Proteomics data analysis in cancer biology with Matlab

Table of Contents

Author Information

Felix E. Rivera-Mariani, PhD date: 08/14/2017 Github repository: https://github.com/friveramariani/Proteomic-Examples

Summary

This report represents an example Matlab proteomic data analysis. The dataset analyzed in this report can be found here, which is the FDA-NCI Clinical Proteomics Program Databank. The samples downloaded from the FDA-NIC Proteomics Programa Databank correspond to SELDI Mass-Spec profiles of ovarian cancer samples: **Cancer Group** vs **Normal Group**. The study related to this dataset was published, in 2004, in the Endocrine Related Cancer journal. **Briefly**, after *transforming* the Mass-Spec data, some *variables were initialized* to facilitate the downstream workflow, *visualization of Mass-Spec profiles* performed, and lastly the *features ranked* with t-test statistic.

Loading pre-processed dataset

After preprocessing the dataset into .mat format (find the code here, the dataset was loaded.

```
load OvarianCancerQAQCdataset
whos
```

```
  Name           Size                 Bytes  Class     Attributes

  MZ         15000x1                 120000  double
  Y          15000x216             25920000  double
  grp          216x1                  26784  cell
```

Initializing variables

A set of vector variables, which will be used in the downstream workflow, are initialized.

```
N = numel(grp);                         % vector of number of samples
Cidx = strcmp('Cancer',grp);            % logical index vector for
 Cancer samples' group
Nidx = strcmp('Normal',grp);            % logical index vector for
 Normal samples' group
Cvec = find(Cidx);                      % index vector for Cancer
 samples
Nvec = find(Nidx);                      % index vector for Normal
 samples
xAxisLabel = 'Mass/Charge (M/Z)';       % x-axis label for plots
yAxisLabel = 'Ion Intensity';           % y-axis label for plots
```

Visualizing a set of the samples

Fine below the spectogram of 10 samples. Figure 1 corresponds to original spectogram, while figure 2 to a zoomed spectrogram.

```
figure; hold on;
hC = plot(MZ,Y(:,Cvec(1:10)),'b');
hN = plot(MZ,Y(:,Nvec(1:10)),'g');
xlabel(xAxisLabel); ylabel(yAxisLabel);
axis([2000 12000 -5 60])
legend([hN(1),hC(1)],{'Control Group','Ovarian Cancer'})
title('Figure 1: Spectrograms of 10 Samples')

figure; hold on;
hC = plot(MZ,Y(:,Cvec(1:10)),'b');
hN = plot(MZ,Y(:,Nvec(1:10)),'g');
xlabel(xAxisLabel); ylabel(yAxisLabel);
axis([8000 9000 -1 7])
legend([hN(1),hC(1)],{'Control Group','Ovarian Cancer'})
title('Figure 2: Zoomed Spectrograms of 10 Samples')
```

