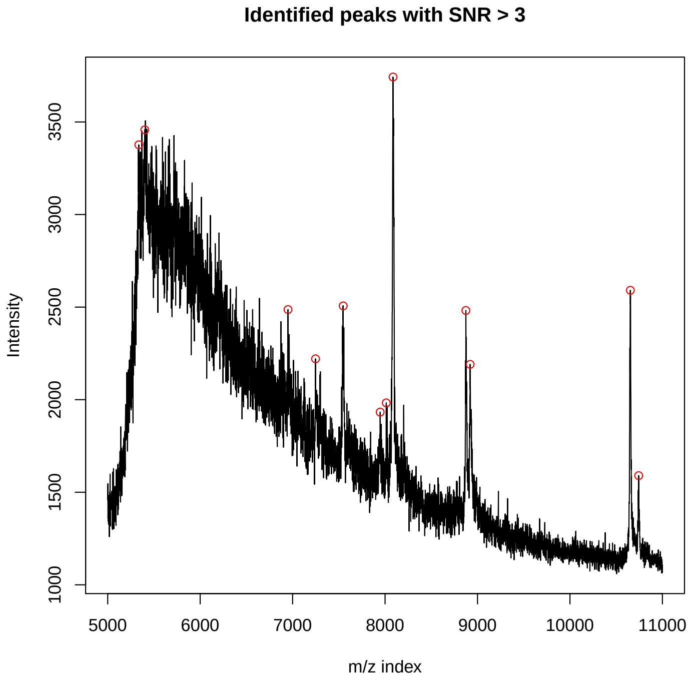

CWT coefficient scale
5
10
15
20
5000
6000
7000
8000
9000
10000
11000
m/z index

Analysis of MS data in R

We then use the `ridgeList` to identify the major peaks.

We will also allow for small peaks near larger ones.

```
> SNR.Th <- 3
> nearbyPeak <- TRUE
> majorPeakInfo <- identifyMajorPeaks(exampleMS,
+ ridgeList, wCoefs, SNR.Th = SNR.Th,
+ nearbyPeak=nearbyPeak)
> peakIndex <- majorPeakInfo$peakIndex
> plotPeak(exampleMS, peakIndex, range=plotRange,
+ main=paste("Identified peaks with SNR >", SNR.Th))
```


Identified peaks with SNR > 3
Intensity
1000
1500
2000
2500
3000
3500
5000
6000
7000
8000
9000
10000
11000
m/z index

Analysis of MS data in R

We can do all of these steps at once using the peakDetectionCWT function.

This is a wrapper function that calls all of the routines we have used.

```
> nearbyPeak <- TRUE
> peakInfo <- peakDetectionCWT(exampleMS,
+ SNR.Th=SNR.Th, nearbyPeak=nearbyPeak)
> majorPeakInfo <- peakInfo$majorPeakInfo
> peakIndex <- majorPeakInfo$peakIndex
```

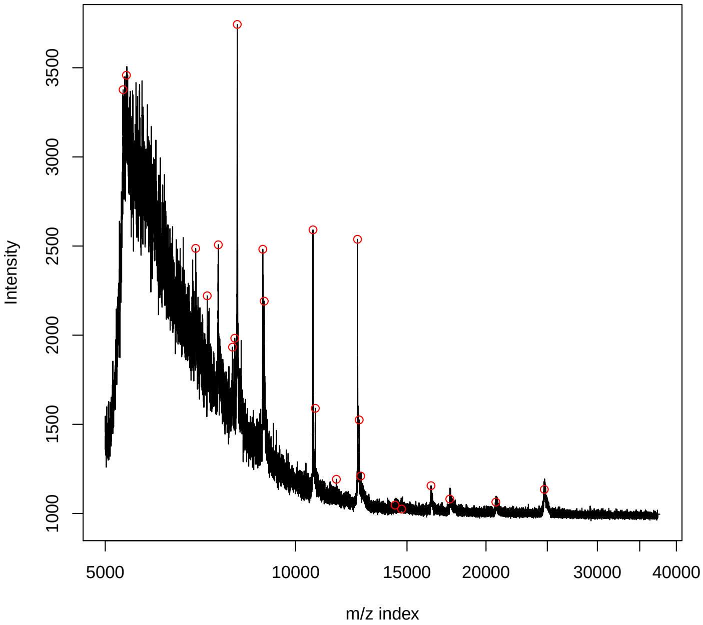
Analysis of MS data in R

Sometimes improvements are possible by using the initial set of peaks as a starting point.

