Supervised analysis of MS images using Cardinal

Kyle D. Bemis and April Harry

November 25, 2014

Contents

1 Introduction

For experiments in which analyzed samples come from different classes or conditions, a common goal in analysis is to predict the class of a new sample, given a labeled training set for which sample classes is known. This task is called classification. *Cardinal* implements a variety of classification methods for mass spectrometry imaging experiments.

2 Analysis of a renal cell carcinoma (RCC) cancer dataset

```
> library(CardinalWorkflows)
> data(rcc, rcc_analyses)
```

[4Kyle: Discuss renal cell carcinoma (RCC) dataset.]

2.1 Pre-processing

[TODO: Mention why peak-picking is not done.]

We begin by pre-processing the data. [4Kyle: for more information on preprocessing methods, see...]The spectra are first normalized to total ion current.

```
> rcc.norm <- normalize(rcc, method = "tic")
```

The normalized data is then resampled to unit resolution.

```
> rcc.resample <- reduceDimension(rcc.norm, method = "resample")
```

2.2 Visualizing the dataset

2.2.1 Visualization of single-ion images

We can plot the diagnosis from pathology for three tissue samples in Figure 1A. [4Kyle: In the code below, where does *diagnosis* come from?] We color cancerous tissue red, while normal tissue is colored blue.

```
> image(rcc.resample, formula = diagnosis ~ x * y, subset = sample %in%
+     c("UH9610_15", "UH9812_03", "UH9912_01"), col.regions = c("red",
+     "blue"), colorkey = FALSE, layout = c(3, 1))
> legend("topright", legend = c("cancer", "normal"), fill = c("red",
+     "blue"))
```

[4Kyle: IS THIS TRUE: In order to display normal tissue in Figure 1B, we plot the single ion image of $m/z215$, an ion known to be more abundant in normal tissue than cancer.]

```
> image(rcc.resample, mz = 215, subset = sample %in% c("UH9610_15",
+     "UH9812_03", "UH9912_01"), contrast.enhance = "histogram",
+     smooth.image = "gaussian", layout = c(3, 1))
```

[4Kyle: IS THIS TRUE: Likewise for cancer tissue in Figure 1C, we plot the single ion image of $m/z886$, an ion known to be more abundant in cancer than normal.]

```
> image(rcc.resample, mz = 886, subset = sample %in% c("UH9610_15",
+     "UH9812_03", "UH9912_01"), contrast.enhance = "histogram",
+     smooth.image = "gaussian", layout = c(3, 1))
```

2.2.2 Exploratory analysis using PCA

We perform Principal Component Analysis on the data.

```
> rcc.pca <- PCA(rcc.resample, ncomp = 5)
```

Scores at each pixel for the first three components are displayed in Figure 2A. [4Kyle: provide some interpetation]

```
> image(rcc.pca, column = c("PC1", "PC2", "PC3"), subset = sample ==
+     "UH9610_15", superpose = FALSE, col.regions = risk.colors(100),
+     layout = c(3, 1))
```

We can also plot loadings for the first three components, as in Figure 2B. [4Kyle: provide some interpetation]

```
> plot(rcc.pca, column = c("PC1", "PC2", "PC3"), superpose = FALSE,
+     layout = c(3, 1))
```

2.3 Classification using PLS-DA

We use cross-validation to find best number of components in partial least squares discriminant analysis.

```
> rcc.cv.pls <- cvApply(rcc.resample, .y = rcc.resample$diagnosis,
+     .fun = "PLS", ncomp = 4:12)
```

Check the cross-validated accuracy.

```
> rcc.acc <- data.frame(ncomp = 4:12, Accuracy = sapply(summary(rcc.cv.pls)$accuracy,
+     function(x) x[1, 1]))
> plot(Accuracy ~ ncomp, data = rcc.acc, type = "b", xlab = "Number of Components")
> abline(v = 10, col = "red", lty = 2)
```

Use 10 PLS components.

```
> rcc.pls <- PLS(rcc.resample, y = rcc.resample$diagnosis, ncomp = 10)
```

