Comparing Unsupervised Classification Methods to Identify Biologically Important Proteins for Learning Behaviours in Mice

Gauri R. Kapse

Introduction:

In humans, Down Syndrome is caused by a duplication in chromosome 21. In mice, it is caused by the same duplications in mouse chromosomes 10, 16 and 17. As a result of the extra copies of the genes in these regions, people and mice affected with Down Syndrome show abnormal protein expression levels – in many cases, these deviations emerge as learning disabilities.

Since Down Syndrome causes learning disabilities in about 1 in 1000 people worldwide, developing pharmacotherapies to combat the learning difficulties posed by this disease is of great interest to the medical community. However, due to the affected portion of the genome being poorly studied and quite large, identifying biochemical targets for pharmaceuticals has been difficult. In the original paper that my project is based off of, a mouse model for Down Syndrome is examined.

The authors of the 2015 paper, "Self-Organizing Feature Maps Identify Proteins Critical to Learning in a Mouse Model of Down Syndrome," proposed that as an alternative to studying each gene in the affected genomic regions individually, we could examine the overall sub-cellular biochemical pathways involved in learning behaviours, and determine which portions of these pathways differ between healthy and affected mice. Then, we could design drugs to correct these perturbations.

The motivation behind the original paper was to use computational methods on protein expression data to do two things:

- *First*: determine which of 77 candidate proteins are critical for learning behaviours in mice affected by Down Syndrome vs wild-type (healthy/control) mice. These proteins had previously been demonstrated to be expressed during learning behaviours in wild-type mice, and differences in their expression levels between wild-type and affected mice could reveal potential drug targets.
- *Second*: explore the effects of the drug memantine on protein levels in a mouse model for Down Syndrome. Memantine is commonly used to treat Alzheimer's disease in humans, and can rescue impaired learning in some mouse strains. Although there are a few studies on the effectiveness of memantine on learning behaviours, little is known about the underlying biochemical processes it targets. The authors tried to identify the proteins whose expression was affected by treatment with memantine.

The data was collected from 72 mice, of which 38 were wild-type, and 34 were partially tri-somic ("with three chromosomes" – these are mice with Down Syndrome, who cannot learn). The mice are otherwise clonal, so any differences in protein levels can be attributed to either the duplicated portions of their genome or to treatments they receive from the researchers.

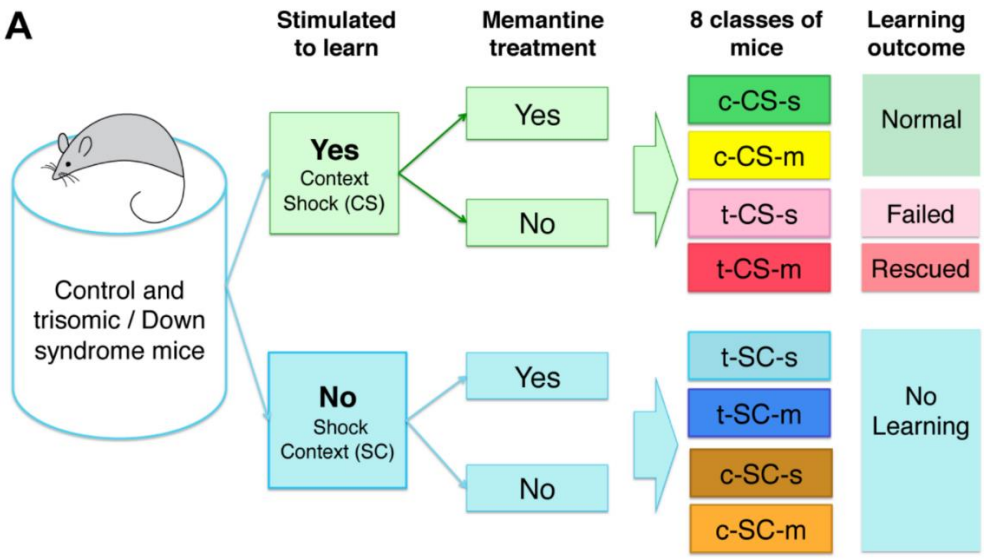

Figure 1. Source [https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4482027/pdf/pone.0129126.pdf]

Both, wild type and trisomic mice, were subjected to one of four treatments, as depicted in Figure 1, taken directly from the original paper:

1. A context-shock treatment to stimulate learning + a saline injection
2. A context-shock treatment to stimulate learning + a memantine injection
3. A shock-context treatment that doesn't stimulate learning + a saline injection
4. A shock-context treatment that doesn't stimulate learning + a memantine injection

This corresponded to eight classes of mice, each of which included between 7 to 10 mice.

