Identifying Proteins Critical to Learning Ability of Down syndrome Mouse Model

Priyansu , student at iitb

Table of Contents

Abstract

Down syndrome (DS) is a chromosomal disorder where organisms have an extra chromosome 21, sometimes known as trisomy 21. Being a syndrome, DS consists of multiple symptoms affecting a large number of systems in the body. It's effect on learning results in it being an intellectual disability **[1]**. Memantine, is currently proposed as a treatment of the learning deficit symptoms in DS. In this project, we have used several supervised machine learning methods: a decision tree classifier, a random forest classifier, and a third classifier which uses a chain of random forest classifiers., to identify which protein(s) are critical to mice learning ability after being exposed to context fear

conditioning (CFC). 77 protein expression levels are analysed from both control and trisomic (Ts65Dn) genotype mice, both with and without treatment from the drug memantine. Result suggest that decision tree and random forest classification approach can identify the most important proteins which may help to identify more effective drug to help learning process in people with DS.

Introduction

Human bodies are made up of millions of cells, and in each cell there are 23 chromosome pairs (46 in total). The DNA in the chromosomes determines how the body develop by encoding the sequence of amino acids in proteins. People with DS have 47 chromosomes in their cells instead of 46. This extra chromosome is known as trisomy of human chromosome 21 (*Hsa21*). It is believed the symptoms of trisomies like DS are the result of an over expression of proteins encoded on the extra chromosome. Trisomy of *Hsa21* is associated with a mild-to-moderate learning disability, craniofacial abnormalities and hypotonia in early infancy. Approximately 0.45% of human conceptions are trisomic for *Hsa21* **[2]**.

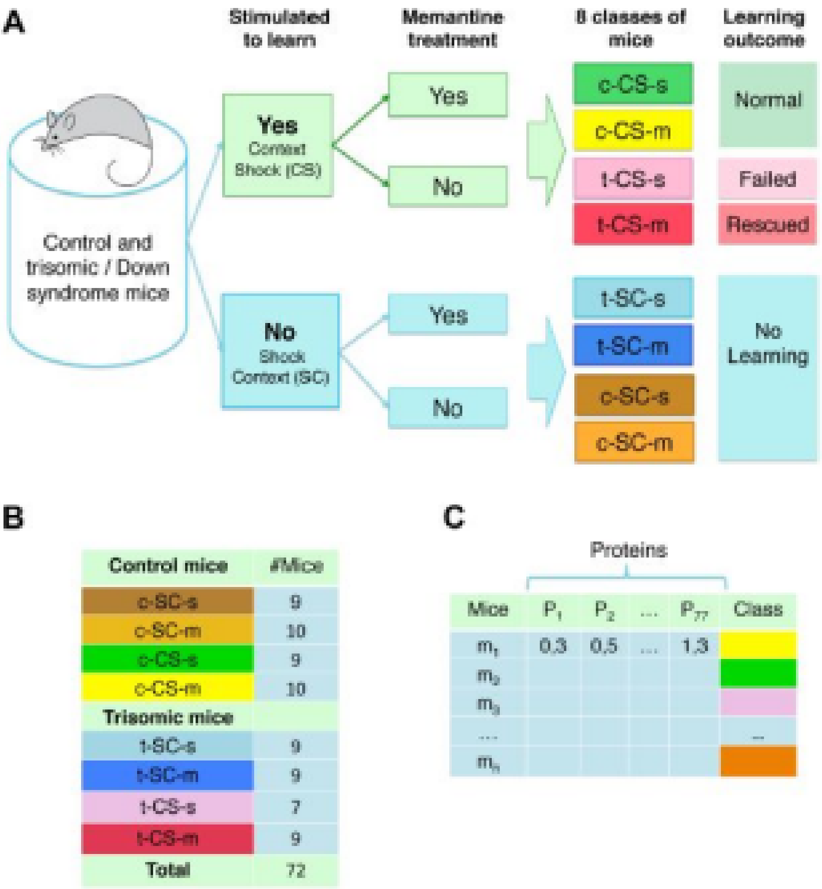
To understand DS, we need to understand the genomic trisomy protein content of *Hsa21* and to evaluate how the expression levels of these genes are altered by the presence of a third copy of *Hsa21*. For example, over-expression of number of *Hsa21* genes trisomy protein such as *DYRK1A* and *SIM2* may contribute to learning disability in people with DS. Also, trisomy protein of neuronal channel proteins such as *GIRK2* may also influence learning in people with DS.

In this project, 77 expression levels of proteins/protein modifications were examined to see which proteins contributed to successful and failed learning. These proteins produced detectable signals in the nuclear fraction of cortex, and in this research, it gathered from a total of 72 mice. There were 38 control/normal mice and 34 trisomic mice, both are trained in context fear conditioning (CFC).

CFC process required these mice to be separated into two groups, context-shock (CS) and shock-context (SC) groups. First, mice from CS group are place in a cage, allowed to explore the cage and then given brief electric shock. Control/normal mice should freeze and learn the association of their cage and the shock, while trisomic mice is supposed to fail to learn. Second, mice in SC group are placed in a cage and given the electric shock immediately, therefore both of normal and trisomic mice will fail to learn the association between cage and the shock.

To assist trisomic mice in learning, memantine is injected prior to training. To control the effect of this injection, half of the mice in CS also SC group is injected with memantine, while the other group is injected with saline (no-drugs).

15 measurements of each proteins are being conducted. Therefore, there were 15x38 (570 measurements) for control/normal mice, and 15x34 (510 measurements) of trisomic mice. In total, this dataset has 1080 measurements from 72 mice in each expression level of proteins. The relationship of these data is listed in figure below:



Control mice	#Mice
c-SC-s	9
c-SC-m	10
c-CS-s	9
c-CS-m	10
Trisomic mice	
t-SC-s	9
t-SC-m	9
t-CS-s	7
t-CS-m	9
Total	72

Mice	P_1	P_2	...	P_{77}	Class
m_1	0.3	0.5	...	1.3	🟨
m_2					🟩
m_3					🟪
...					...
m_n					🟧

Figure 1 [4] – Classes of mice. **(A)** 8 classes of mice based on Genotype (control, c, and trisomy, t), stimulation (Content-Shock, CS, Shock-Content, SC), and treatment (saline, s, memantine, m). **(B)** The number of mice in each class. **(C)** 77 proteins expression data

The goal of this project is to understand which trisomy protein classes that contribute to the success and the failure of mice learning. This is done by creating a model which predicts which of the 8 classes of mice some mouse was in based on their protein expression levels. By creating a successful model, we can then determine which proteins were significant in the predictions, which would support a hypothesis that that particular protein affects learning in trisomic mice.

Methodology

Data analysis of this project consisted of 4 steps: 1) data pre-processing, 2) data exploration, 3) data modelling, 4) test the model.

1. Data pre-processing.

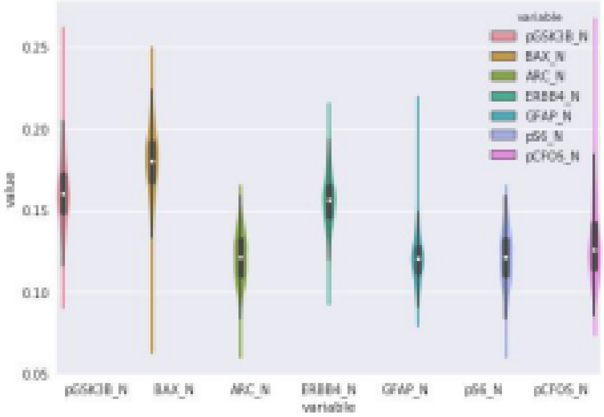
This dataset in each protein contain missing values. To deal with these, the mice will receive the mean value of the mice in their class, rather than eliminating the entire mouse from the data, or by using the mean of all of the mice which have very different genetic and environmental factors in their development.

