Tutorial For Metabolomics Toolbox

BCH 6745 Lab

Modified from Steve Robinette Tutorial

Type commands here

Make sure your data is in this folder so you can upload it

You will find the variables you make stored here

Introduction to Matlab screen:



Step 1: Uploading and processing spectra

First, make sure that the current folder that you are in has your data in it. Then in the command window type:

loadallft

This will bring in data in the format of ‘filename.ft’ which usually has been preprocessed using NMRpipe. There are ways to bring in other file types and this can be found at the end of this tutorial.

You will then need to change the data that you just brought in to a matrix that contains all of your spectra (X), the chemical shift axis (ppm) and the titles of each sample so you know the order they were read in.

[X,ppm,XTitles]=Setup1D(spectra);

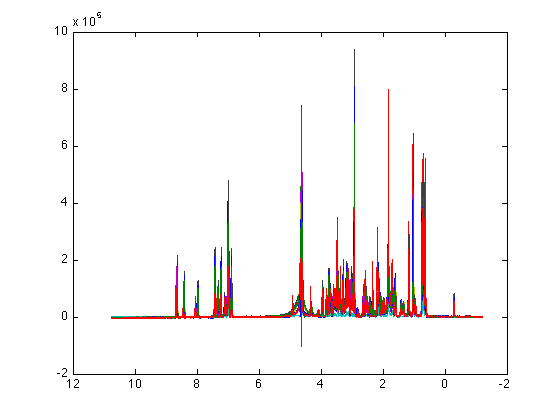
You will then want to set up a vector that will label which spectra belongs to each group. We arbitrarily will use a 0 for Mix A and a 1 for Mix B

y=[0 0 0 0 0 1 1 1 1 1];

Now, to make sure that you have done this correctly, plot the spectra.

figure, plot(ppm,X)

set(gca,'XDir','rev')



We can also look at the two groups separated by different colors

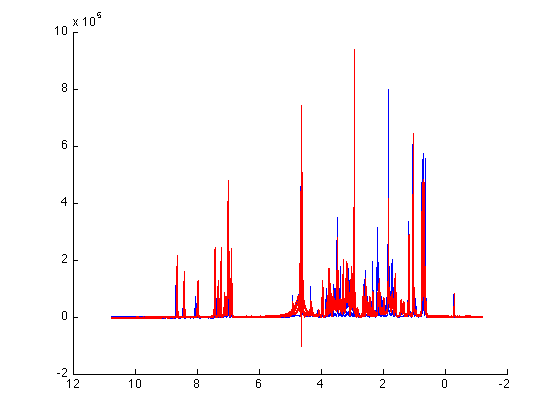
figure

hold

plot(ppm,X(y==0,:),'b')

plot(ppm,X(y==1,:),'r')

set(gca,'XDir','reverse')



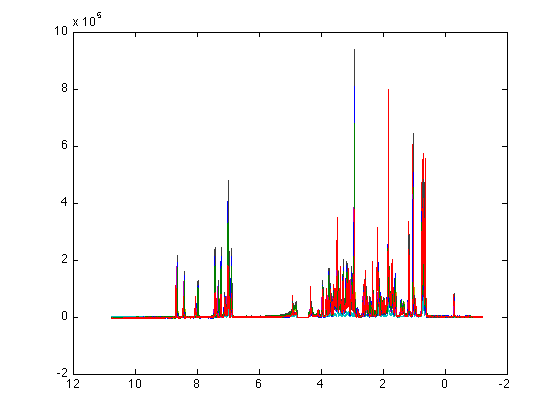
Here we notice that although a PRESAT pulse sequence was used there is still residual water signal. We will remove this signal so that it does not effect how the rest of the data is analyzed.