We will now look at the entire spectrum.

```
> plotRange <- c(5000, length(exampleMS))
> betterPeakInfo <- tuneInPeakInfo(exampleMS,
+ majorPeakInfo)
> plotPeak(exampleMS, peakIndex, range=plotRange, log="x",
+ main=paste("Identified peaks with SNR >", SNR.Th))
> plotPeak(exampleMS,betterPeakInfo$peakIndex,
+ range=plotRange, log="x",
+ main=paste("Identified peaks with SNR >", SNR.Th))
```

Identified peaks with SNR > 3



Analysis of MS data in R

One can get peak locations and heights by accessing the peakIndex object.

```
>  peakIndex
  1_106   1_143   1_175   1_231   1_265   1_336   1_498   1_563   1_694  1_1314
    106     143     175     231     265     336     498     563     694    1314
...
 1_5401  1_6950  1_7248  1_7547  1_7947  1_8013  1_8086  1_8874  1_8920 1_10653
   5401    6950    7248    7547    7947    8013    8086    8874    8920   10653
...
1_24720

  24720
```

To check this, let's examine the spectrum around the peak at location 8086.

```
> exampleMS[8076:8096]
 [1] 2295 2471 2471 2664 2882 2999 3192 3356 3494 3721 3743 3606 3509 3514 3425

[16] 3520 3281 3142 3007 2935 2984
```

So that does appear to be a local maximum.

Analysis of MS data in R

We can also generate a list of peaks and the intensity at that peak.

```
> cbind(peakIndex, exampleMS[peakIndex])
          peakIndex
  1_106           106 1784
  1_143           143 1387
  1_175           175 1068
  1_231           231 1161
...
1_17526         17526 1109
1_20716         20716 1084
1_24720         24720 1177
```

So one can process the samples separately to get a sample specific peak list.

Testing for differences in proteomics

When you combine data across samples, one needs to allow some deviations in the m/z value.

Usually there will be many zeros after combining as certain peaks in one sample will have no corresponding peak in another sample.

This has led to a number of approaches to developing 2 sample tests that outperform existing methods as they have been developed specifically for the case of many zeros.

What these tests do is combine a test for the difference in the sample proportions with a test for a difference in the non-zero values to obtain a single test statistic.

Such tests are usually based on the theory of likelihood ratio tests.

Analysis of MS data in R

The IPPD package has similarly functionality but uses an extension of the normal distribution to model the shape of the peaks.

The generalization allows for some skewness in the shape.

The PROcess package has a set of tools for baseline subtraction, smoothing and peak picking.

This package requires lots of user specification of parameters.

iTRAQ

iTRAQ is the name of a system that is used to label peptides with one of up to 8 different tags.

The tags are designed so that the ions resulting from them during MS/MS differ in mass by 1 weight unit.

The tags reflect different biological samples (some tags can be used as internal controls so that one can compare across more than 8 samples).

The different tags are designed to not alter the relative mass of different peptides so that the tagged peptides with the same identity but different tags all have the same mass.

Hence in single MS mode one can extract a peptide, break it up, then use tandem MS to determine the sequence of the peptide.

iTRAQ

During MS/MS the tag breaks off so that one gets reports on the quantity of each tag (since the mass of the tag ions is known) at the same time as one determines the sequence of the peptide.

The system for running the assay comes with some software (Pro Quant), however there are a couple of R packages that allow for exploratory examination of the data.

i-Tracker is a perl script that can be used to link the results from the iTRAQ system to other peptide identification systems, such as Mascot and Sequest.

iTRAQ in R

The R package MSnbase has a set of classes to enable analysis of proteomics data sets in a manner that is similar to R packages for analysis of microarrays (i.e. the eSet and Expression classes).

It can read in data in the mzXML, mzData, and mzML formats or in the form of peak lists (in the mgf format).

It has tools to display the spectrum and a number of quality checks and methods for cleaning up the data.

iTRAQ in R

The isobar package is designed with the same goal in mind, however it currently has less functionality in terms of input files.

It implements a normalization method that computes a factor such that the median intensity (across all peptides) for all reporter channels are equal.

Other than this the MSnbase package has more tools for quality assessment.

Metabolomics

A metabolite is a small molecule (less than 1000 Daltons) that is involved in biological processes.

This includes many familiar biological molecules, e.g. sugars, lipids.

By measuring the molecules actually involved in a metabolic process the hope is that we will develop more sensitive measures of disease processes.

The 2 primary tools for this are nuclear magnetic resonance spectroscopy and liquid chromatography coupled with mass spectrometry (LC-MS).