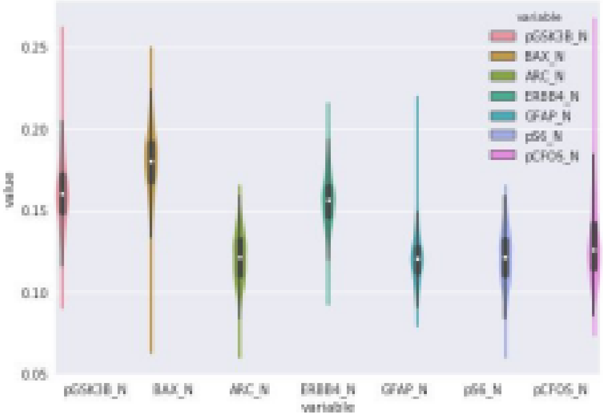
The second thing that needs to be dealt with is the high variance of measurement values. For example, some proteins have exploration value range from 0 to 1 while other proteins ranged from 0 to 8. With this condition, proteins with higher value will have more influence in this classification outcome and make the model unable to learn correctly. Therefore, after making sure no null values in the measurements, all value of measurements are standardized with zero means and unit variance.

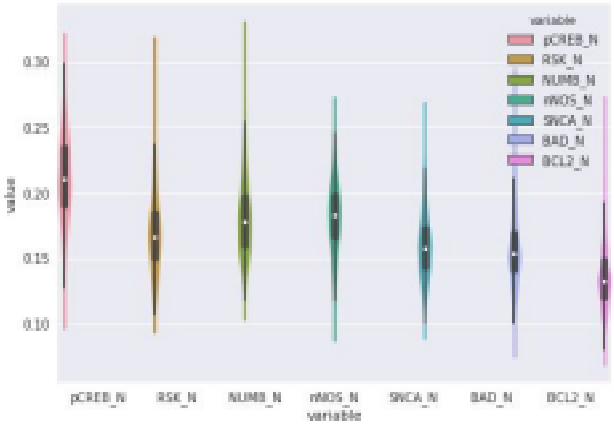
2. Data Exploration

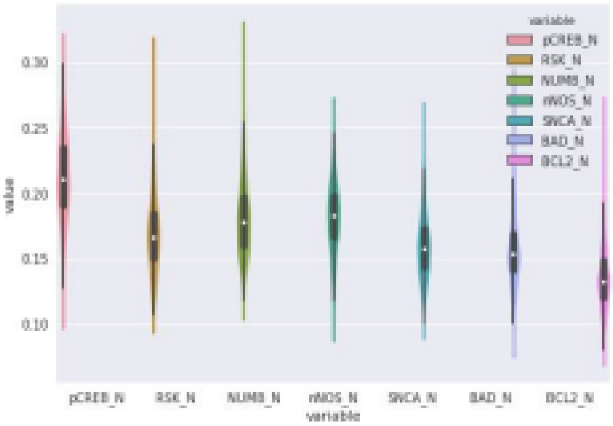
The data exploration was done on the data following the replacement of nulls with the means, but before it was normalised.

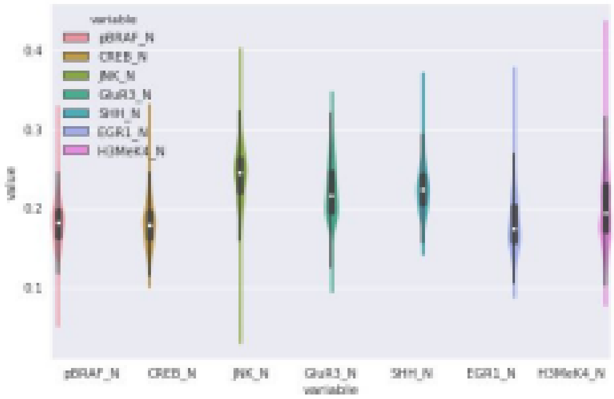
We made series of violin plots of the expression value for each protein type as our first step of data exploration. Violin plots are useful to see full distribution of the data with their probability density. To make this violin plot series, the mean value of each protein is first calculated and the data frame is sorted based on their mean magnitudes. This helped to organise the data for easier visual inspection.

