Cancer Dataset : LASSO, RIDGE, PCR

1. EXPLORING THE DATA

1.1 Explore the dataset: How many variables? What do they represent? How many samples? What do these samples represent

188 rows :

- different breast cancer tumors

4949 columns :

- 4948 gene expression levels (continuous)
- 1 response variable : categorical : 'DM' : Distant metastasis / 'NODM' : No Distant Metastasis

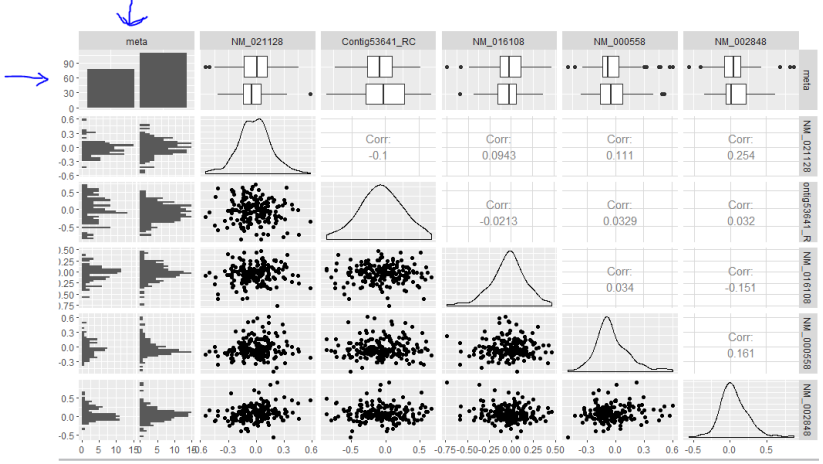
1.2 What challenges do you foresee in using gene expression for the stated goal (predict distant metastases) ?

The number of parameters is much larger than the number of observations. Just using standard linear regression on all parameters is therefore not possible. In any case the number of parameters will have to be reduced. But even then we will have to be careful. Having a lot of predictors will most likely give rise to collinearity of parameters. The final model will not be unique, it's one among many. Another set of genes could have the same predictive power.
Also, every parameter will also add some extra noise to the model, showing signs of association to the response where in fact there is none. To separate the noise from the signal in these high dimensions you need a huge amount of data. In the end, we will have to be modest in our conclusions.

1.3 For a couple of genes evaluate association with the phenotype. Do you see proof for some predictive potential? Test your intuition with a formal statistical test.

When I take 5 genes at random, I see no (visual) indication of an association

```
> set.seed(1)
> rndCols<-sample(seq(2:ncol(data)), 5, replace = FALSE)
> rndTestData<-cbind(meta=data[,"meta"],data[,rndCols])
> ggpairs(rndTestData)
```



This is confirmed by doing some hypothesis tests. I did an analysis of deviance test (anova test). You see that no gene explains a significant amount of the deviance.

```
> anova(lr_mod, test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: meta

Terms added sequentially (first to last)


              Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                            187     255.15
NM_021128      1  0.30148       186     254.85   0.5830
Contig53641_RC 1  1.80249       185     253.05   0.1794
NM_016108      1  1.19556       184     251.85   0.2742
NM_000558      1  0.07592       183     251.77   0.7829
NM_002848      1  0.58793       182     251.19   0.4432
>
```

I come to the same conclusion when I look at the p-values associated with the coefficients of the logistic fit. No one coefficient is significant.

```
> summary(lr_mod)

Call:
glm(formula = meta ~ ., family = "binomial", data = rndTestData)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7215  -1.2777   0.9206   1.0347   1.4563

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)      0.3746     0.1659   2.258    0.024 *
NM_021128        0.4738     0.8812   0.538    0.591
Contig53641_RC  -0.6136     0.4827  -1.271    0.204
NM_016108        0.7207     0.7754   0.929    0.353
NM_000558       -0.1435     0.8439  -0.170    0.865
NM_002848       -0.6105     0.7986  -0.765    0.445
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 255.15  on 187  degrees of freedom
Residual deviance: 251.19  on 182  degrees of freedom
AIC: 263.19
```

