Assingment

Ivana Janickova (r0816203)

1. Exploring the data

Potential isues when statistically modelling data

- High dimensionality p > n. The fact that there is larger number of predictors then observations causes OLS to not have a unique solution.
- Many of the predictors are likely to be correlated.
- Fitting a model with high number of predictors decreases the interpretability of the model.
- Large number of predictors means large number of degrees of freedom which inevitably leads to high variance of the model and consequently large test set error (failure of model to generalize).

Data Preparation

The `cortex` data set was split into train & test set in the ration train set = 2/3 of data; test = 1/3 of data. The categorical variable `Behaviour` transformed into a dummy variable as it is later used as a response variable.

Exploring the Correlations in the dataset

The correlation matrix was constructed with aim to explore correlations between the expression levels of different genes. The correlation matrix produces 48300 pairs of non-equal proteins (i.e. proteins with correlation equal to 1). From these pairs the percentage of pairs that are more than 90% correlated is 0.62% and the percentage of pairs that are more than 50% correlated is 16.77%. This result is supprising since given the nature of the data - measurements of protein expression level in different conditions - the expected percentage of highly correlated predictors was larger.

The two heatmaps were constructed from the correlation matrices (only the partialis included in the report *Figure 1*). The first heatmap contains all of the protein expression levels. In order to zoom in on and make the visualization more meaningful a second heatmap for sample of 10 proteins was constructed. The light blue color displays regions of higher correlations of protein expression levels.

2. Training Ridge and LASSO models

2.1 Ridge model

A logistic regression model using ridge regularization was used to find out how efficient is set of predictors to predict `Behavior` variable.

2.1.1 Searching for optimal model with cross-validation

From the Cross-Validation plot it can be observed that a small value of lambda (`best.lambda = 4.143532`) minimizes the deviance. On the right plot we can see that with the rising value of *log(lambda)* the deviance is sharply increasing. The plot resembles a logarithmic curve. The plot on the left displays the change of coefficients value with respect to change in lambda. Each curve corresponds to a single predictor.