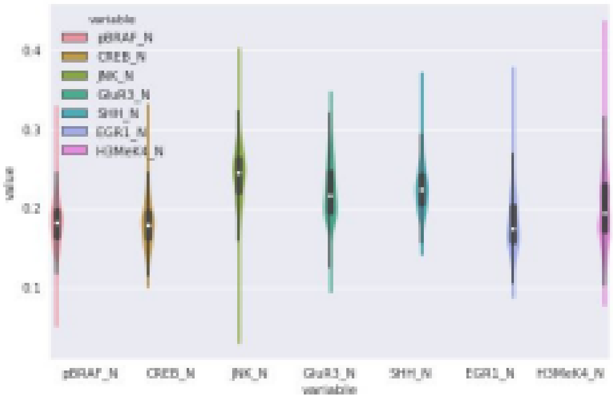


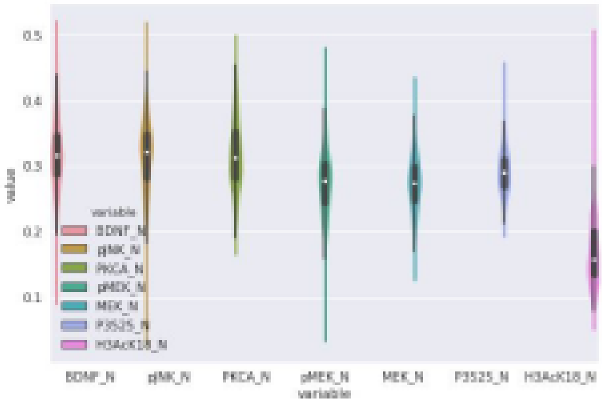


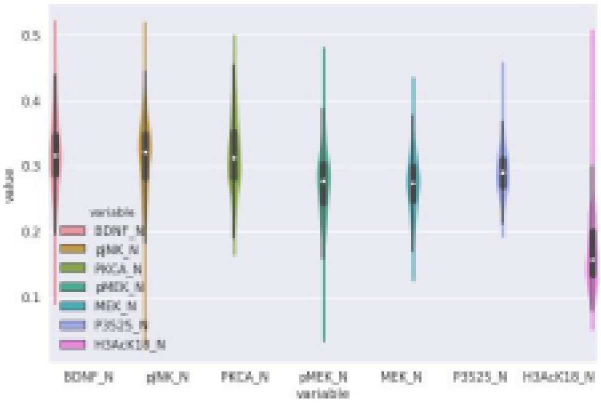


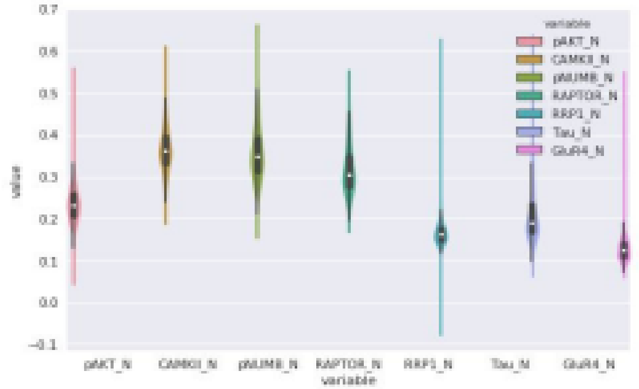


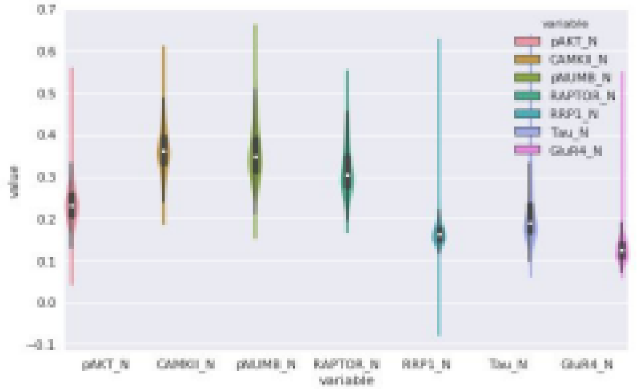


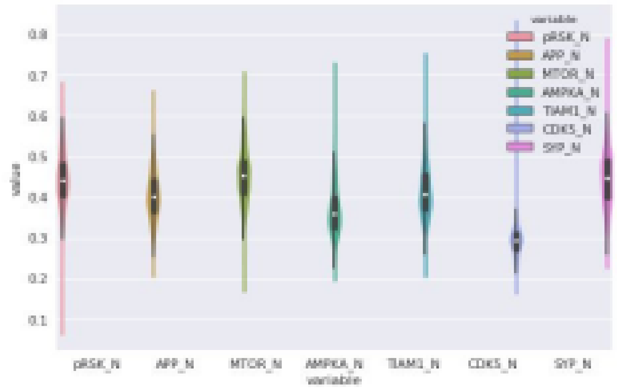


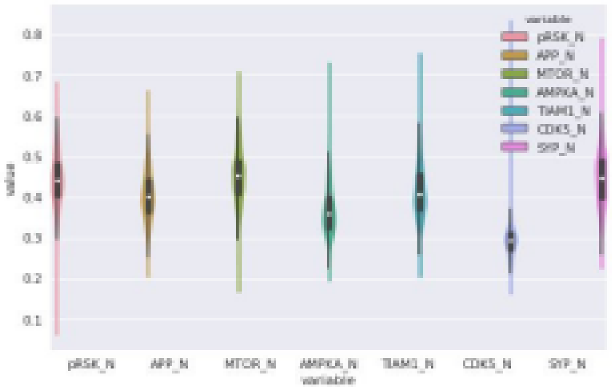


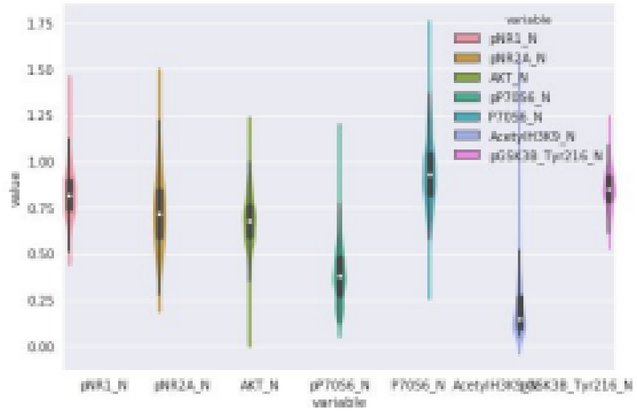



value
variable
pAKT_N
CAMKII_N
pNUMB_N
RAPTOR_N
RRP1_N
Tau_N
GluR4_N
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-0.1


value
variable
pAKT_N
CAMKII_N
pNUMB_N
RAPTOR_N
RRP1_N
Tau_N
GluR4_N
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-0.1


value
variable
pRSK_N
APP_N
MTOR_N
AMPKA_N
TIAM1_N
CDK5_N
SYP_N
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1


value
variable
pRSK_N
APP_N
MTOR_N
AMPKA_N
TIAM1_N
CDK5_N
SYP_N
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1


value
variable
pNR1_N
pNR2A_N
AKT_N
pP70S6_N
P70S6_N
AcetylH3K9_N
pGSK3B_Tyr216_N
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00

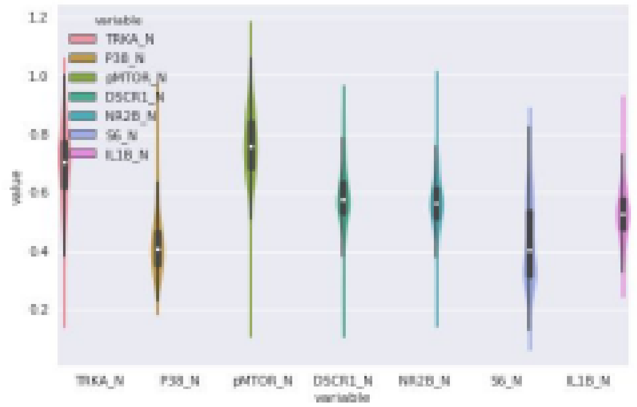

value
variable
TRKA_N
P38_N
pMTOR_N
DSCR1_N
NR2B_N
S6_N
IL1B_N
1.2
1.0
0.8
0.6
0.4
0.2

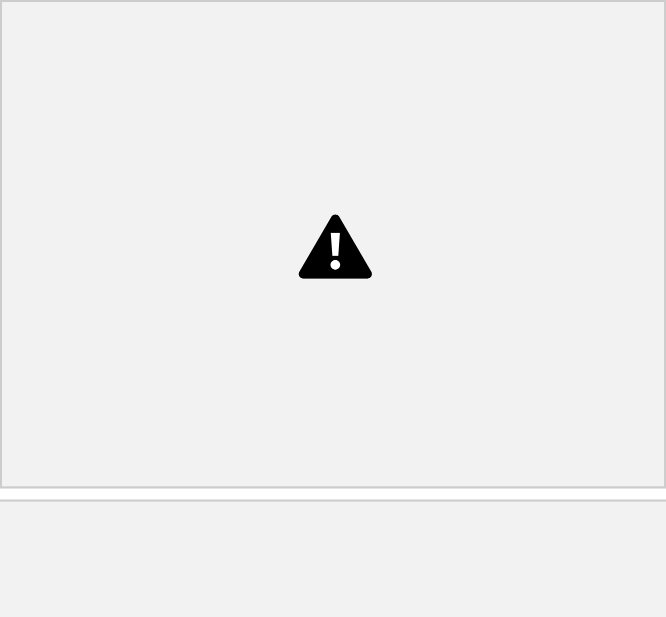
Figure 2 – Exploration of proteins based on their magnitudes

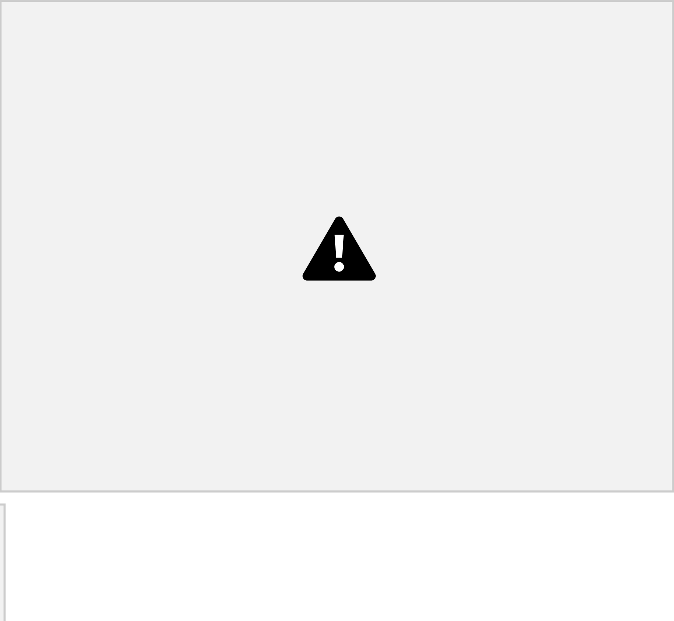
These violin plots showed how varied the magnitudes of each protein was, and shows why data standardization is needed. Also, the plots show most of the data have large ranges of outliers. To

understand where the outliers come from, we will do inspection of selected single protein by using pair wise exploration with their class.