XR=remove\_region(X,ppm,4.4,4.8);

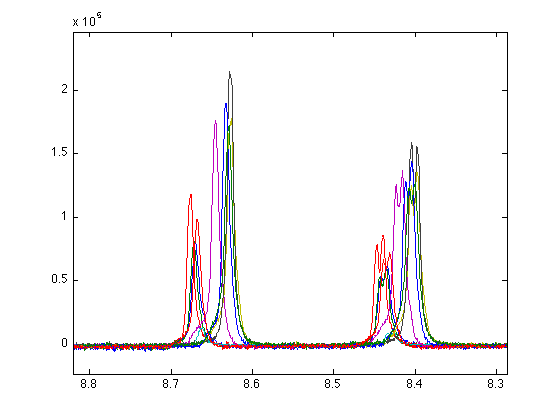
and plot so that you can be sure that the correct region was removed

figure, plot(ppm,XR)

set(gca,'XDir','rev')



If we look closer at the aromatic region we can see some peaks that are not correctly aligned.



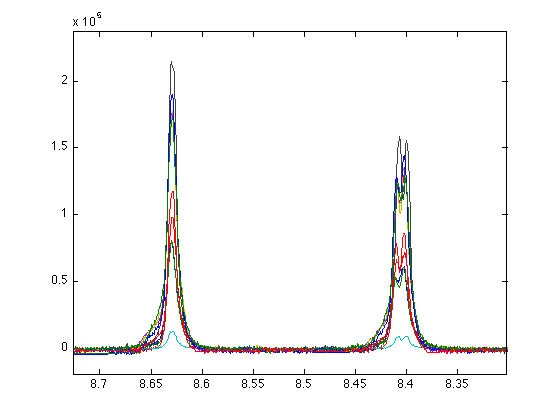
We can align the spectra using:

XAL=star\_align1D(XR,ppm,'mean','PAFFT');

Now plot the aligned spectra and look at the same peaks to make sure that they are now aligned.

figure, plot(ppm,XAL)

set(gca,'XDir','reverse')



Step 2: Normalization and Scaling

The purpose of normalization is to counter for the intensity multiplier effects- normally this means concentration, but can include receiver gain, ion saturation, etc- any effect that will modify approximately all signal intensities by a (linear) multipliernormcheck(XAL)

There are many different ways to normalize your data. To be sure that you are getting the best results you need

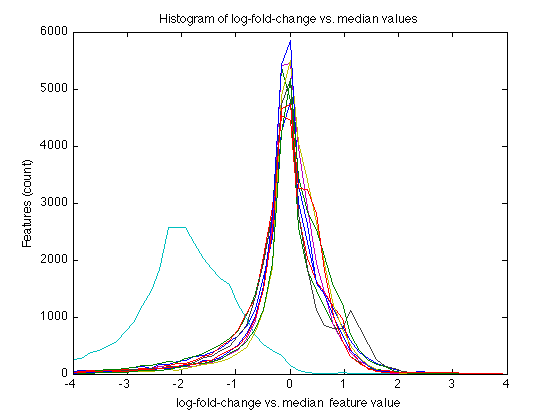
to try them out to see which one works best for your data. We will show 2 examples.

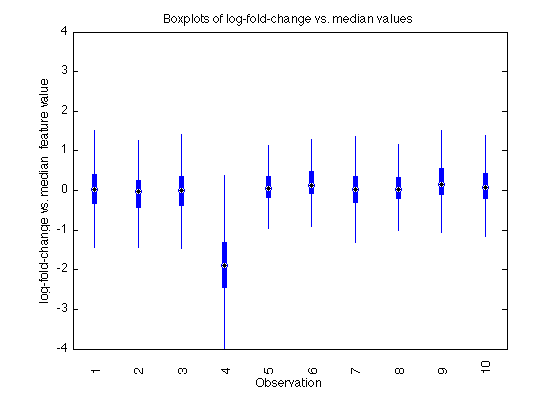
In cases where a large (>50%) percentage of datapoints are linearly increased/decreased in intensity, probabilistic quotient normalization (PQN) works quite well. However, in other cases where normalization to an internal reference, such as TSP or CRN is desired, spectra can be normalized to peak intensities or integrals. Total area normalization is also supported, but tends to be less useful.