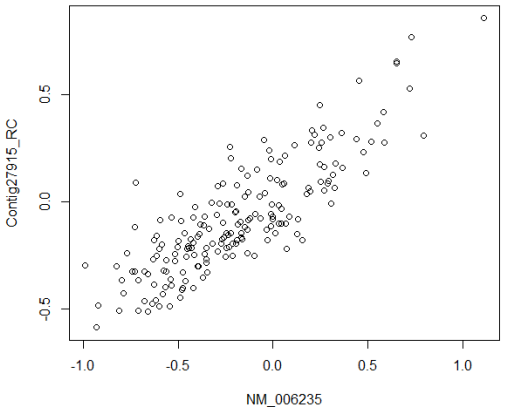
1.4 Demonstrate if collinearity occurs between genes in this dataset. Do you think this represents a challenge in the analysis?

Collinearity is a problem in this dataset.

If I take a correlation of > 0.8 as a cut-off point, I find 1667 variables (of the 4948) that are collinear

```
> corMat<-cor(data[,-1])
> (sum(abs(corMat) > 0.8)-4948)/2  #don't count diagonal, and mirror correlation
[1] 1667
```

For example



Collinearity leads to imprecise estimates of the coefficients. Too many variables are trying to do the same job of explaining the response. This causes the estimates to become numerically unstable. So slight changes in the inputdata or parameters of the fit can cause major changes in the coefficientvalues. A coefficient once labelled as significant, can 'flip', and become insignificant.

2. MAKING PREDICTIONS

2.1 Use lasso, ridge and PCR methodology and make a predictor based on the gene expression values. How many genes are used for an optimal predictor? Evaluate the performance of the predictors, and comment on what you find.

I have split the dataset 75/25 into train and testset :

- 143 rows = trainingset = used to train the model.
  A 10-fold cross-validation was used to determine the optimal value for lamba (lasso, ridge), and M (the number of principal components, in case of PCR)

- 45rows = testset = used to compare the performance of the 3 predictors

In the analysis "DM" (distant metastasis) is represented by 1, "NO DM" by 0.
The threshold p-value is 50% (so a prediction with p>0.5 is classified as "DM")

A summary of the results :
Lasso :

- 12 genes in the final model
- Test performance :
  - 37.8 % misclassification
  - AUC : 0.6166008 (area under ROC curve)

Ridge :

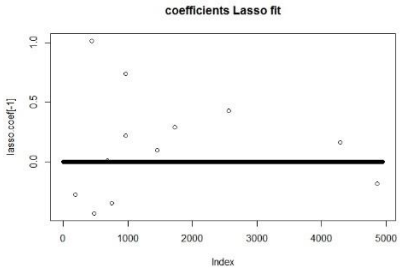
- 4949 genes in the final model
- Test performance :
  - 42.2 % misclassification
  - AUC : 0.5731225

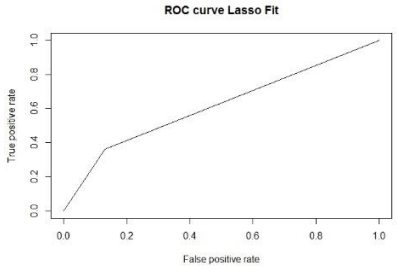
PCR

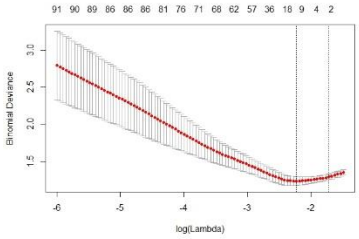
- 29 principal components in the final model

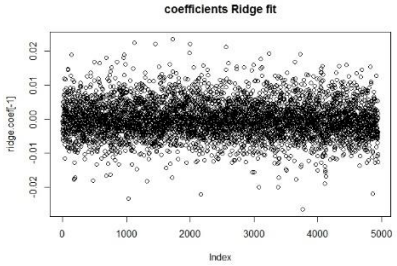
- Test performance :
    - 40.0 % misclassification
    - AUC : 0.5978261