The 5 proteins with the most significant range of outliers will be chosen and examined:DYRK1A_N, AcetylH3K9_N, RRP1_N, NR2A_N, and pP70S6.

Besides the big range of outliers, these protein are also chosen because 1 of them have negative values (protein RRP1_N) and 1 of them have the most vary range of magnitudes (NR2A_N). The result below:





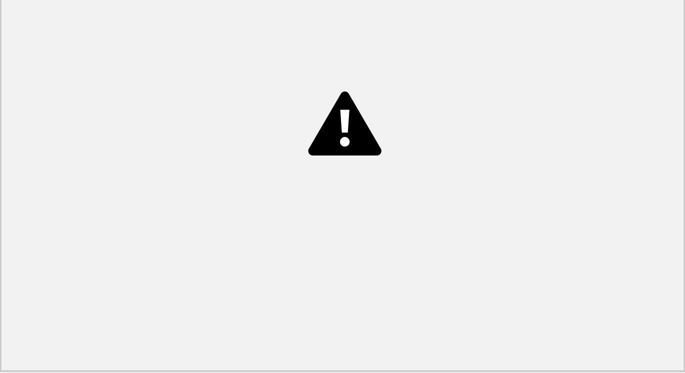
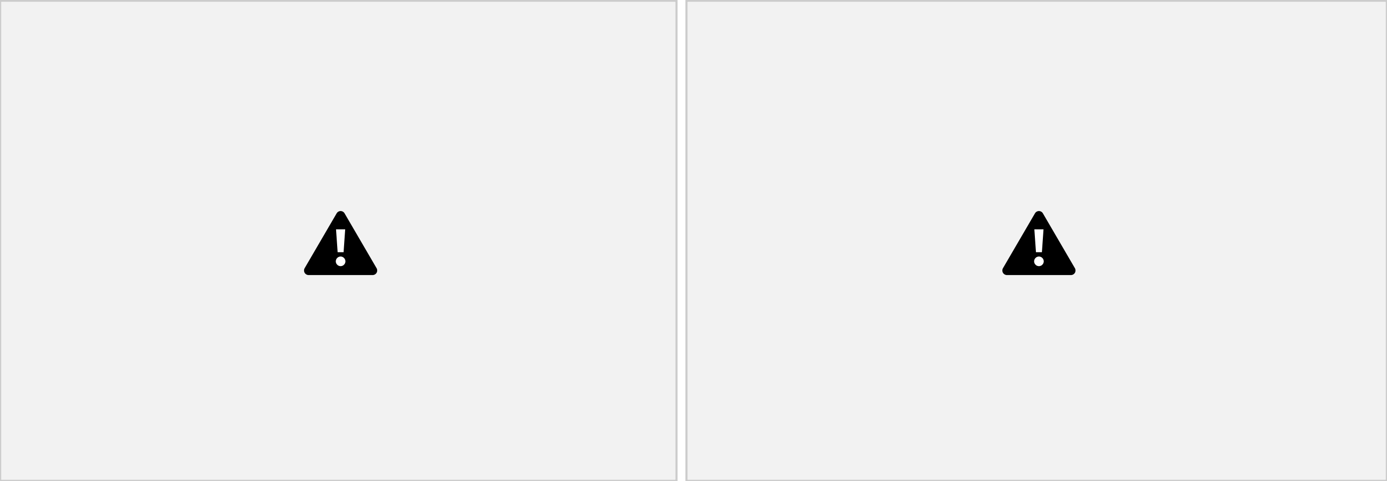


Figure 3 – Exploration of proteins based on mice class

These pairwise explorations show most of the big range of data magnitudes caused by outliers that usually dominated by one of the class. For example, big range of data magnitudes in protein DYRK1A_N is caused by outliers from c-CS-s class. Also, in protein pP70S6 its caused by outliers in t-CS-m class.

Now we will go deeper into each class that contains those outliers by examining the mouse numbers in those class. The result is below:



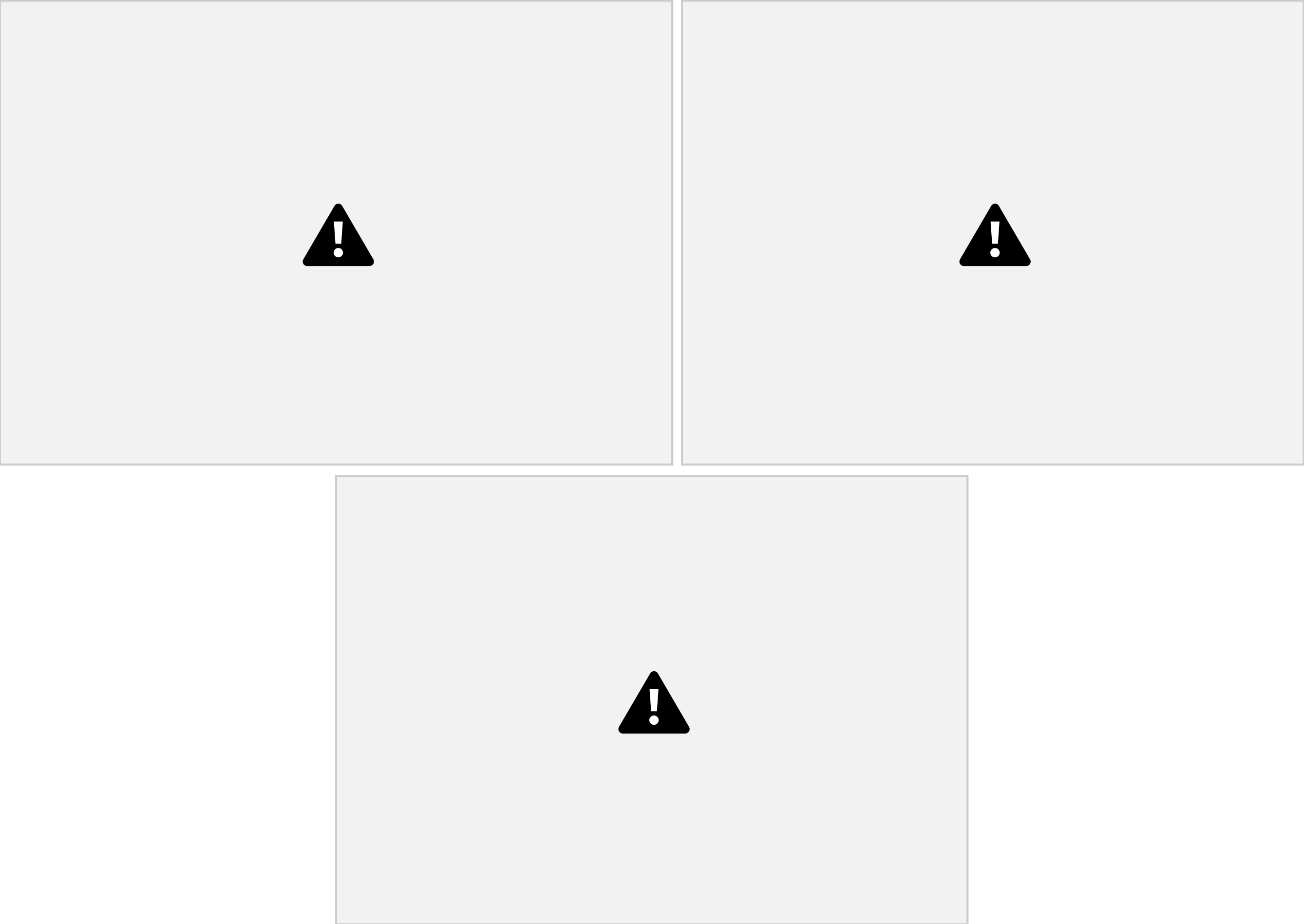


Figure 4 – Exploration of proteins based on mouse numbers in class contains dominating outliers.