Lets first check our normalization.

normcheck(XAL)

You should get something like these two figures. Normcheck plots histograms and barplots of log-fold change vs. median for all spectra. Notice that these spectra are right-shifted with high fold-changes for nearly all datapoints. This distribution indicates a normalization issue





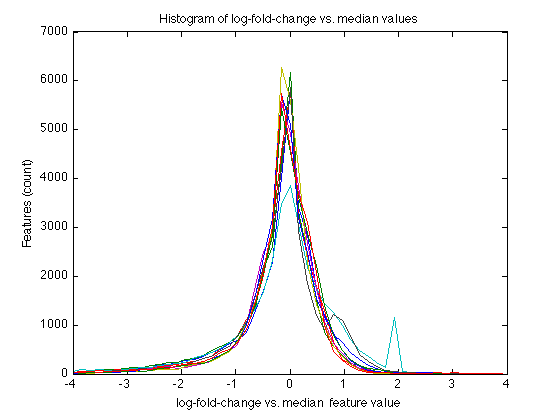
So we can try normalizing using two of the many methods.

**PQN**

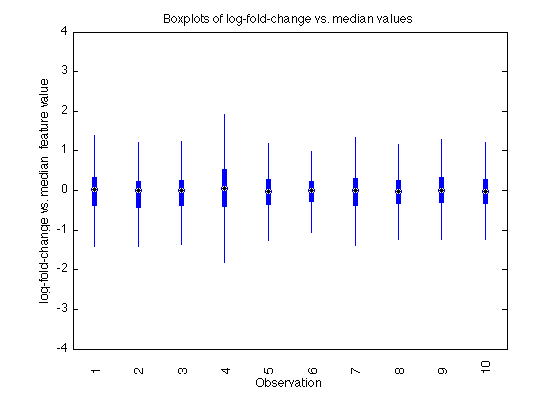
XALN=normalize(XAL,'PQN');

and visualize

normcheck(XALN)



and

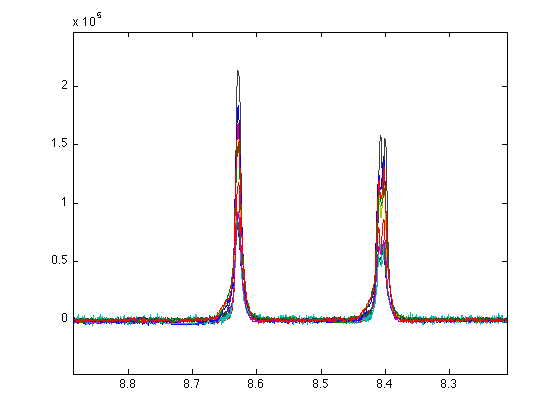


Notice that LFC distributions are now centered at 0.

This resolves previously mentioned with the non-normalized data. Also, potential class differences in this signal are now more notable.

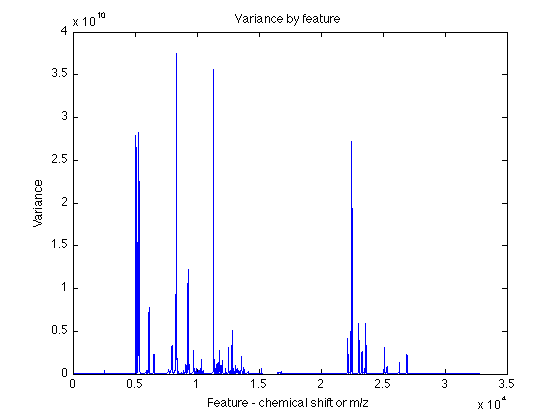
figure, plot(ppm,XALN)

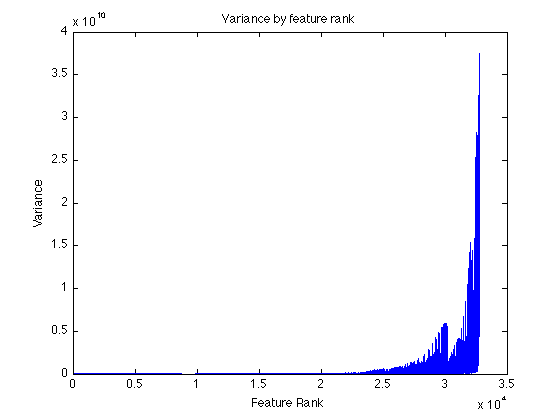
set(gca,'XDir','rev')



Let’s now check the scaling of this data.

varcheck(XALN)





These two plots show signal variance by rank (top) and chemical shift (bottom). They indicate that a small number of intense peaks dominate the variance of the un-scaled dataset.