After treatment, the mice were sacrificed and protein levels were measured in the cell lysates from their brain cortices. For each mouse, for each of the 77 proteins, 15 measurements were taken – a five-point dilution series, and three replicates at each dilution.

Problem Statement:

The authors mention that the reason they favour self-organizing maps (SOM) over other unsupervised learning methods like k-means, is that SOM can

- preserve the topology of the input dataset,
- provide a 2-dimensional visual representation of the data clusters, and
- produce the 'true' number of clusters instead of forcing all datapoints into a pre-determined number of clusters

My goal is to explore whether the combined methods of clustering and non-linear dimensionality reduction can provide equivalent results to the ones attained by the authors.

Combined, clustering and non-linear dimensionality reduction will preserve the topology of the original data, while still making it possible for us to visualise our clustering in a flat, 2-dimesnional figure.

Data Source:

I used the *Mice Protein Expression Dataset* from the UCI Machine Learning repository, and compared my results against the paper "Self-Organizing Feature Maps Identify Proteins Critical to Learning in a Mouse Model of Down Syndrome."

The dataset contains protein expression data of 77 proteins from 72 mice. 38 mice are controls and 34 are trisomic. Each mouse falls into one of eight classes, as previously described in Fig 1. For each mouse, there are 15 datapoints (rows), ie, 15 measurements per protein per mouse. This totals to 1080 rows in the dataset, when arranged in wide-format, according to the sketch shown in Figure 2.

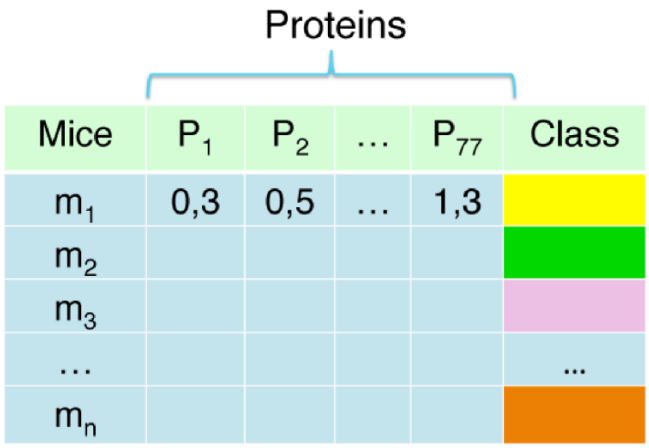

Figure 2. A graphic of the dataset. Source [https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4482027/pdf/pone.0129126.pdf]

Methodology:

Code:

I used built-in functions provided by Scipy and Scikit-learn to do my analysis.

Data Pre-processing:

I used the same imputation and standardization techniques as the authors of the paper, to maintain comparability.

High-throughput experiments such as these are automated using robots. In these situations, if some measurements fail, there's no feasible way to repeat them under the same experimental conditions, so the dataset will see missing data. I imputed such missing values with the average expression level of the corresponding protein for the same class of mouse, like the original authors.

One of the mice in the t-CS-m group (mouse 3417) has widely different protein expression levels than the others. Following the original authors, I treated this mouse as an outlier and removed it from the data-set.

Since different proteins naturally have different baseline expression levels in the animal, a comparison of absolute concentrations is inappropriate. Therefore, all protein levels were scaled to a 0-1 range.

Thresholding:

In order to reduce dimensions using ISOMAP, first I decided on an appropriate threshold for local distance. Based on heatmaps of adjacency matrices (Fig 3), I chose a threshold of 1.75 for constructing the weighted adjacency matrices, and compressed the data to two dimensions.