In deeper analysis, we can see that most of the dominating outliers in each class comes from specific mouse numbers. For example, the obvious outliers from c-CS-s class in protein DYRK1A_N is comes from mouse number 3484. Also, the big magnitudes in protein NR2A_N in c-CS-s class is comes from mouse number 3497.

Now we are examining the mouse version from these 'anomaly' mouse numbers to get more understanding about these outliers that makes the data magnitudes vary. The result is below:

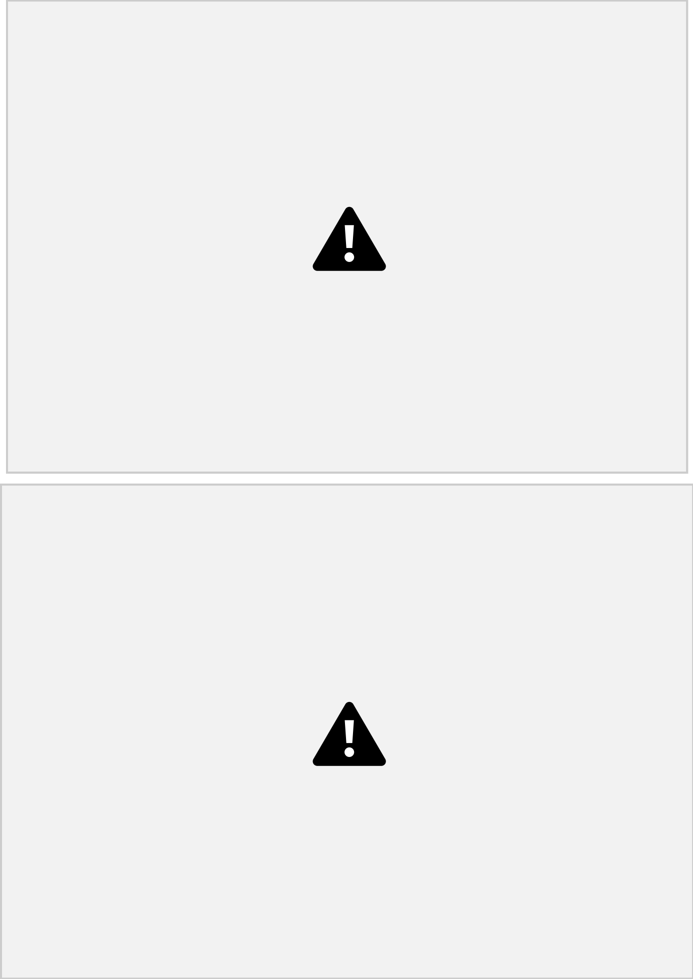


Figure 5 – Exploration of proteins based on mouse numbers in class contains dominating outliers.

Different from previous result, now most of the obvious outliers is not coming from single mouse version, but from multiple mouse versions.

Therefore, the conclusion of this exploration is most of the magnitude data in proteins is vary because the outliers. These outliers usually comes from one specific mouse class and one specific mouse number in those class. However, from those specific mouse number comes multiple mouse versions that produced these outliers. Standardization of these magnitudes is clearly needed for the next step, data modelling.

3. Data Modelling

Model Background

Decision tree and random forest were used to model and predict the mice class and predict which proteins were critical for each class. These are both examples of supervised machine learning methods with the goal of creating model that predicts the value of a target variable based on several input variables.

Decision tree algorithm tries to solve the problem by using tree representation of a series of decisions. A tree can be "learned" by splitting the source set into subsets based on an attribute value test. The splitting creates two types of nodes: decision nodes, and leaves. The starting node is called the root. If the root is a leaf then the decision tree is degenerate and the same classification is made for all data. A single variable is being examined for each decision nodes, and move to another node based on the outcome of a comparison. This is repeated until a leaf node is reached. At a leaf node the decision is being made: whether the training data routed to the leaf node as a classification decision, or return the mean-value of outcomes as a regression estimate. **[5]**

Similar with decision tree, random forest also use tree representation to solve the problem. The difference is instead of using one tree, random forest averaging multiple deep decision trees and trained on different parts of the same training set, with the goal of reducing the variance and overfitting. **[6]**

Model Implementation

Following the data pre-processing step, the raw_df dataframe was broken up into two data frames, with only the data kept.

Figure 6 – The dataframe was broken up into two dataframes.

The sklearn module was used to create these models. Due to the consistent API with which sklearn creates classifiers, a wrapper function `run_classifier` was created that:

1. Takes as input an instantiated classifier
2. Chose a random set of samples from the data to be the training data (two thirds of the data set was used for training)
3. Set the test data to the complement of the training data
4. Fitted the training data
5. Ran the model on the test data
6. Created the performance report

The code snippet below provides more detail.

Figure 7 – Wrapped up “run_classifier” function for splitting the training and test data and create performance report.

In the run_classifier function above, you can see the call to the performance report function. This function was written so that we can judge the effectiveness of the classification based on how well it can predict y_test using the x_test as input using the following performance metrics/methods:

- **Confusion Matrix**
- **Classification error rate**: the percentage of observations in the test data that the model mislabelled
- **Precision**: The precision is the ratio tp/(tp+fp) where *tp* is the number of true positives and *fp* is the number of false positives i.e the ability of the classifier not to label as positive a sample that is negative.
- **Recall**: The recall is the ratio *tp / (tp + fn)* where *tp* is the number of true positives and *fn* the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples.
- **F1-score**: the F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is: F1 = 2 (precision recall) / (precision + recall)

The function (snippet below) allows easy repeatability of the performance report across a range of classifiers which follow the sklearn api.

Results

In this project, we built and used 3 models. First, decision tree model with “class” variable as a compound of three binary variables (8 in total). Second, random forest model with also “class” variable. Third, we created our own classifier that first predicted the 3 binary variables (“Genotype”, “Treatment”, and “Behavior”), before passing these variables in as binary features to another Random Forest classifier.

Performance

Decision Tree

The decision tree model analysis with “class” variable result is presented below:

Class	preci sio n		recall f s	
c-CS-m	0.83	0.85	0.84	53
c-CS-s	0.79	0.86	0.82	43
c-SC-m	0.84	0.78	0.81	46
c-SC-s	0.89	0.89	0.89	45
t-CS-m	0.85	0.73	0.79	45
t-CS-s	0.78	0.8	0.79	35
t-SC-m	0.84	0.88	0.86	42
t-SC-s	0.9	0.92	0.91	51
avg / total	0.84	0.84	0.84	360

Figure 8 – Performance matrix of decision tree classifiers with “class” variable as predictor

Random Forest

Figure 9 – Confusion matrix of random forest classifiers with “class” variable as predictor.

Class	preci sio n		recall f s	
c-CS-m	1	1	1	39
c-CS-s	1	0.98	0.99	47
c-SC-m	1	1	1	54
c-SC-s	1	0.98	0.99	48
t-CS-m	1	1	1	47
t-CS-s	0.97	1	0.99	37
t-SC-m	0.98	0.96	0.97	48
t-SC-s	0.95	1	0.98	40
avg / total	0.99	0.99	0.99	360

Figure 10 – Performance matrix of random forest classifiers with “class” variable as predictor.

Compound Binary Classifier

The class variable (with 8 classes) is a compound of three binary variables (“Genotype”, “Treatment”, and “Behaviour” variables). It then makes sense to try and take advantage of this and instead predict individual binary variables separately. It is not enough to just predict them separately, because there is significant interplay between the genotype, treatment and behaviour on protein expression levels, and so the predicted binary variables are then included in another model where they are used as features along with the expression levels.