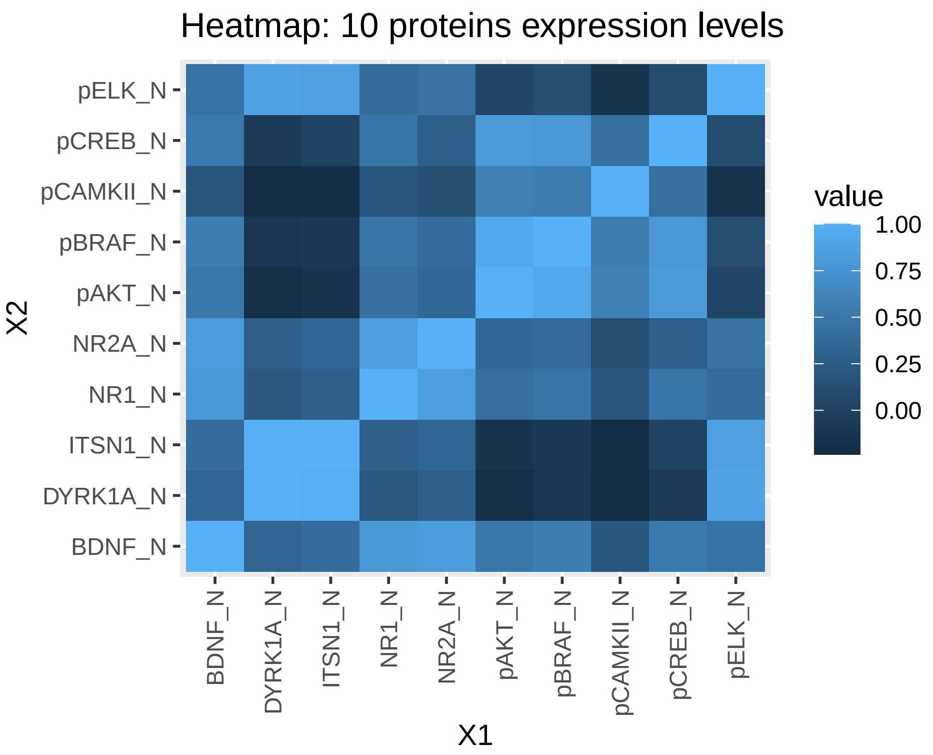


Figure 1: Heatmap of correlations between 10 protein expression levels

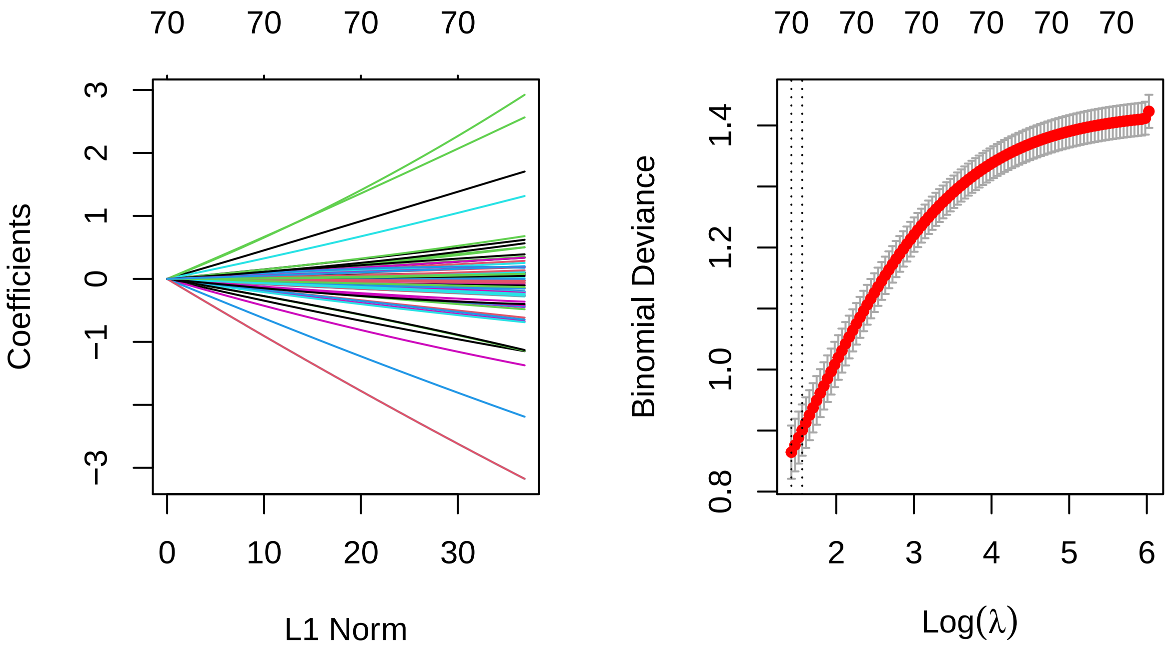


Figure 2: Ridge classification model

2.1.2 Displaying the values of chosen coefficients

For better visualization of the most relevant predictors a barplot was used. The predictors with the largest absolute value are contributing with the highest wight to the model.

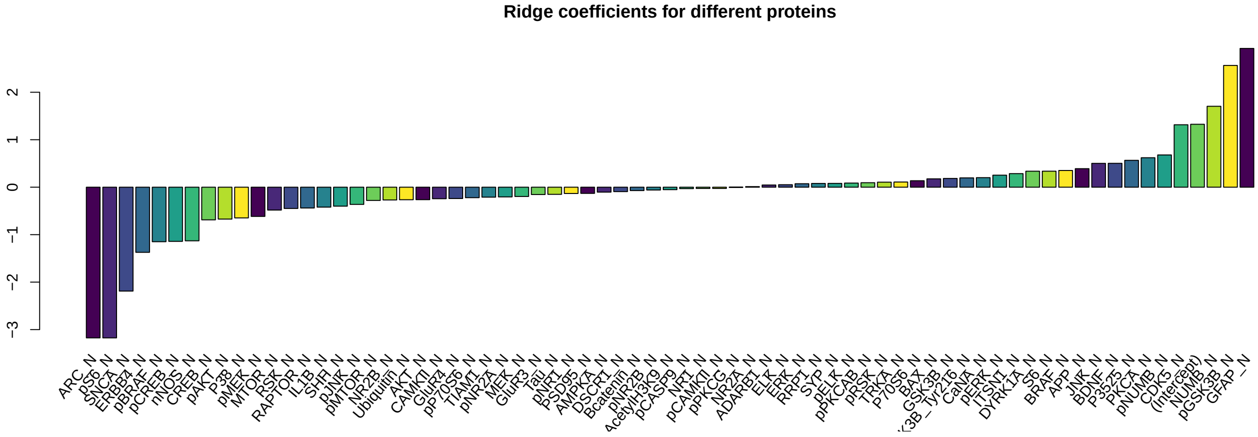


Figure 3: Values of RIDGE model coefficients

2.1.3 What is the predictive power of the model?

Dummy encoding of the categorical variable: C/S = 1, S/S = 0.