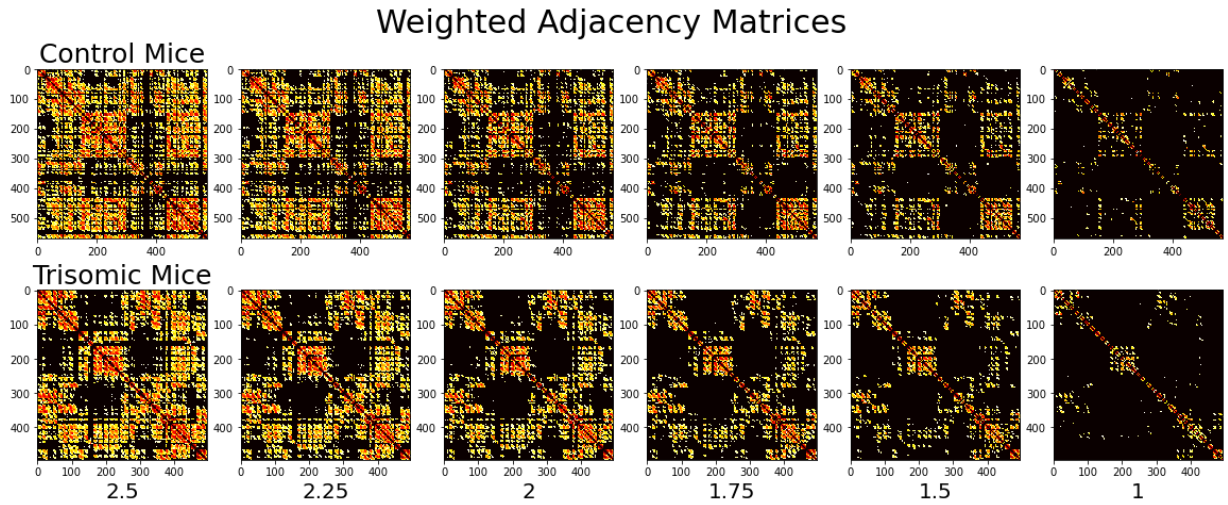

Figure 3: Heatmaps of adjacency matrices for selecting best distance threshold. Threshold values between 1 and 2.5 were tried.

Clustering:

Since the purpose of ISOMAP is visualization, it wasn't clear whether ISOMAP should be implemented before or after clustering, so I tried both methods. Additionally, in my proposal I had indicated that I plan to use spectral clustering, but I tried a few clustering algorithms, including spectral clustering, fuzzy c-means clustering, and DBSCAN, to find the best clusters.

In short, fuzzy clustering is a probabilistic clustering model, similar to Gaussian Mixture Models. I tried using this because the clusters are also not Gaussian, so using a GMM would not have worked, but I had hoped that a probabilistic clustering method would offset the noisy nature of the data. However, spectral clustering outperformed fuzzy clustering, so I've left any discussion about this out of the results section.

For a brief overview of the DBSCAN algorithm please refer to the third citation in the references section. In essence, DBSCAN is similar to spectral clustering, in that we need to build a connectivity graph (similar to a weighted adjacency matrix) to implement this. The difference is that we don't supply a pre-determined number of clusters – the algorithm chooses based on regions of sparsity in the dataset. Additionally, DBSCAN filters out some datapoints as noise.

Overall, implementing DBSCAN before compressing the data resulted in the best results, so this is the clustering method I employed. I clustered control mice separately from trisomic mice.

I only considered points that were not marked as noise by the DBSCAN algorithm. The DBSCAN algorithm yielded 8 clusters for both, the control and trisomic mice. Points in clusters containing more than one mouse class were treated as noise, i.e., these clusters were ignored, with one exception. Each

legitimate cluster was labelled with the identity of the mouse class corresponding to the data within it. If multiple clusters corresponded to the same type of mouse, they were combined.

After all combinations, I had 8 clusters, each of which corresponded to one mouse class.

Compare Protein Expression Levels Between Clusters

For each pair of classes outlined in Table 1, I performed a Wilcoxon rank-sum test on each protein (77 tests per pair of mouse classes), to determine if there were any significant differences in the protein expression levels across mouse classes. Proteins with p-value less than 0.05 were identified as class-discriminating.

Groups to Compare	Biological Interpretation
c-CS-s vs c-SC-s	Effects of CFC training in saline treated controls
c-CS-m vs c-SC-m	Effects of CFC training in memantine treated controls
c-SC-m vs c-SC-s	Baseline effects of memantine in controls
c-CS-m vs c-CS-s	Effects of memantine on learning in controls
t-CS-s vs t-SC-s	Effects of CFC training in saline treated trisomics
t-CS-m vs t-SC-m	Effects of CFC training in memantine treated trisomics
t-SC-m vs t-SC-s	Baseline effects of memantine in trisomics
t-CS-m vs t-CS-s	Effects of memantine on learning in trisomics

Table 1: Groupwise comparisons

Results:

Clustering:

I compressed the full dataset to 2 dimensions using ISOMAP, using a threshold local Euclidean distance as 1.75, and coloured the points according to the true classes (Fig 4).