Separately, the binary features are all well predicted:

Genotype

Class	**preci sio n**		**recall f s**	
Control	0.99	0.99	0.99	179
Ts65Dn	0.99	0.99	0.99	181
avg / total	0.99	0.99	0.99	360

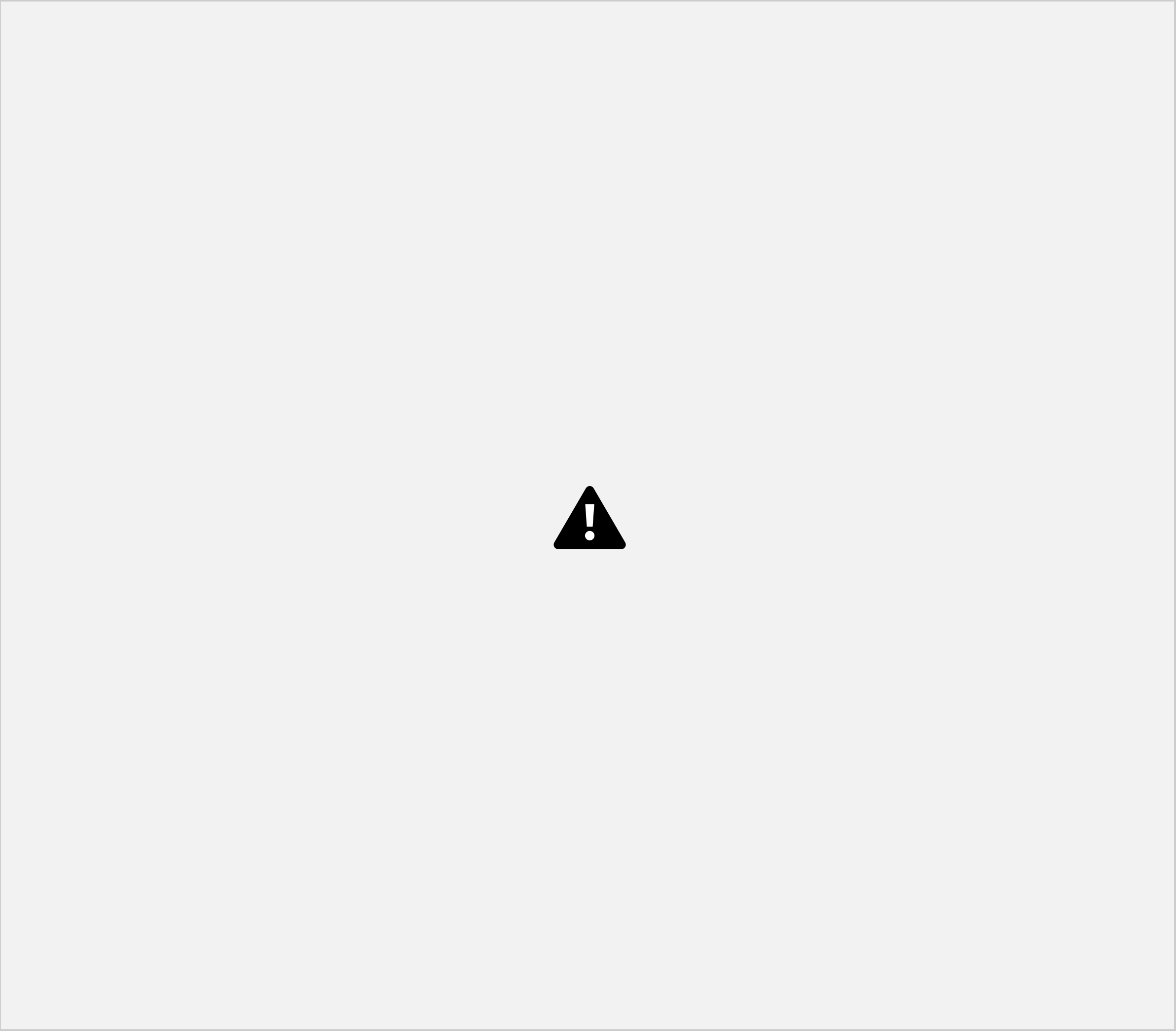


Figure 11 – Performance and confusion matrix of random forest classifiers with “Genotype” variable as predictor.

Behavior

Class	preci sio n		recall f s	
C/S	1	1	1	176
S/C	1	1	1	184
avg / total	1	1	1	360

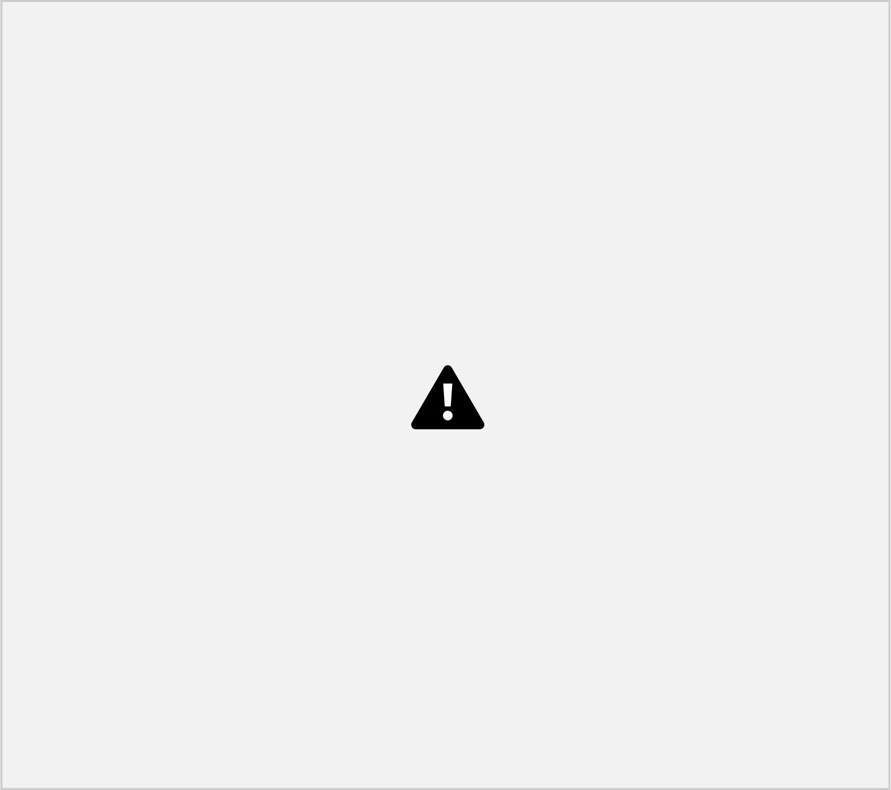


Figure 12 – Performance and confussion matrix of random forest classifiers with “Behavior” variable as predictor.

Treatment

Class	**precisio n**		**recall f s**	
Memantine	1	0.98	0.99	194
Saline	0.98	1	0.99	166
avg / total	0.99	0.99	0.99	360