The table below displays counts of:

- True negative results = 11
- True positive results = 9
- False negative results = 0
- False positive results = 4

The total performance of the model calculated as *(TN count + TP count)/total count* is approximately 83%.

```
ridge.pred=predict(ridge.model ,s=best.lambda ,newx=x.test,type="response")
predictions=rep(0 ,length(y.test))
predictions[ridge.pred>0.5]= 1
table(y.test,predictions)
```

```
##       predictions
## y.test  0  1
##      0 11  4
##      1  0  9
```

```
performanceRidge=length(which(predictions==y.test))/length(y.test)
performanceRidge #11+9)/24 = 83%
```

```
## [1] 0.8333333
```

2.1.4 Selected proteins

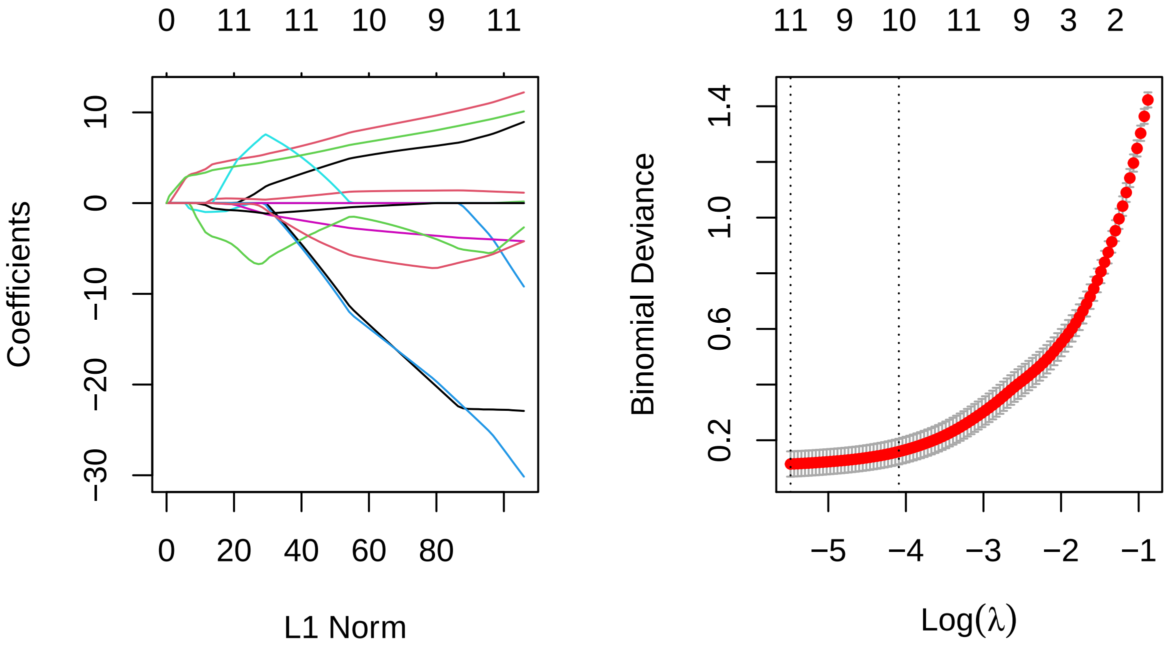
The Ridge regularization method does not perform a variable selection as it is the case with the LASSO - hence all of the 70 predictors are `selected`.

2.2 LASSO model

A logistic regression model using LASSO regularization was used to find out how efficient is set of predictors to predict `Behavior` variable. The expectation is to find a subset of predictors that would be sufficient for this classification problem.

2.2.1 Searching for optimal model with cross-validation

From the Cross-Validation plot it can be observed that a small value of lambda (`best.lambda = 0.00414`) minimizes the deviance. With the rising value of *log(lambda)* the deviance is slowly increasing for smallest values of *log(lambda)*. With the increasing *log(lambda)* value, the slope is increasing. This plot - as opposed to the ridge mode CV plot - resembles an exponential curve.