We will examine the latter here.

Metabolomics

In LC-MS, one first separates the sample of interest via chromatography, so that different peptides in the complex mixture elute out of the column in a manner that depends on specific properties of the peptides.

As the sample elutes, the sample is subjected to an MS run.

These data sets generate many spectra for each sample, the Water's based system used here with which I am most familiar generates about 7.5 million observations per sample.

Thus the sample is separated into about 8 million “bins” so hopefully each bin contains only one compound.

The file formats are the same as for proteomics, e.g. mzXML files.

Metabolomics

There is an R package called xcms that provides a complete analysis strategy for these data sets.

Currently there is not much support for quality assessment and remediation provided by this package.

We will discuss the package called xcms-it implements the original algorithm (published in 2006) but allows for choices.

These methods have extremely limited functionality: basically they can just test for differences between groups without any covariate adjustment.

Metabolomics in R

The xcms package determines group membership based on the directory structure from which it reads the data.

So in the directory mzXMLFiles I have a directory called testDir1 in which I put collections of files obtained under the same conditions into 2 distinct subdirectories.

These subdirectories are called testDat1 and testDat2 and each has 4 mzXML files obtained by applying markerWolf to the files that come out of the Waters's pipeline. For example

```
> tset1 <- xcmsSet("C:/mzXMLFiles/testDir1")
```

Metabolomics in R

```
> tset1
An "xcmsSet" object with 8 samples
Time range:  8.5-748.5 seconds (0.1-12.5 minutes)
Mass range:  50.5756-999.8372 m/z
Peaks:  26159 (about 3270 per sample)
Peak Groups:  0
Sample classes:  testDat1, testDat2
Profile settings:  method = bin
                  step = 0.1
Memory usage:  3 MB
```

Metabolomics in R

The XCMS algorithm is structured as follows

1. peak detection: slice the 2 dimensional m/z, rt space into strips that are some fraction of a mass unit (e.g. 0.1 m/z) wide and then for each pair of slices the maximum is computed for all time points across the pair-this gives the extracted ion base-peak chromatogram (EIBPC).

Then filter this with the second derivative of a Gaussian density with sd=13 (the zero crossings of the density define the endpoints used to compute the area and so no background correction is performed).

Then keep peaks where the signal to noise ratio exceeds 10.

Metabolomics in R

2. peak matching: there is more variation in the retention time axis, so use fixed 0.25 m/z bins to match peaks in the m/z axis with overlapping bins (e.g. 100.0-100.25, 100.125-100.375).

Once peaks are matched across samples based on m/z, a nonparametric density estimate is applied to the sets of retention times.

The modes of the smoothed density are called meta-peaks.

Then meta-peaks are only retained when at least half of the subjects are part of that meta-peak.

Metabolomics in R

3. Retention time alignment: From this one typically gets several hundred “well behaved” peak groups (i.e. those that most samples have a peak and very few have multiple peaks).

Then for each of these groups one computes a median and a deviation from median.

This typically results in a detailed nonlinear retention time deviation as it depends upon retention time.

A loess curve is then fit to these data and this gives the alignment function to use.

At the ends where there are no well behaved peaks the alignment function goes to a constant.

Then one can use the corrected peak lists to do peak matching again. Could do this over and over but just 2 groupings is what is generally recommended.

Metabolomics in R

So let's do the peak grouping.

```
> tset2 <- group(tset1)
> tset2
An "xcmsSet" object with 8 samples
Time range:  8.5-748.5 seconds (0.1-12.5 minutes)
Mass range:  50.5756-999.8372 m/z
Peaks:  26159 (about 3270 per sample)
Peak Groups:  3255
Sample classes:  testDat1, testDat2
Profile settings:  method = bin
                   step = 0.1
Memory usage:  3.42 MB
```

Metabolomics in R

Then we can do RT alignment as follows, here we try both options.

```
> tset3a <- retcor(tset2, family = "symmetric",
+ plottype = "mdevden")
Retention Time Correction Groups:  866
> tset3b <- retcor(tset2, family = "gaussian",
+ plottype = "mdevden")
Retention Time Correction Groups:  866
```

Then regroup using the retention time alignment.

```
> tset4a <- group(tset3a)
113 175 238 300 363 425 488 550 613 675 738 800 863
925 988
> tset4b <- group(tset3b)
113 175 238 300 363 425 488 550 613 675 738 800 863
925 988
```

Metabolomics in R

Then take a look at the resulting objects

```
> tset4a
An "xcmsSet" object with 8 samples
Time range:  8.4-748.6 seconds (0.1-12.5 minutes)
Mass range:  50.5756-999.8372 m/z
Peaks:  26159 (about 3270 per sample)
Peak Groups:  3257
Sample classes:  testDat1, testDat2
Profile settings:  method = bin
                   step = 0.1
Memory usage:  4.32 MB
```

And we see fewer peaks using the Gaussian family specification.