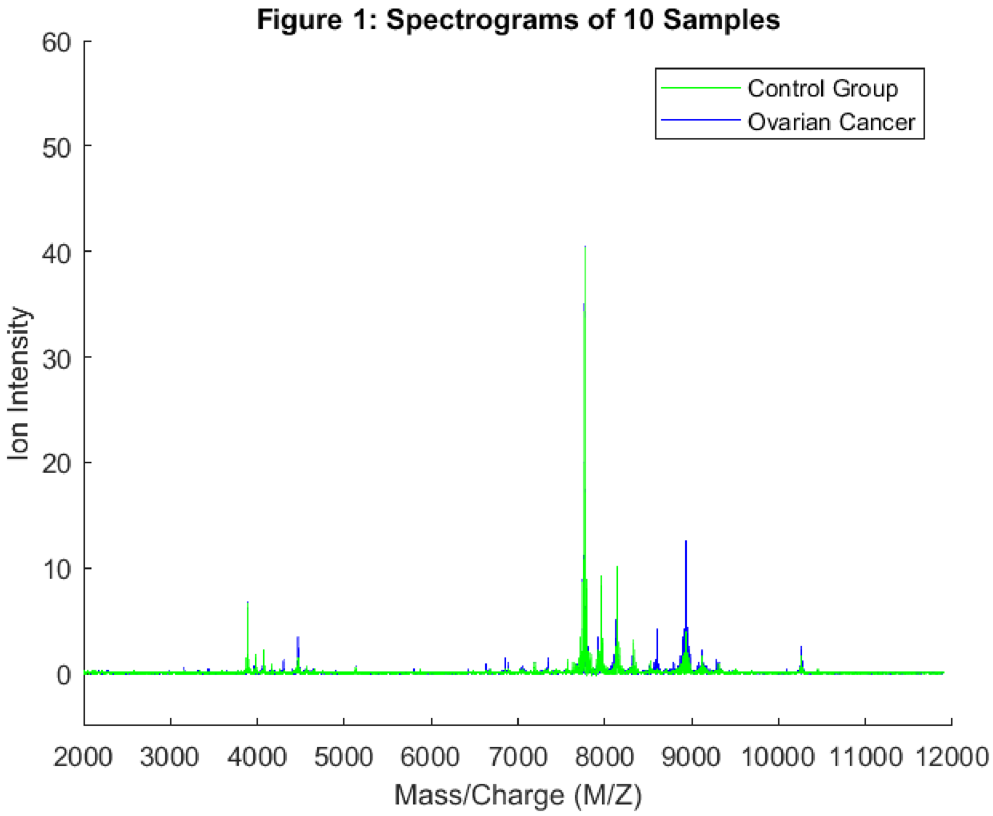


Figure 1: Spectrograms of 10 Samples
Control Group
Ovarian Cancer
Ion Intensity
60
50
40
30
20
10
0
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000
12000
Mass/Charge (M/Z)