###

2.2.2 Displaying the values of chosen coefficients

For better visualization of the most relevant predictors a barplot was used. The predictors with the largest absolute values are contributing with the highest weight to the model.

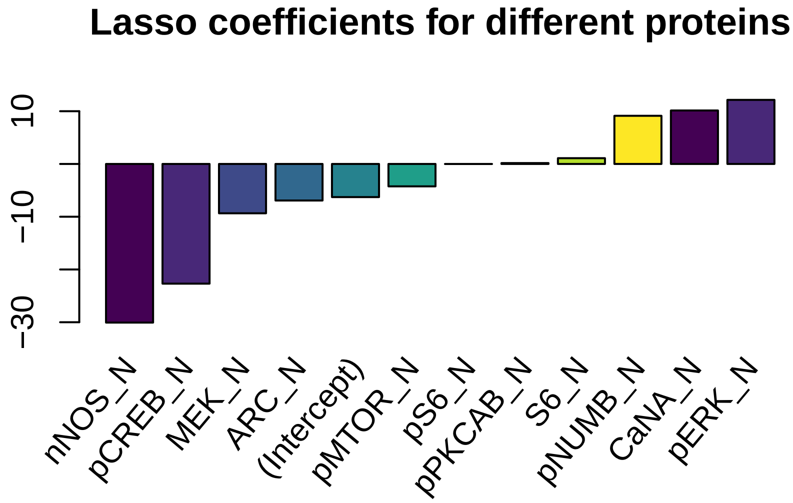


Figure 4: Values of LASSO model coefficients

2.2.3 What is the predictive power of the model?

Dummy encoding of the categorical varible: C/S = 1, S/S = 0.

The table below displays counts of:

- True negative results = 14
- True positive results = 9
- False negative results = 0
- False positive results = 1

The total performance of the model calculated as *(TN count + TP count)/total count* is approximately 96%.

```
lasso.pred=predict(lasso.model ,s=best.lambda ,newx=x.test,type="response")
predictions=rep(0 ,length(y.test))
predictions[lasso.pred>0.5]= 1
table(y.test,predictions)
```

```
##       predictions
## y.test  0  1
##      0 14  1
##      1  0  9
```

```
performanceLasso=length(which(predictions==y.test))/length(y.test)
performanceLasso #14+9)/24 = 95,8%
```

```
## [1] 0.9583333
```

2.2.4 Selected proteins

The Lasso model selects 11 predictors. It can do so by setting the value of other coefficients equal to zero. Below, there is a list of the selected predictors.

```
vals=predict(model.lasso,s= best.lambda,type="coefficients")
selected=colnames(x)[vals@i]
selected
```

```
##  [1] "pCREB_N"  "pERK_N"   "pPKCAB_N" "MEK_N"    "pMTOR_N"  "pNUMB_N" 
##  [7] "S6_N"     "ARC_N"    "nNOS_N"   "pS6_N"    "CaNA_N"
```

2.3 Comparison of Lasso and Ridge Models

The test-set performance of Lasso model was much higher than the perfomance of the Ridge model. Moreover the final Lasso model contains only 11 predictors and hence allows for better interpretablity. From the values of the coefficients displayed in *Figure 4* we can see that predictors `nNOS_N` and `pCREB_N` have the largest weight in the classification model.

2.3.1 Do correlations between variables influence the results? How?

From the correlation we found the names of all predictors that are more than 90% correlated with at least one another predictor. Those are:

`"ITSN1_N"  "pERK_N" "BRAF_N" "DYRK1A_N" "pNR1_N"  "pBRAF_N"  "pMEK_N" "pAKT_N"  "CREB_N" "NR1_N" "pELK_N" "MEK_N" "JNK_N"   "NR2B_N" "MTOR_N" "pS6_N" "ARC_N"`

We would not expect both of the highly correlated partners to have assigned a high coefficient value, as when one is present in the model, the other (highly correlated) predictor does not contribute with much new information to the model. Hence there is a little intersection between highly correlated variables and the selected variables by LASSO; the resulting intersection is: `"MEK_N"  "ARC_N"  "pS6_N"  "pERK_N"`

We can verify the assumption of absence of both highly correlated partners in the lasso-selected variables by scanning through the `ordered` data frame of pairs with their correlation values. Among those that are more that 90% correlated we indeed do not observe any pairs of variables from the resulting intersection (4 proteins listed above).