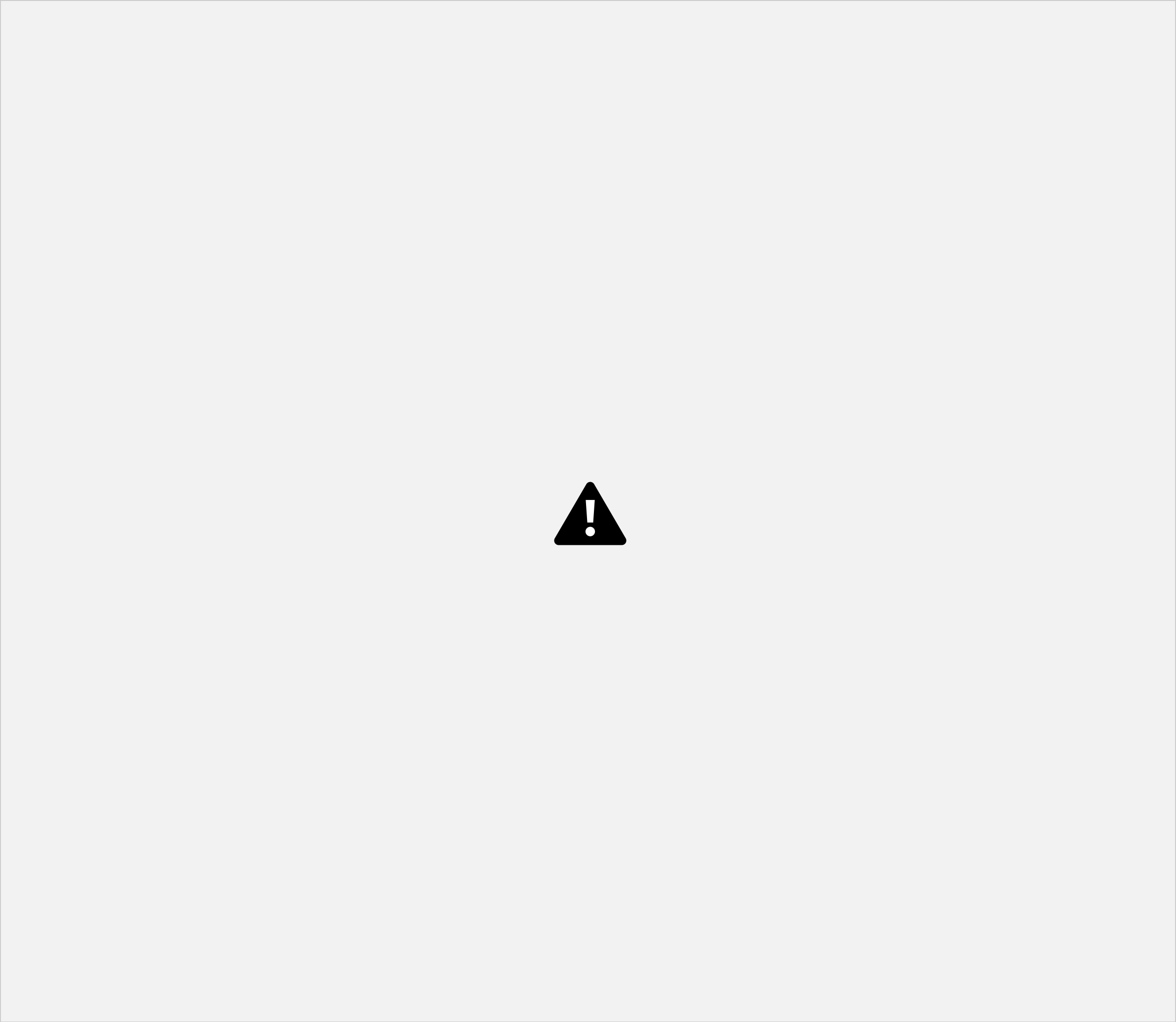


Figure 13 – Performance and confusion matrix of random forest classifiers with “Treatment” variable as predictor.

Class

Class	**preci sio n**		**recall f s**	
c-CS-m	1	0.95	0.97	59
c-CS-s	0.94	1	0.97	45
c-SC-m	1	1	1	55
c-SC-s	1	1	1	43
t-CS-m	1	1	1	31
t-CS-s	1	1	1	35
t-SC-m	1	1	1	44
t-SC-s	1	1	1	48
avg / total	0.99	0.99	0.99	360

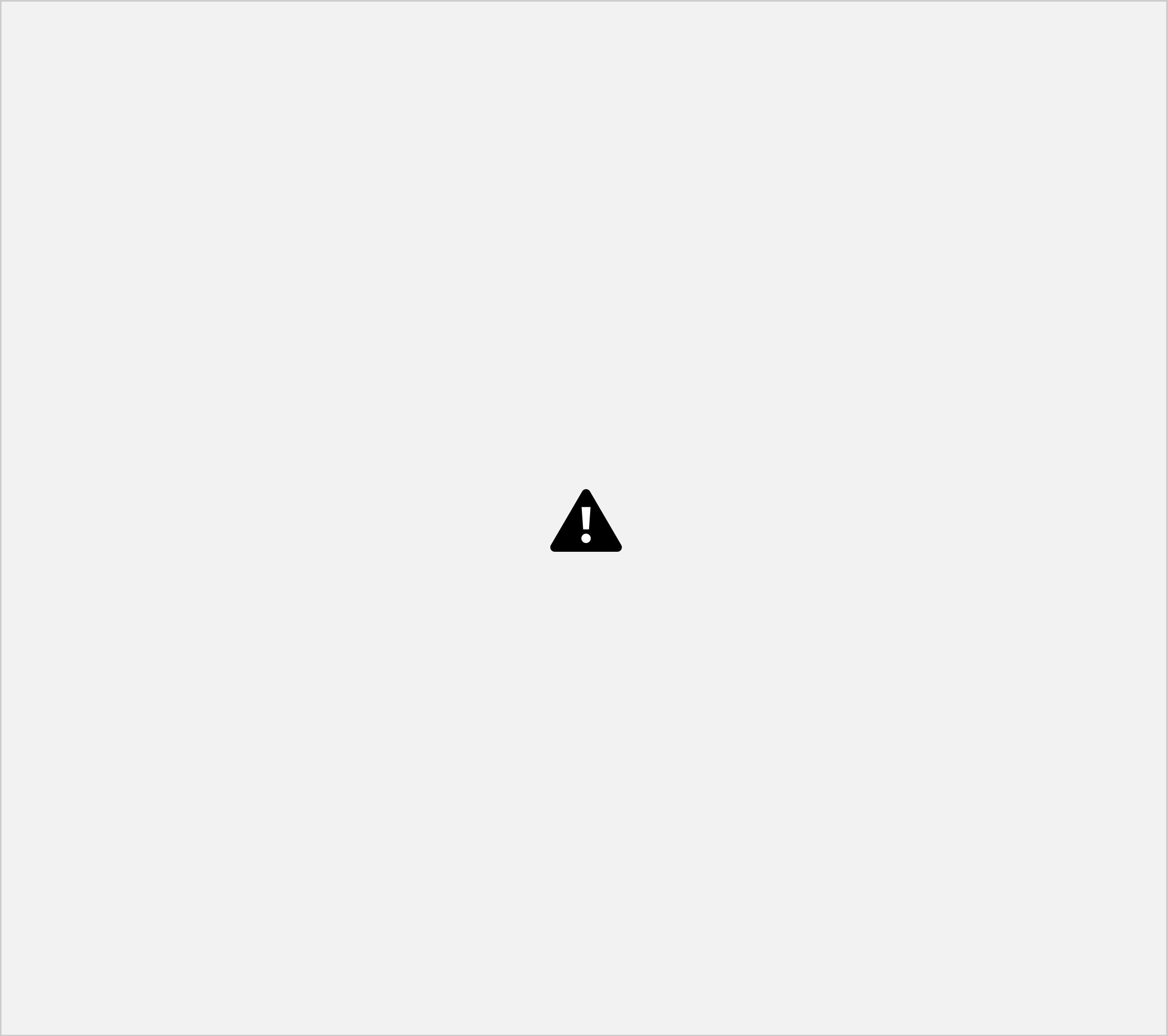


Figure 14 – Performance and confusion matrix of random forest classifiers with "Treatment" variable as predictor.