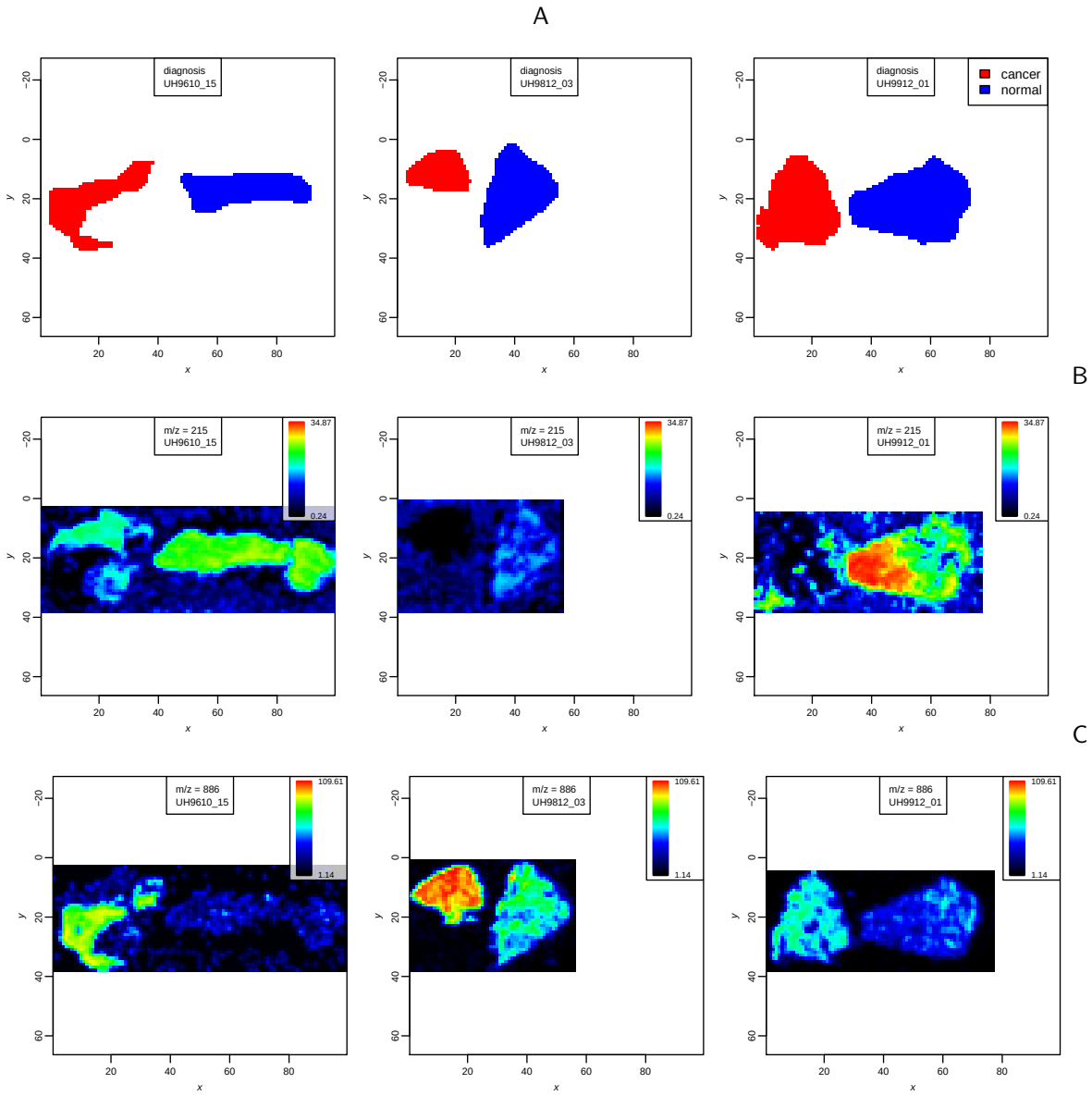

Figure 1: (A) Diagnosis as determined by histopathology. (B) Ion image a m/z 215 showing normal healthy tissue. (C) Ion image at m/z 886 showing cancer tissue.

2.3.1 Plotting the classified images

Plot the fitted values.

```
> image(rcc.pls, subset = sample %in% c("UH9610_15", "UH9812_03",
+     "UH9912_01"), layout = c(3, 1))
```

2.3.2 Plotting the coefficients of m/z values

```
> plot(rcc.pls)
```

```
> plot(rcc.pls, mode = "loadings")
```

```
> plot(rcc.pls, mode = "weights")
```