To conclude the correlation between the variables influences the result - the more are the variables correlated,the fewer variables will be assigned non-zero weight in lasso or higher weights in the ridge model.

2.3.2 Can a reduced set of variables predict the Behavior variable?

From the performance measured on the test set we can conclude that the model using ridge regularization can predict the `Behavior` variable with a performance of 83% and model with lasso regularization can predict `Behavior` variable with a performance of 96%. The notable observation was that both models have 0 count of false negative results.

3. Boosting model

3.1 Searching for optimal model

A Boosting tree - based model was used for the classification. Hyperparameters for number of trees = 5000 and interaction.depth = 5 were used in the initial model.

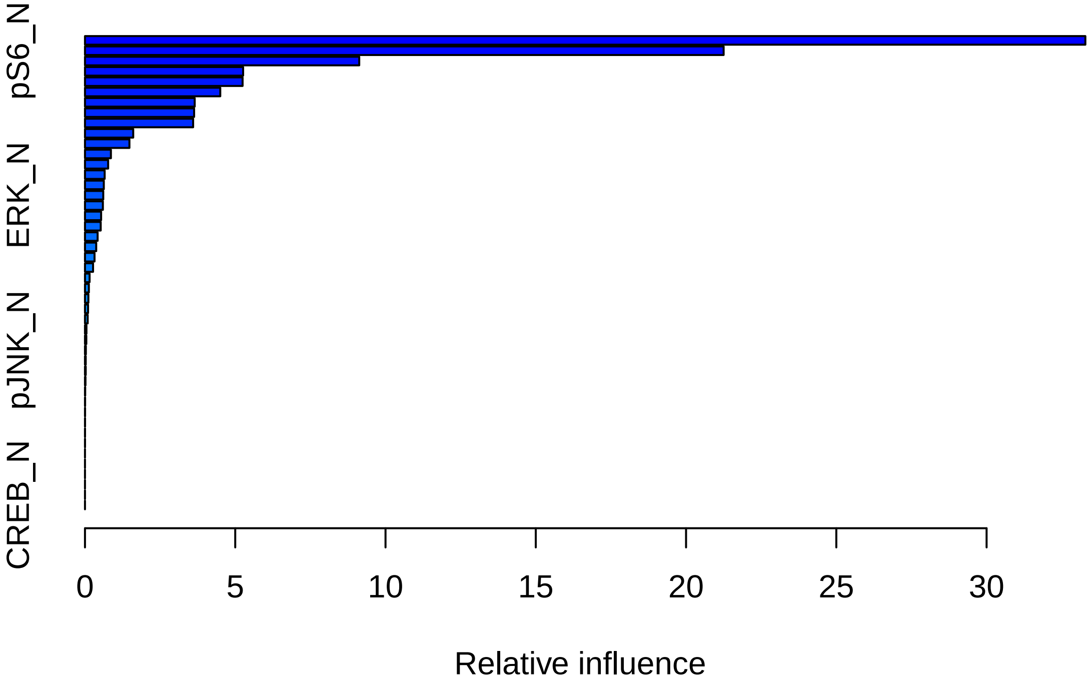
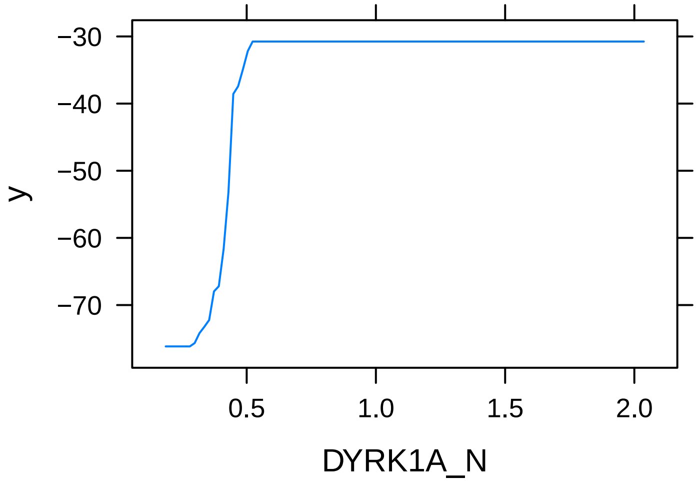