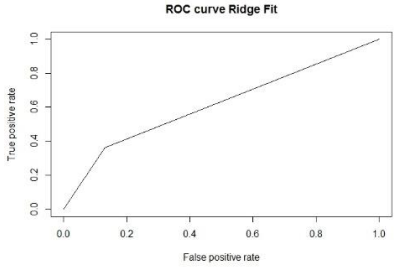
Of the three models, lasso shows the best results. But the differences are minor. Overall the predicition performance is pretty poor. A 37% misclassification rate is not much better then random chance

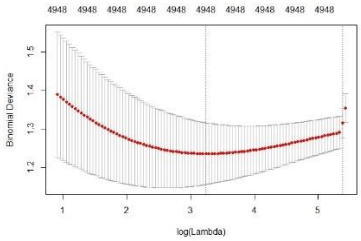


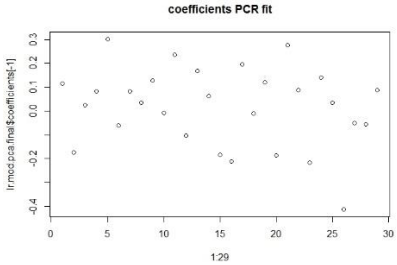


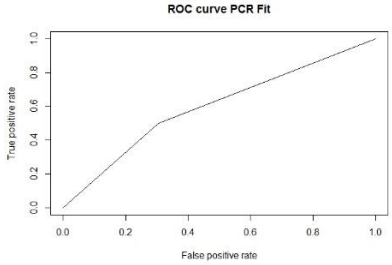


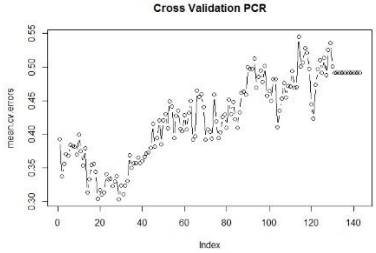








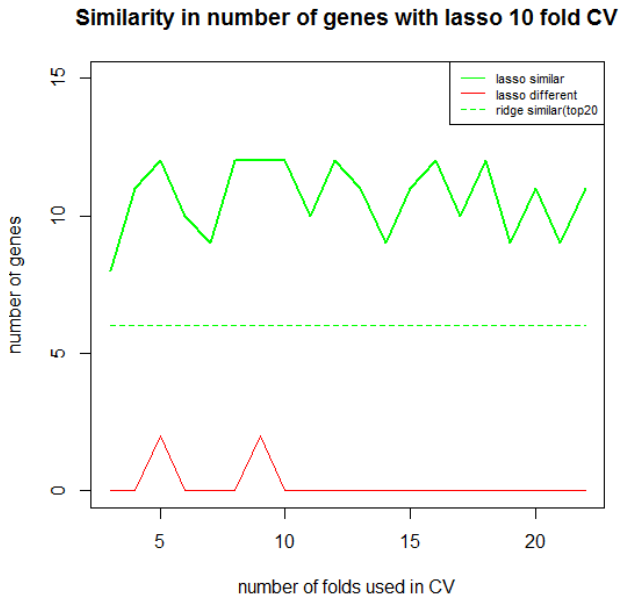




2.2 Repeat lasso and ridge for different cross-validation folds (partitions of the dataset into training and test set): Do the predicted coeffiicients change, or in the case of lasso, does the set of selected genes change? Are some genes consistently chosen across folds? Explain your observations.