A

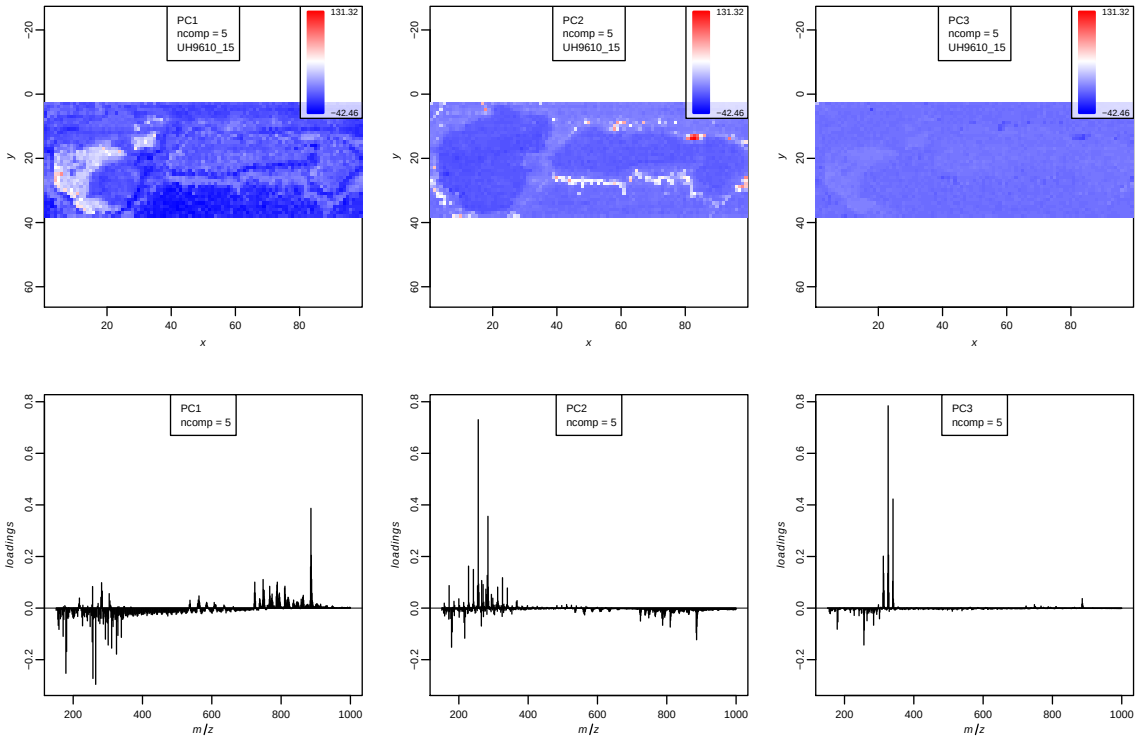

Figure 2: (A) Images of the PCA scores for the first three principal components. (B) Plot of the PCA loadings for the first three principal components.

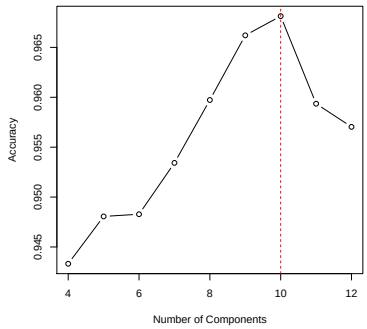

Figure 3: Accuracy of PLS classification for number of components used.

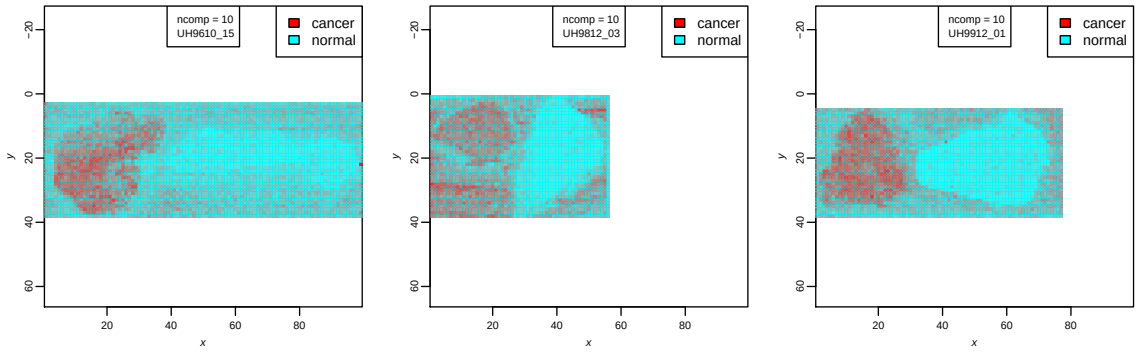

Figure 4: Plot of the fitted values indicating cancer or normal tissue.

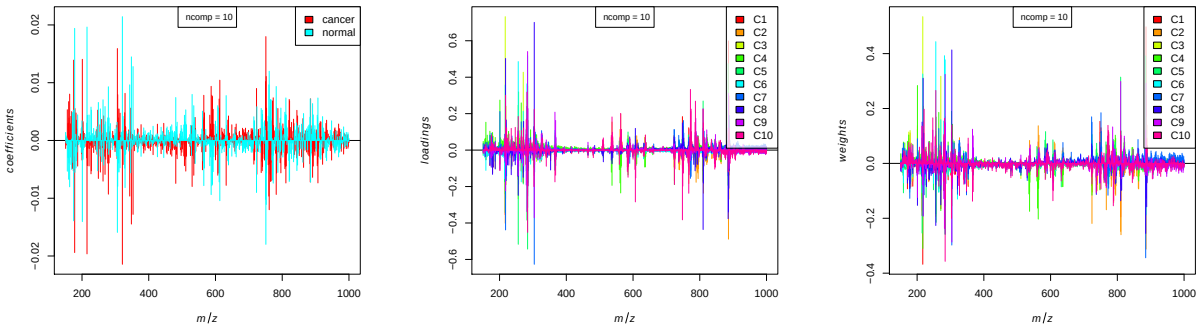

Figure 5: (A) Plot of PLS coefficients. (B) Plot of PLS loadings. (C) Plot of PLS weights.