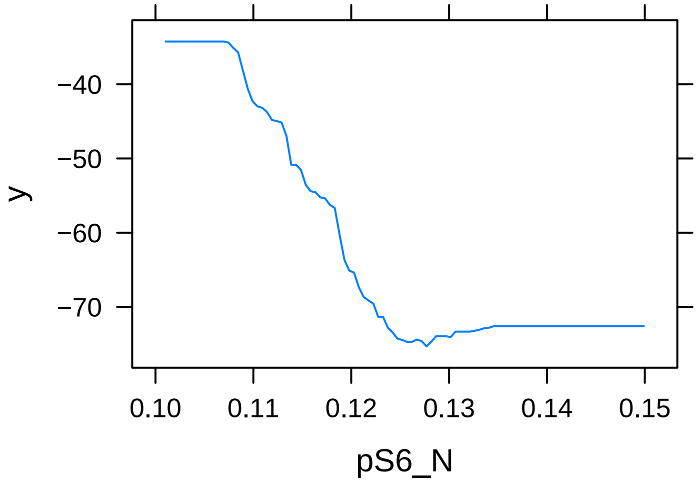
Figure 5: Relative influence plot

```
##                               var      rel.inf
## DYRK1A_N                 DYRK1A_N 3.328668e+01
## pS6_N                       pS6_N 2.125086e+01
## pPKCAB_N                 pPKCAB_N 9.125515e+00
## P38_N                       P38_N 5.262076e+00
## BRAF_N                     BRAF_N 5.244478e+00
## pCAMKII_N               pCAMKII_N 4.501554e+00
## pGSK3B_N                 pGSK3B_N 3.651040e+00
## pP70S6_N                 pP70S6_N 3.631353e+00
## pMEK_N                     pMEK_N 3.598859e+00
## pCREB_N                   pCREB_N 1.604710e+00
## AKT_N                       AKT_N 1.478945e+00
## pNUMB_N                   pNUMB_N 8.626719e-01
## TIAM1_N                 TIAM1_N 7.709266e-01
## GluR3_N                 GluR3_N 6.548936e-01
## nNOS_N                   nNOS_N 6.268206e-01
## ERK_N                     ERK_N 6.084895e-01
## PSD95_N                 PSD95_N 5.959918e-01
## RRP1_N                   RRP1_N 5.369568e-01
## CDK5_N                   CDK5_N 5.238459e-01
## GluR4_N                 GluR4_N 4.208383e-01
## SYP_N                     SYP_N 3.736765e-01
## pGSK3B_Tyr216_N pGSK3B_Tyr216_N 3.177303e-01
## GSK3B_N                 GSK3B_N 2.688539e-01
## P3525_N                 P3525_N 1.539358e-01
## pELK_N                   pELK_N 1.297542e-01
## ERBB4_N                 ERBB4_N 1.138437e-01
## pBRAF_N                 pBRAF_N 1.030662e-01
## pAKT_N                   pAKT_N 9.028600e-02
## pNR1_N                   pNR1_N 4.891044e-02
## APP_N                     APP_N 4.413855e-02
## pJNK_N                   pJNK_N 2.948385e-02
## RSK_N                     RSK_N 2.515556e-02
## AcetylH3K9_N       AcetylH3K9_N 2.459927e-02
## Tau_N                     Tau_N 1.682476e-02
## pMTOR_N                 pMTOR_N 7.322088e-03
## IL1B_N                   IL1B_N 4.537302e-03
## AMPKA_N                 AMPKA_N 3.563702e-03
## MEK_N                     MEK_N 2.609887e-03
## NR2B_N                   NR2B_N 2.130106e-03
## S6_N                       S6_N 1.619800e-03
## SHH_N                     SHH_N 3.528012e-04
## pPKCG_N                 pPKCG_N 6.180739e-05
## MTOR_N                   MTOR_N 4.378246e-05
## NUMB_N                   NUMB_N 1.580404e-06
## BAX_N                     BAX_N 3.646384e-07
## CREB_N                   CREB_N 6.408819e-09
```

The first two variables - `DYRK1A_N` and `pS6_N` seem to be of greatest importance. Interestingly, these variables were not assigned the largest weight in the lasso or ridge models, in fact `DYRK1A_N` was not even selected with a lasso model. When comparing the the top 11 predictors according to the relative importance in the Boositing model with the 11 variables selected by lasso there is an overlap of only 2 variables `pPKCAB_N` and `pCREB_N`.