Metabolomics in R

```
> tset4b
An "xcmsSet" object with 8 samples
Time range:  7.4-748.7 seconds (0.1-12.5 minutes)
Mass range:  50.5756-999.8372 m/z
Peaks:  26159 (about 3270 per sample)
Peak Groups:  3240
Sample classes:  testDat1, testDat2
Profile settings:  method = bin
                   step = 0.1
Memory usage:  4.32 MB
```

Then after this last grouping we need to determine peak intensities for those samples that don't have a peak at one of the meta-peaks

```
> tset5a <- fillPeaks(tset4a)
```

Metabolomics in R

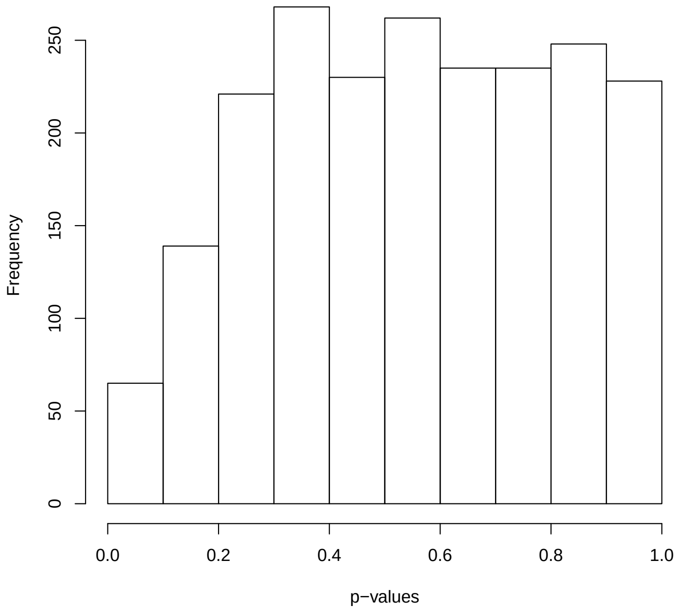
Finally we can generate a report which provides a test for differences between groups.

Here we specify that we are willing to accept a mass difference of 0.15 when trying to determine the identity of a metabolite when looking up the mass in the metlin database (a database for identifying metabolites).

```
> r1a <- diffreport(tset5a, "testDat1", "testDat2",
+ metlin=0.15)
```

Then r1a is a dataframe whose fourth column holds the set of *p*-values for testing for a group difference.

So we can look at a histogram and determine that there is not much going on here.


Frequency
0
50
100
150
200
250
0.0
0.2
0.4
0.6
0.8
1.0
p−values

Biomarkers

A *biomarker* is a quantity that is measured in samples that indicates a biological state.

Many of the things we have discussed in this course can be the basis for developing biomarkers.

Some biomarkers are static while some are dynamic.

Some biomarkers are more invasive than others: the less invasive the better.

The most common use of a biomarker is for guidance of treatment.

Biomarkers

For example, CD4 levels are a commonly used biomarker for HIV positive subjects.

This is easily and reliably measured from blood samples.

CD4 levels are mechanistically linked to HIV progression, and this is also highly desirable.

They change over time with a decline being associated with an increased risk of developing opportunistic infections.

In the past CD4 levels were used to determine the time to initiate anti-retroviral therapy.

Biomarkers

All of these features

1. easy to get
2. reliably measured
3. obvious link to disease
4. you can intervene to prolong patient survival

make CD4 levels an exemplary biomarker.

Biomarkers

CD4 levels are a *surrogate marker* for the health of one's immune system.

By using a well validated surrogate marker one can design studies with endpoints other than "all cause mortality".

There are complex practical and ethical issues involved in studies that use surrogate endpoints.

For example, hard endpoints like "all cause mortality" require that patients die however evaluation of outcomes is much more straightforward.

Biomarkers

CD4 is a protein and a number of other proteins have been suggested for other conditions.

One of the primary problems with biomarker identification in proteomics and metabolomics is determining the identity of the compound that differs between groups.

This is especially problematic for metabolomics as the databases one must use to identify compounds based on m/z and retention time are very much incomplete.

As compound identification is the last step of the entire procedure, this can be frustrating.