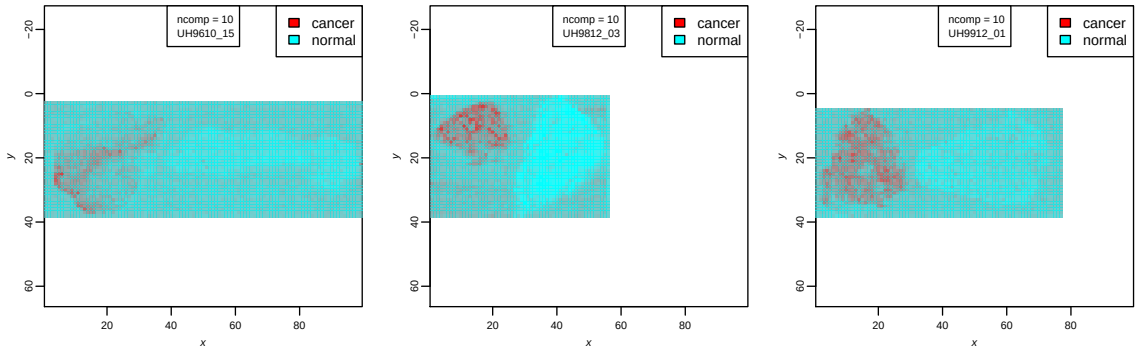

Figure 6: Plot of the fitted values indicating cancer or normal tissue.

2.4 Classification using O-PLS-DA

Comparable accuracy with improved interpretability with orthogonal partial least squares discriminant analysis.

```
> rcc.opls <- OPLS(rcc.resample, y = rcc.resample$diagnosis, ncomp = 10,
+     keep.Xnew = FALSE)
```

```
> image(rcc.opls, subset = sample %in% c("UH9610_15", "UH9812_03",
+     "UH9912_01"), layout = c(3, 1))
```

Show that the coefficients are more stable and interpretable.

```
> plot(rcc.opls)
```

```
> plot(rcc.opls, mode = "loadings")
```

```
> plot(rcc.opls, mode = "weights")
```

3 Session info

- R version 3.1.2 (2014-10-31), x86_64-apple-darwin13.4.0
- Locale: en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
- Base packages: base, datasets, graphics, grDevices, methods, parallel, stats, utils
- Other packages: Biobase 2.24.0, BiocGenerics 0.10.0, Cardinal 0.8.7, CardinalWorkflows 0.3.0
- Loaded via a namespace (and not attached): BiocStyle 1.2.0, fields 7.1, grid 3.1.2, irlba 1.0.3, lattice 0.20-29, maps 2.3-9, MASS 7.3-35, Matrix 1.1-4, signal 0.7-4, sp 1.0-15, spam 1.0-1, stats4 3.1.2, tools 3.1.2

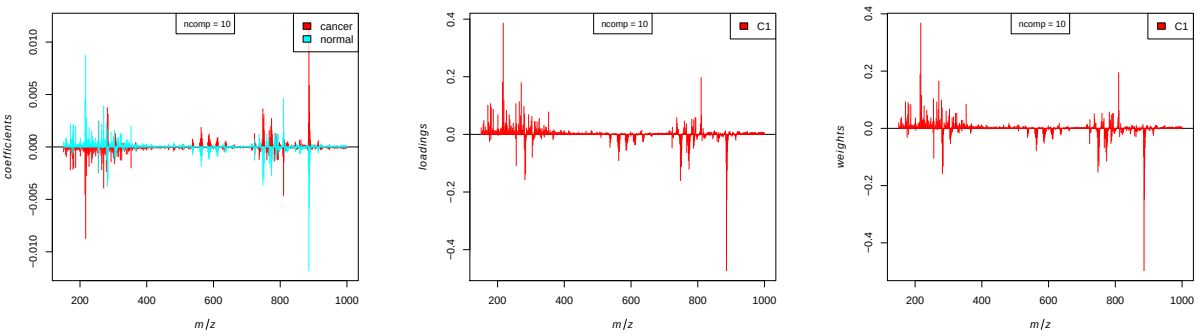

Figure 7: (A) Plot of O-PLS coefficients. (B) Plot of O-PLS loadings. (C) Plot of O-PLS weights.