Below we can see partial dependence plots that isolation the effect on individual variables. Plots for the two most important variables were constructed: we can see that for increasinig expression levels of protein `DYRK1A_N` the likelihood of `C/S` behavior is increasing, while the opposite is true for the expression levels of protein `pS6_N`.





3.1.1 What is the predictive power of the model?

The table below displays counts of:

- True negative results = 15
- True positive results = 9
- False negative results = 0
- False positive results = 0

The total performance of the model calculated as *(TN count + TP count)/total count* is 100%.

```
boost.pred = predict(model.boost, cortex[test, ], n.trees = 3000)
predictions=rep(0 ,length(cortex[test, "dummy"]))
predictions[boost.pred>0.5]= 1
table(cortex[test, "dummy"],predictions)
```

```
##    predictions
##      0  1
##   0 15  0
##   1  0  9
```

```
performanceBoost=length(which(predictions==cortex[test, "dummy"]))/length(cortex[test, "dummy"])
performanceBoost # 100%
```

```
## [1] 1
```

3.2 Comparison of Boosting model to Ridge and Lasso Models

3.2.2 Are the same variables important for the predictions?

As it was previously mentioned (*section 3.1*) the most relevant predictors from Boosting model and Lasso/Ridge differ significantly. The biological function of the most relevant protein selected by boosting model is: * DYRK1A = Dual specificity tyrosine-phosphorylation-regulated kinase 1, the literature suggests its potential link to Down syndrome

3.2.3 Do you see evidence for non-linear effects or interactions between the most important predictor variables?

Boosting tree-based model automatically takes into account the interactions between the variables. For example: once the decision tree is divided at node based on predictor A and the two nodes of A are further divided based on predictor B; the 'decision value' in B predictors can be different for the two branches created from A. Also the model is also not inherently linear. Since for this model we observed 100% test set performance we can conclude that there are likely non-linear effects and interactions that were explained in Boosting model and hence a better test set perfomance could have been achieved compared to the lasso/ridge models.

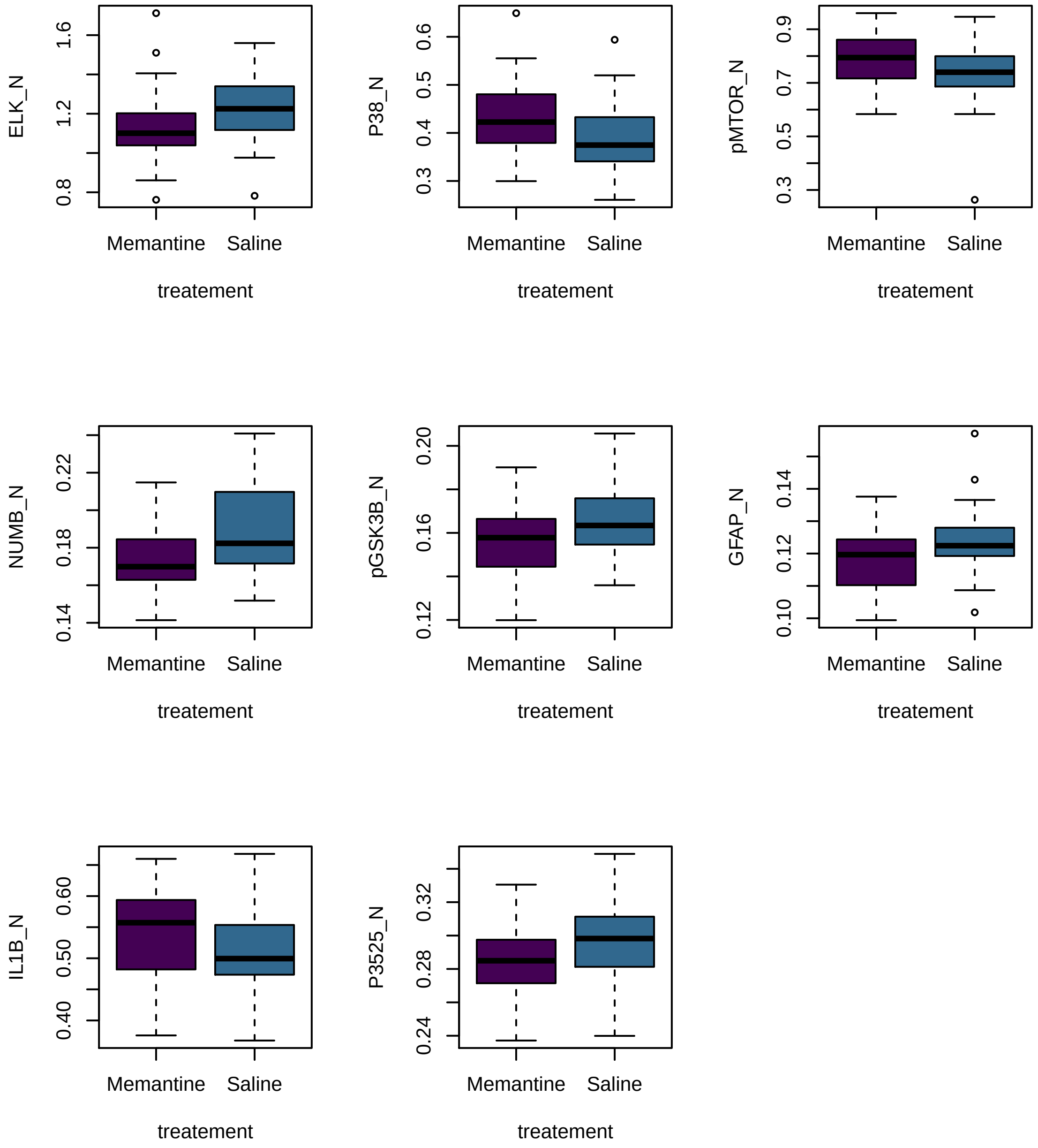
4. Does Memantine injection influence protein values when controlling for genotype and treatment?

