Self Organizing Systems
Exercise 3: SOM - Group 20

Michael Bernreiter
01307069
mbernr@outlook.com

Timothée Durand
11831520
durandtimothee@orange.fr

Giulio Pace
11835706
giulio.pace93@gmail.com

A) INTRODUCTION

The goal of the project is to train and analyse several self-organized maps (SOM) for a selected dataset, in order to understand the dataset's characteristics and the influence that a set of parameters can have on it. To this end we will use a variety of visualizations that will allow us to investigate different aspects of the SOM and of the dataset. To train the SOMs and generate the visualizations we will use "*JavaSOMToolbox*" , a tool developed at TU Wien specifically for this purpose.

B) DATA DESCRIPTION AND ANALYSIS

The dataset that we picked for the project is called "Mice-Protein" (https://www.openml.org/d/40966). It describes the expression levels of 77 proteins measured in the cerebral cortex of mice.

There are 8 classes of mice with down syndrome and without (control group) exposed to context fear conditioning, a task used to assess associative learning. Some of them were also injected with saline, while others with memantine. Therefore, the classes of mice are distinguished by three features: genotype, which can be trisomic (t) or control (c); behavior, which can be stimulated to learn via context-shock (CS) or not (SC); treatment, which can be injected with memantine (m) or with a saline solution (s).

In Table 1 we can see how the individuals are distributed over the classes. There are 38 control mice and 34 trisomic mice (affected by the Down Syndrome), for a total of 72 mice. For each mice, 15 measurements have been performed for each protein, so the dataset contains 1080 measurements per protein. Each measurement can be considered as an independent sample/mouse.

The goal is to identify subsets of proteins that are discriminant between the classes.

1) Description of the attributes

The first attribute is the ID of the Mouse.
Attributes from 2 to 78 represent the values of expression levels of the 77 proteins; all of these attributes are numeric. The names of proteins are followed by the string "_n", indicating that they were measured in the nuclear fraction. E.g.: DYRK1A_n.
Attributes 79, 80 and 81 are binary attributes that identify genotype, treatment type and behavior respectively. These attributes can be ignored for the purposes of the data analysis as the same information can be obtained through the class attribute.
Finally attribute 82 is the class attribute and can take one of the following values: c-CS-s, c-CS-m, c-SC-s, c-SC-m, t-CS-s, t-CS-m, t-SC-s, t-SC-m.

There are some missing values in the dataset. All attributes from 2 to 43 have exactly 3 missing values, with the exception of ELK_N and Bcatenin_N that have 18 missing values. Towards the end there are 5 attributes with a big number of missing values (around 200). The remaining attributes have no missing values.

As mentioned, all of the protein attributes are numeric. Their value almost always spans from 0 to +2 with some exceptions (ranging from 0 to 5). More details can be seen in Tables 55 and 56 in the appendix.

2) Preprocessing

The data was provided as a *csv* file. First of all, for each attribute we substituted the missing values with the mean of its values.

We then used the *z-score* to perform a standardization of the data. This operation was needed because of the presence of some attributes that have a completely different range from the others. We chose *z-score* over the min-max approach because it deals with outliers better. Furthermore, *z-score* is the obvious choice because of the nature of the attributes. Since each attribute represents the level of a protein, it makes sense to represent it as the distance from the "normal value".

We finally wrote a python script that generated the *.vec*, *.tv* and *.cls* files as required by *JavaSOMToolbox*.

C) SOM TRAINING AND ANALYSIS

1) Base case: Regular SOM

After some initial experimentation with the parameters, we decided to train a basic SOM with the parameter values outlined in Table 2. For the map size, we chose a ratio of 1:5 between the number of units and input vectors. We also experimented with smaller/bigger maps, but this ratio seemed to be a good balance between density and sparsity. The learning rate and initial neighbourhood radius (sigma) were kept at their default values, after some experiments with