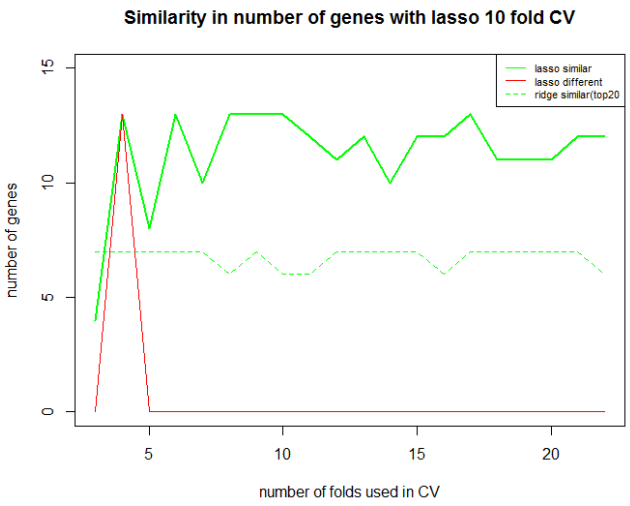
I first kept the 75/25 split into train and test samples like above, but used different folds to estimate lambda, (instead of the default 10 fold cross validation)

In the original lasso fit 12 genes were included, and I looked at which of those 12 genes also showed up when using different folds (from 3 folds , up to 20 folds)
I also performed ridge regression with these different folds. Here I only considered the 20 genes with biggest coefficients.



The different lasso fits more or less kept selecting the same genes, although some fits ended up with 8 instead of 12 genes. Ridge consistently selected 6 genes similar to lasso (also implying that the other 6 genes were never present in the top 20)

When I used a 50/50 split between training and test set (instead of 75/25) I got a similar pattern



Overall 1 pattern keeps repeating : there seem to be around 6 genes that keep reappearing in the different fits

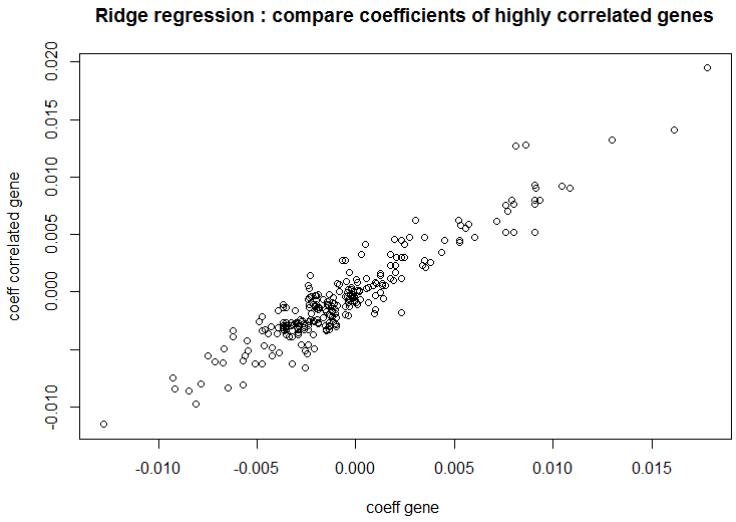
```
[1] "NM_006054"      "Contig51519_RC" "Contig49670_RC" "NM_007267"
[5] "NM_002811"      "NM_002688"
```

2.3 Look at the highly correlated genes in the dataset (correlation > 0.9): what happens to the coefficients lasso and ridge assign to them? If one is used as a predictor in lasso, is a highly correlated gene also found? Interpret your observations.

I found 244 pairs of highly correlated genes in the dataset.

Effect in Lasso : None of those highly correlated genes were selected by my lasso fit, so I could not check this directly . What I would suspect is that in lasso, only 1 gene of these pairs would be present in the lassomodel. The coefficient of the other would be pushed to zero. Which one lasso would favor would be impossible to predict, and altering the parameters or data slightly could make this prediction 'flip', selecting the other gene instead.

Effect in Ridge : In ridge you don't see this on/off effect like in lasso, and the highly correlated genes will be assigned more or less equal weights. So ridge regression will behave more stable in this scenario. The plot below demonstrates that the coefficients are indeed very similar.


Ridge regression : compare coefficients of highly correlated genes
coeff correlated gene
-0.010
0.000
0.005
0.010
0.015
0.020
-0.010
-0.005
0.000
0.005
0.010
0.015
coeff gene