Then, I applied both, spectral clustering and DBSCAN to the original dataset and to the compressed dataset, and labelled datapoints according to the new class assignments (Fig 5).

Spectral clustering on the datapoints prior to compression gave better results, but only marginally. Neither of the spectral clustering results really agreed with the true class labels, because the true class labels are quite mixed in Euclidean space (Fig 4).

Overall, I had the best luck with applying DBSCAN before compression, but there were two drawbacks:

1. It classified a majority of the datapoints as noise.
2. It gave me 8 clusters instead of 4, for each mouse genotype (control vs trisomic).

I accepted the first drawback because despite the noise, the results resembled the true classes much better than spectral clustering (Fig 5).

To deal with the second drawback, I combined clusters into “superclusters” as described next.

For the control mice, I modified DBClusters as follows (Fig 7):

1. clusters 0, 1 and 3 are grouped to form the c-CS-m group
2. cluster 5 alone forms the c-SC-m group
3. clusters 6 and 7 are grouped to form the c-SC-s group
4. from cluster 2 we only keep data corresponding to c-CS-s.

For the trisomic mice, I modified DBClusters as follows (Fig 7):

1. cluster 0 alone forms the t-CS-m group
2. cluster 4 alone forms the t-CS-s group
3. clusters 1 and 2 are combined to form the t-SC-m group
4. clusters 5, 6 and 7 are grouped to form the t-SC-s group

I labelled the mixed cluster number 2 from the control mice (Fig 7) as a cluster corresponding to the mouse class c-CS-s because this cluster is the only non-noise cluster that c-CS-s appears in. From the plot (Fig 4), it's apparent that this mouse class is not cleanly separable from c-CS-m (blue and orange points), so I wouldn't have been able to get a single cluster corresponding only to it anyway.

I evaluated the closeness of these new clusters to the true classes by plotting them on the same compressed axes from ISOMAP (Fig 6).

Hereafter, "cluster" refers to a DBSCAN supercluster.

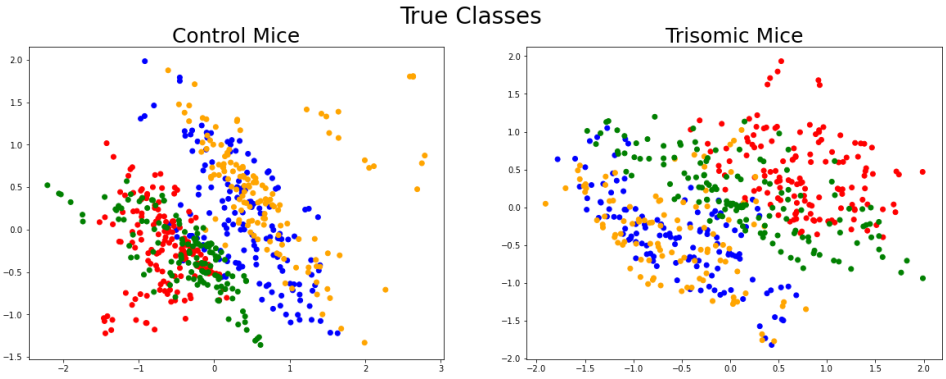

Figure 4: True class labels vs class labels assigned by spectral clustering

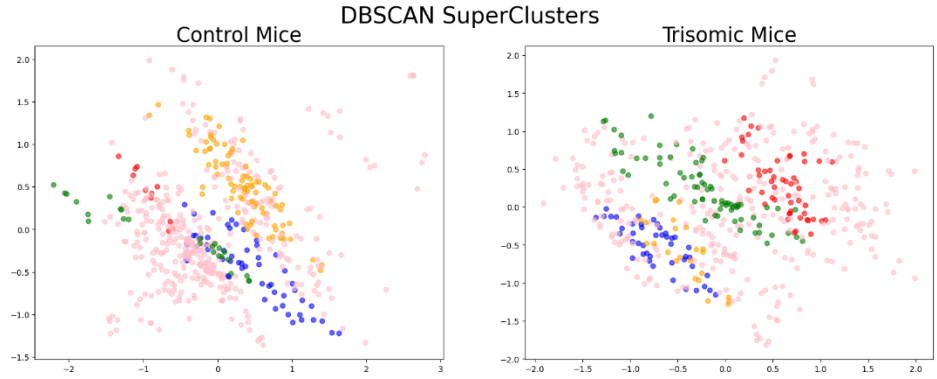

Figure 6: Combined DBSCAN clusters. Points designated as noise, or belonging to mixed clusters are colored light pink.