Feature Significance

A key feature of the RandomForestClassifier and the Decision Tree classifier is the ability to extract feature importance from the models. This is highly relevant to our aim, as it allows us to infer which proteins are more crucial in the metabolic pathways related to learning. The results are summarised below:

Feature	Random Forest Rank	Random Forest Importance	Decision Tree Rank	Decision Tree Importance
SOD1_N	1	0.056	1	0.130
pERK_N	2	0.039	56	0.000
pPKCG_N	3	0.034	2	0.114
APP_N	4	0.031	13	0.023
ITSN1_N	5	0.031	4	0.070

DYRK1A_N	6	0.026	23	0.013	
CaNA_N	7	0.026	16	0.017	
pS6_N	8	0.026	76	0.000	
pCAMKII_N	9	0.025	3	0.081	
BRAF_N	10	0.024	18	0.015	
ARC_N	11	0.024	12	0.024	
Ubiquitin_N	12	0.021	8	0.035	
pP70S6_N	13	0.021	66	0.000	
S6_N	14	0.018	28	0.007	
Tau_N	15	0.018	44	0.003	
pPKCAB_N	16	0.018	14	0.020	
pNUMB_N	17	0.017	32	0.006	
P38_N	18	0.017	63	0.000	
pMTOR_N	19	0.016	40	0.004	
AcetylH3K9_N	20	0.016	41	0.003	
pGSK3B_N	21	0.016	67	0.000	
AKT_N	22	0.015	20	0.014	
BCL2_N	23	0.015	75	0.000	
H3AcK18_N	24	0.014	5	0.045	
MTOR_N	25	0.013	7	0.037	
ADARB1_N	26	0.013	34	0.006	
NR2B_N	27	0.013	39	0.004	
pNR2A_N	28	0.012	33	0.006	
PKCA_N	29	0.012	10	0.032	
IL1B_N	30	0.011	11	0.029	

Table 1: A comparison of feature importance between the Random Forest and Decision Tree models

Feature	Genotype Rank	Genotype Importanc e	Behavio r Rank	Behavio r Tree	Treatmen t Rank	Treatmen t Tree	Random Forest Rank	Random Forest Importance
APP_N	1	0.113266	13	0.016 4 6 7		3 020306	13	0.021311

Tau_N	2	0.056407	44	0.0022 6		4 009762	44	0.008629
				8				
ITSN1_N	3	0.043478	4	0.0647 4	4	0.02676 9	4	0.031143
AcetylH 3K 9_N	4	0.039605	41	0.002 5 2 9		1 010644	41	0.008853
H3MeK 4_ N	5	0.028513	77	0.000 4 2 9		7 005661	77	0.00478
H3AcK1 8_ N	6	0.026362	5	0.061 8 9 2		5 0.026074	5	0.030514
S6_N	7	0.02563	28	0.005 0 6 3		8 012117	28	0.012113
DYRK1 A_ N	8	0.024302	23	0.005 6 4 4		3 013871	23	0.014642
pMTOR_ N	9	0.021581	40	0.002 6 7 9		0 010669	40	0.008915
NR2B_N	10	0.018981	39	0.002 6 8 6		9 010979	39	0.009139
pPKCG_ N	11	0.018683		2 0.13618		2 0.034435	2	0.039076
MTOR_N	12	0.018367	7	0.039 6 7 3		7 0.023722	7	0.025962
P38_N	13	0.018066	63	0.001 0 1		3 007016	63	0.006943
AMPKA_ N	14	0.016513	64	0.000 9 6		4 006921	64	0.006941

				3				
pCAMKII_ N	15	0.015503	3	0.091 1 4	3	0.02693	3	0.033574
GSK3B_ N	16	0.014441	60	0.001 0 5 6		) 007344	60	0.007173
TIAM1_N	17	0.01327	65	0.000 9 3 3		5 006894	65	0.006732
GluR3_N	18	0.013214	70	0.000 8 4 5		) 006519	70	0.006302
ARC_N	19	0.012858	12	0.024 2 4 2		2 020791	12	0.021383
SYP_N	20	0.012563	51	0.001 1 2		1 008536	51	0.008121
SOD1_N	21	0.012438	1	0.176 4 4 7		1).059713	1	0.055669
NR2A_N	22	0.012419		3 00152		3 008024	53	0.008055
RAPTOR_ N	23	0.012328	42	0.002 4 5 8	42	0.0104	42	0.008663
pP70S6_ N	24	0.012224	66	0.000 9 1 8		6 006882	66	0.006587
BRAF_N	25	0.011992	18	0.009 2 7 3		3 015949	18	0.017191
pNR1_N	26	0.01137	9	0.033 4 4 7		9).021907	9	0.024585

ADARB 1_ N	27	0.011365	34	0.003 9 1 2		4 011332	34	0.010255
ERK_N	28	0.011346	27	0.005 2 2 7		7 012421	27	0.012566
NR1_N	29	0.0107	43	0.002 2 8 3		3 010175	43	0.00866
Ubiquiti n_ N	30	0.010596	8	0.034 9 0 6		3).022909	8	0.02577

Table 2: A comparison of feature importance between the each binary predictor variables

Discussion

Overall, we can see that the classification models we've used are useful in trying to infer which proteins are significant contributors to the learning of mice in Cognitive Stress.

As expected, the Random Forest performed much better than the decision tree, which is a much less sophisticated model. The Random Forest had an almost perfect Recall, F1-Score and Precision scores of 0.99 for each, compared with the 0.84 for the decision tree. This was done by using exactly the same training and test data set by setting the seed of the random number generator for data selection. This does not affect the randomness within the Random Forest model.

The compound classifier was very interesting. It perfectly predicted 6/8 classes, but struggled on c-CS-m/s classes (mice with a normal genotype, from the context-shock group). The results from this classifier should therefore be treated with some level of scepticism as we're not sure why they behaved in this way. However given the strong scores in other categories, there would be merit in continuing to improve this method.

Aside from the scores obtained, it's worth looking more closely at the relative feature importance for each the Random Forest and the Decision Tree models in Table 1.

The top 30 features in the Random Forest model were used in the comparison. The Relative Importance fields is the relative contribution of this feature to the model. All features importance's sum to 1. If a features has a relative importance of 0.00, then it made no contribution. If it is equal to 1.00, then the model used this feature exclusively. There are two key results in this table worth noting:

1. SOD1_N, pPKCG_N, and ITSN1_N were each in the top five of a model
2. The Relative Importance of features in Random Forests drop off a lot more quickly than in the Decision Tree model, such that the set of features making a large contribution is quite small. The significance of this is that the Decision Tree model could be overfitted to the data, Random Forests are well known to be good at overfitting, and certainly seem to be avoiding this. 50% of the model decisions are made by the top 7 features in decision trees compared with the top 20 in the Random Forests, and 90% are made by the top 27 in decision trees, compared with 60 in the decision trees.

We can also look at the relative importances in Table 2. What's interesting here is that the Behavior and Treatment classes have an equal ranking of features (but with significantly different weightings), which is quite different to the Genotype expression levels. This is consistent with the genotype determining base levels of expression, with behaviour and treatment perturbing these base levels. The top 4 proteins for Treatment and Behaviour were SOD1_N, pPKCG_N, pCAMKII_N and ITSN1_N, the same as for the Random Forest model that predicted class. What this suggests, is that the protein expression levels of these proteins are significantly affected by treatment and behaviour.

Conclusion

In this project, random forest model behaves better than decision tree to predict which proteins are important for learning process of mice. Decision tree and random forest model shown the similar most crucial proteins in the metabolic pathways related to learning. Also, the feature importance comparison table showed us that protein that crucial in influencing mice behavior and mice treatment were similar.

References

1. http://www.downsyndrome.org.au/what_is_down_syndrome.html

2. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2657943/

3. Arron J.R., Winslow M.M., Polleri A., Chang C.P., Wu H., Gao X., Neilson J.R., Chen L., Heit J.J., Kim S.K., et al. NFAT dysregulation by increased dosage of DSCR1 and DYRK1A on chromosome 21. Nature. 2006;441:595–600

4. Higuera C, Gardiner KJ, Cios KJ (2015) Self-Organizing Feature Maps Identify Proteins Critical to Learning in a Mouse Model of Down Syndrome. PLoS ONE 10(6): e0129126. 5. https://www.r-bloggers.com/why-do-decision-trees-work/

6. https://en.wikipedia.org/wiki/Random_forest

7. http://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html