Lets try different scaling methods

XALSN=scale(XALN,'logoff');

varcheck(XALSN)

XALSN=scale(XALN,'pareto');

varcheck(XALSN)

Step 3: Statistical Analysis and Pattern Recognition

Principle Component Analysis- lets look at 5 components

PCA=nipalsPCA(XALSN,5);

Visulaize the different scores plots to see which component separates the groups best

VisScores(XALSN,PCA,[1 2],y);

VisScores(XALSN,PCA,[1 3],y);

VisScores(XALSN,PCA,[1 4],y);

VisScores(XALSN,PCA,[1 5],y);

VisScores(XALSN,PCA,[2 3],y);

VisScores(XALSN,PCA,[2 4],y);

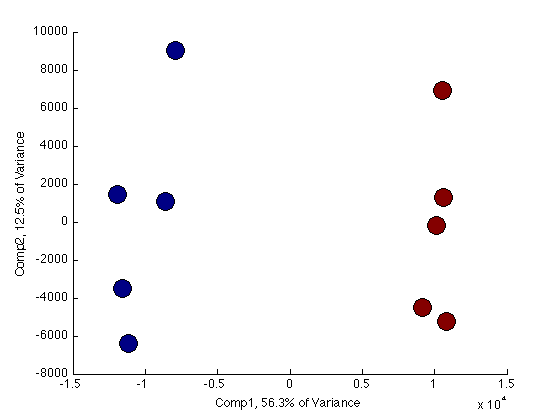
VisScores(XALSN,PCA,[2 5],y);

VisScores(XALSN,PCA,[3 4],y);

VisScores(XALSN,PCA,[3 5],y);

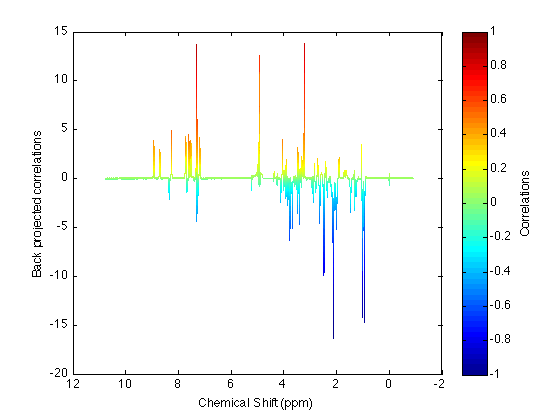
VisScores(XALSN,PCA,[4 5],y);

Example: VisScores(XALN,PCA,[1 2],y);

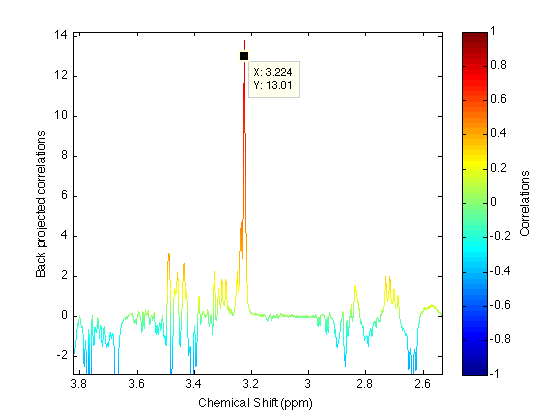


Visualize the loadings plot associated with the scores plot that separates best

VisLoadings1D(XALN,PCA.loadings(1,:),ppm)



Take a closer look at the peaks that are separating the two groups



STOCSY with the peak that separates the two groups from your loadings plot such as 7.306.

STOCSY(7.3,XALN,ppm);