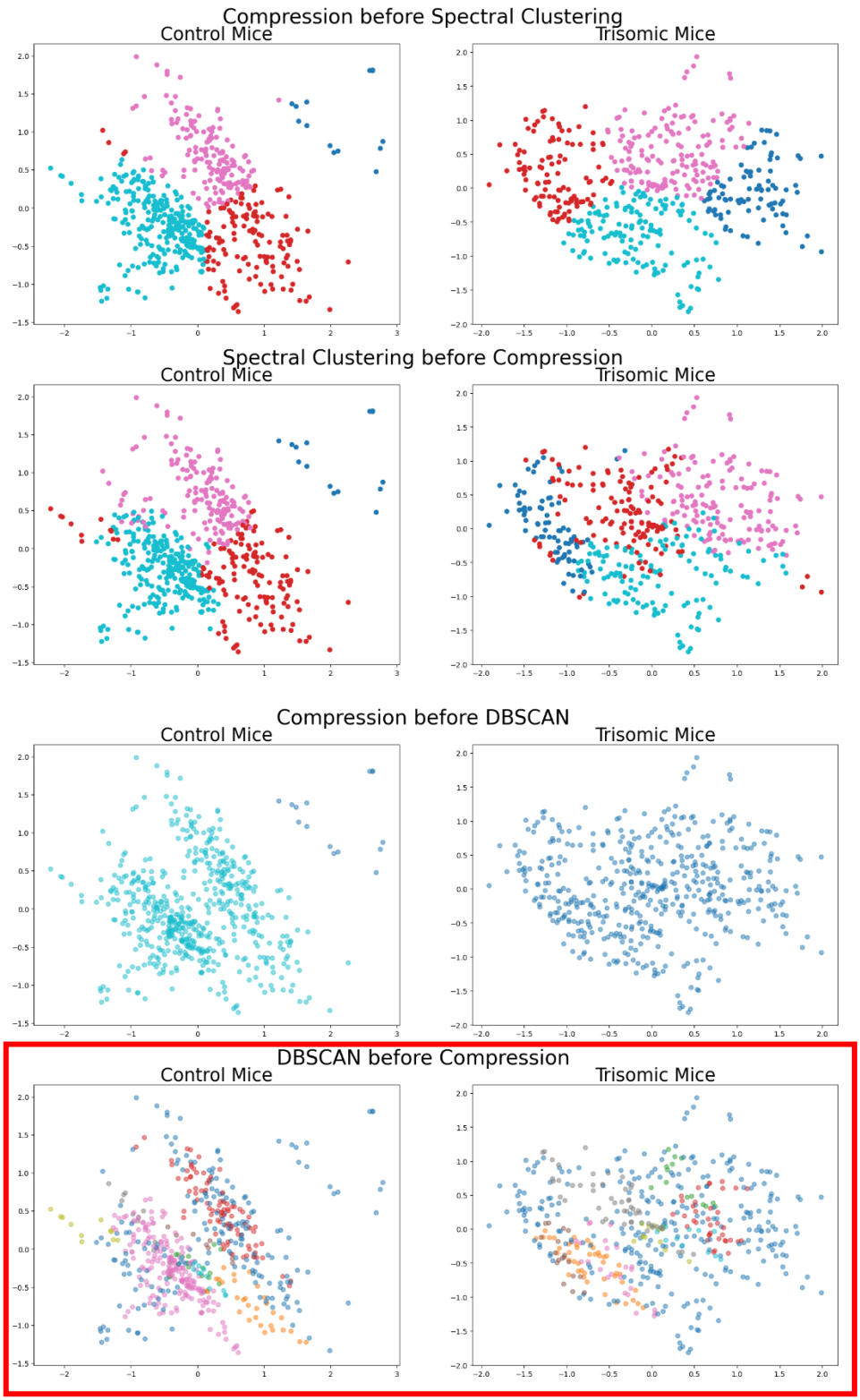

Figure 5: Predicted Class labels by Clustering algorithm and time of clustering. Applying DBSCAN before compression retained true classes the best.