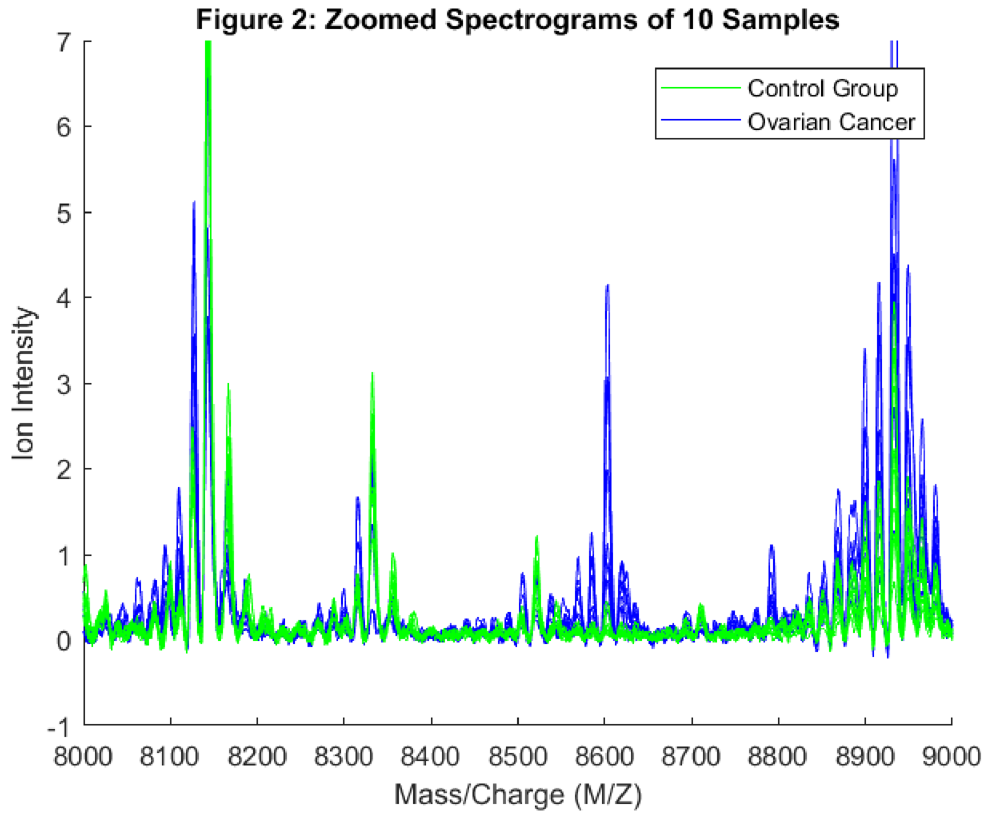


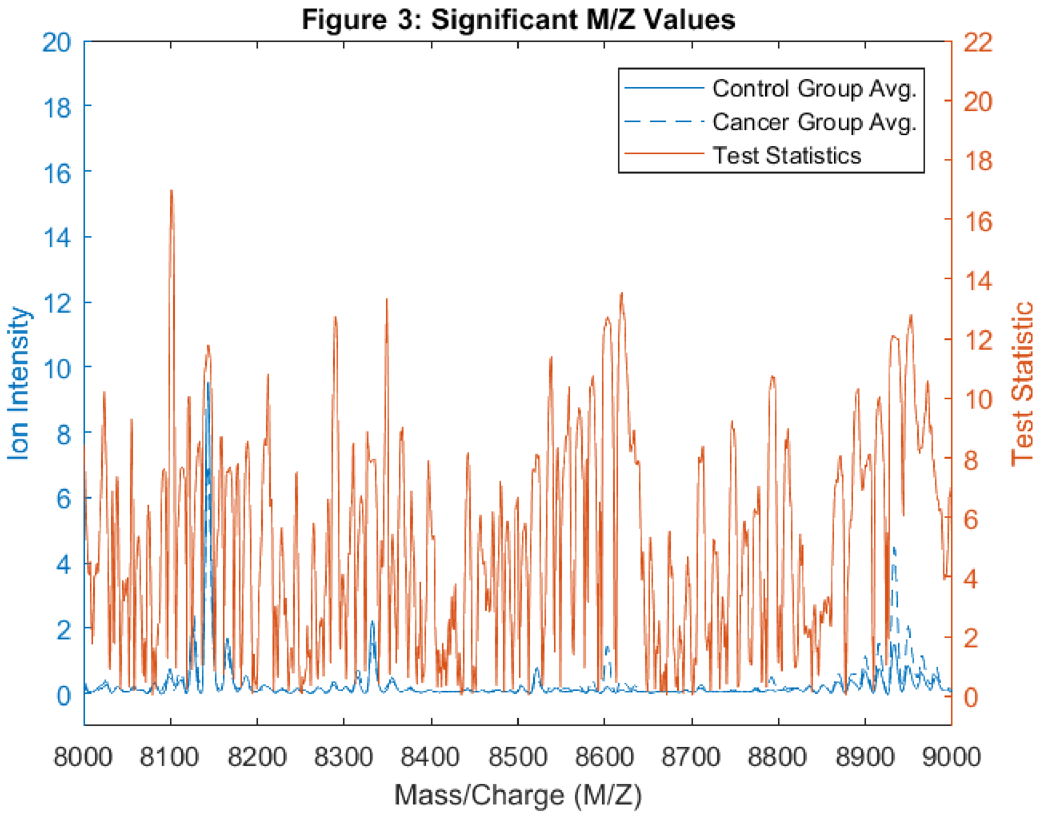
Figure 2: Zoomed Spectrograms of 10 Samples
Control Group
Ovarian Cancer
Ion Intensity
7
6
5
4
3
2
1
0
-1
8000
8100
8200
8300
8400
8500
8600
8700
8800
8900
9000
Mass/Charge (M/Z)

Ranking features

Significant masses were identified using a two-way t-statistic. After ranking the features, a set of variables were initialized to generate the plot (figure 3) for the spectogram with two-way t-statistic.

```
[feat,stat] = rankfeatures(Y,grp,'CRITERION','ttest','NUMBER',100);
sig_Masses = MZ(feat);
sig_Masses(1:10)' %display the first 10 significant masses

mean_N = mean(Y(:,Nidx),2);  % group average for control samples
max_N = max(Y(:,Nidx),[],2); % top envelopes of the control samples
min_N = min(Y(:,Nidx),[],2); % bottom envelopes of the control samples
mean_C = mean(Y(:,Cidx),2);  % group average for cancer samples
max_C = max(Y(:,Cidx),[],2); % top envelopes of the control samples
min_C = min(Y(:,Cidx),[],2); % bottom envelopes of the control samples

figure;

yyaxis left
plot(MZ, [mean_N mean_C]);
ylim([-1,20])
xlim([8000,9000])
title('Figure 3: Significant M/Z Values')
xlabel(xAxisLabel);
ylabel(yAxisLabel);

yyaxis right
plot(MZ,stat);
ylim([-1,22])
ylabel('Test Statistic');

legend({'Control Group Avg.','Cancer Group Avg.', 'Test Statistics'})
```

```
ans =

   1.0e+03 *

  Columns 1 through 7

    8.1009    8.1016    8.1024    8.1001    8.1032    7.7366    7.7359

  Columns 8 through 10

    7.7374    7.7253    7.7245
```



Other possible approaches

Although not performed in this example proteomic data analysis, other approaches that could have been added include:

1. identify the amino acid sequences of the statistically significant features
2. and identify the proteins by matching amino acid sequences to databases.

In future *"omics" data analysis in Matlab, as well as in R and Python more thorough and detailed workflow will be shared.

Published with MATLAB® R2017a