4.1 Effect of Memanatine

To see if the Memantine treatment has influence of the protein expression values **Wilcoxon rank sum test** and was used. Since it was found that normality cannot be assumed a non-parametric alternative to t-test was used. For each protein the expression level values were split into two groups - treated and not-treated with Memantine. The following hypothesis was tested: *H0: The center values in both treated and untreated samples are identical.* If the p.value smaller that 0.05 was reached, H0 could be rejected in the favor of the alternative. In this way a list of differentially expressed proteins was found:

```
"ELK_N"    "P38_N"    "pMTOR_N"  "NUMB_N"   "pGSK3B_N" "GFAP_N"   "IL1B_N"   "P3525_N"
```

The following boxplots display the distributions for the differentially expressed in the two conditions observed.



4.2 Effect of Memnatine fixing `Behavior` and `Genotype`

To see if Memantime in itself has a significant effect on gene expression, the gene expression data we divided into groups based on their `Behaviour` and `Genotype` values. In this way the gene expression data were grouped into 4 groups:

- `Genotype = Ts65Dn Behavior = C/S`
- `Genotype = Ts65Dn Behavior = S/C`
- `Genotype = Control Behavior = C/S`
- `Genotype = Control Behavior = S/C`

This approach was attempting to minimize the effect of the two other variable on gene expression and isolate only the effect of `Treatment`. For analysis Wilcoxon rank sum test was used as in the previous analyses. With the combinations of both variables no significant result was found. Therefore the next step was to fix just one variable at a time and observe the effect of `Treatment`. However with this approach it is not possible to disregard the effect of the second variable. For this analysis the following four groups were created and the corresponding differentially expressed proteins were found:

- `Genotype Ts65Dn: "NR1_N"    "NR2A_N"   "pPKCAB_N" "JNK_N"    "APP_N"    "ADARB1_N"`
- `Genotype Control: "pNR2A_N"  "NR2B_N"   "RAPTOR_N" "ADARB1_N" "P3525_N"`
- `Behavior C/S: "S6_N"`
- `Behavior S/C: "DYRK1A_N" "ITSN1_N"   "pAKT_N"  "pCAMKII_N"   "PKCA_N" "pPKCAB_N" "BRAF_N" "GSK3B_N" "TRKA_N" "APP_N" "DSCR1_N" "pNUMB_N" "TIAM1_N" "pPKCG_N" "GluR3_N" "Ubiquitin_N"`