Identifying Discriminating Proteins:

According to the original authors' analysis, proteins with p-value < 0.05 were considered to be significantly different between the two classes, and these proteins were identified as having class-discriminating properties.

The biological interpretation is that the expression of these proteins differs based on the condition of the mouse (genotype, learning condition, and drug treatment). These proteins may be studied as potential drug targets for developing pharmacotherapies.

Using their approach, the authors were able to really narrow down on the most important proteins, but my approach seems to mark most proteins as important (Table 2).

I have not presented the explicit lists of proteins generated here because they are quite tedious, but for each comparison, I have presented the number of significant proteins identified by my method, by the author's method, the intersection of the most significant proteins in my result and the author's result, and the intersection of my full result and the author's result in Table 2.

(a) Controls

DBCluster	Class	
-1	c-CS-s	92
	c-CS-m	53
	c-SC-m	31
	c-SC-s	21
0	c-CS-m	30
1	c-CS-m	16
2	c-CS-s	43
	c-CS-m	42
3	c-CS-m	9
4	c-SC-m	108
	c-SC-s	89
5	c-SC-m	11
6	c-SC-s	11
7	c-SC-s	14

(b) Trisomics

DBCluster	Class	
-1	t-SC-m	84
	t-CS-m	72
	t-CS-s	66
	t-SC-s	63
0	t-CS-m	44
1	t-SC-m	14
2	t-SC-m	37
3	t-CS-s	15
	t-CS-m	4
4	t-CS-s	24
5	t-SC-s	49
6	t-SC-s	12
7	t-SC-s	11

Figure 7: DBSCAN cluster compositions

Compared Groups	Number of Discriminant Proteins (DBSCAN)	Number of Discriminant Proteins (SOM)	Number of Proteins in Common with reduced list	Number of Proteins in Common with Complete LIst
c-CS-s vs c-SC-s	55	31	20	26
c-CS-m vs c-SC-m	68	23	13	20
c-SC-m vs c-SC-s	54	13	1	9
c-CS-m vs c-CS-s	70	12	1	11
t-CS-s vs t-SC-s	56	10	4	9
t-CS-m vs t-SC-m	59	36	28	35
t-SC-m vs t-SC-s	66	12	7	11
t-CS-m vs t-CS-s	45	9	3	7

Table 2: A comparison of the number of discriminant proteins identified by my method vs the original method.

The 'reduced lists' were generated by selecting the most significantly different proteins for each pair (the length of the list will be equal to the length of the author's list of significant proteins).

For example, for the first pairwise comparison, I extracted 31 discriminating proteins with the lowest p-values from my full list to create a reduced list, and compared this against the original author's list. The two lists had 20 proteins in common. On the other hand, comparing my full output to the authors output yielded 26 common proteins.

Evaluation:

The purpose of doing this analysis was to check if a clustering step and a dimensionality reduction step could replicate the results of the original paper, where a neural network was used to assign measurements to clusters.

The original method, using self-organizing feature maps definitely outperformed my methods, and was able to pick up on a lot of nuances about the relationships between the different classes of mice. On the other hand, although my method effectively discriminated between mouse classes based on protein levels, it could not clearly identify a smaller, workable subset of discriminant proteins.

Overall, although my method is not as performant as the original, my expanded lists seem to include most of the proteins in the original author's lists, especially for trisomic mice.

References:

1. Higuera, C., Gardiner, K. and Cios, K., 2015. Self-Organizing Feature Maps Identify Proteins Critical to Learning in a Mouse Model of Down Syndrome. *PLOS ONE*, [online] 10(6), p.e0129126. Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4482027/pdf/pone.0129126.pdf> [Accessed 3 July 2022].
2. Raman, R. and Hedeker, D., 2005. A mixed-effects regression model for three-level ordinal response data. *Statistics in Medicine*, [online] 24(21), pp.3331-3345. Available at: <http://bstt513.class.uic.edu/RamanStatMed.pdf> [Accessed 3 July 2022].
3. Fahim, A.M et al. An Enhanced Density Based Spatial Clustering of Applications with Noise. Proceedings of the 2009 International Conference on Data Mining, pp517-523. Available at: <https://www.researchgate.net/publication/220704900_An_Enhanced_Density_Based_Spatial_clustering_of_Applications_with_Noise> [Accessed 29 July 2022].

Dataset downloaded from: https://archive.ics.uci.edu/ml/datasets/Mice+Protein+Expression#