Table 1: Classes distribution and color code

c-CS-m	control mice	stimulated to learn	injected with memantine	10 mice
c-CS-s	control mice	stimulated to learn	injected with saline	9 mice
c-SC-m	control mice	not stimulated to learn	injected with memantine	10 mice
c-SC-s	control mice	not stimulated to learn	injected with saline	9 mice
t-CS-m	trisomy mice	stimulated to learn	injected with memantine	9 mice
t-CS-s	trisomy mice	stimulated to learn	injected with saline	7 mice
t-SC-m	trisomy mice	not stimulated to learn	injected with memantine	9 mice
t-SC-s	trisomy mice	not stimulated to learn	injected with saline	9 mice

cCSm	
cCSs	
cSCm	
cSCs	
tCSm	
tCSs	
tSCm	
tSCs	

smaller/bigger values. The number of iterations is 5 times the number of input vectors, as recommended in the Java-SOMToolbox step-by-step guide.

Table 2: Parameters for basic SOM

Parameter	Value
Random seed	42
x-size	15
y-size	15
Learning rate	0.75
Sigma	1
Num iterations	5400

The class distribution can be seen in Figure 1. It is evident that classes are not perfectly separated from each other on the map, but they do form small groups, i.e. there is some noticeable structure in the class distribution.

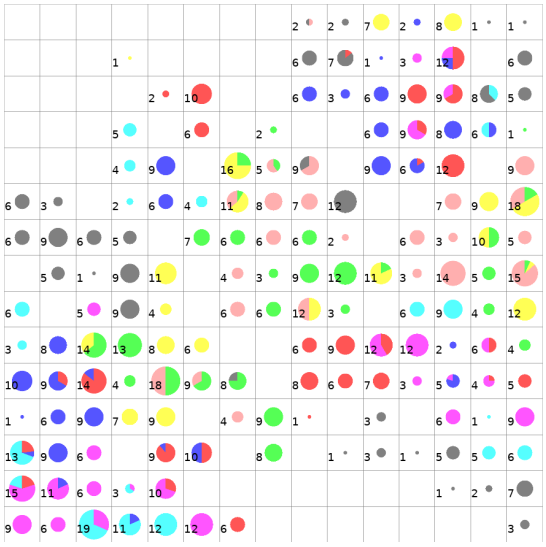


Figure 1: Class distribution in basic SOM

In Figure 2, one can see the result of a Ward's linkage (fast version) clustering with 8 clusters. Clusters do not seem to correlate with the class distribution. Most clusters contain several classes, and some groups of classes are contained in several clusters. Moreover, there is one big cluster in the center. When the number of clusters is increased to 16, as can be seen in Figure 3, this big cluster in the center is divided into several smaller clusters. These clusters still do not correlate perfectly with the classes, but now there are some clusters that consist mainly (or even exclusively) of one class.

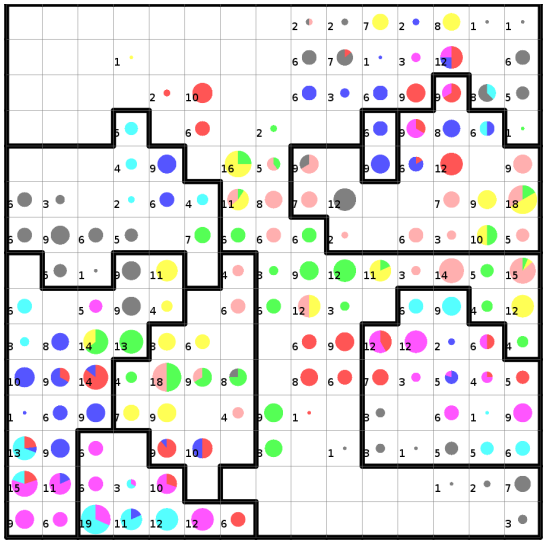


Figure 2: Class distribution and clusters in basic SOM, 8 clusters

However, when grouping the classes by only one of their defining characteristics at a time, a clearer picture emerges. For example, Figure 4 shows the classes partitioned by their genotype. One can see that more clusters (approximately half of them) are now dominated by a given genotype. The cluster to the bottom left-center consists of mainly trisomic genotypes. When grouping the classes by their behavior, as visualized in Figure 5, the clusters seem to strongly correlate with the behavior type. Finally, partitioning the classes by their treatment has a less clear cluster correlation than

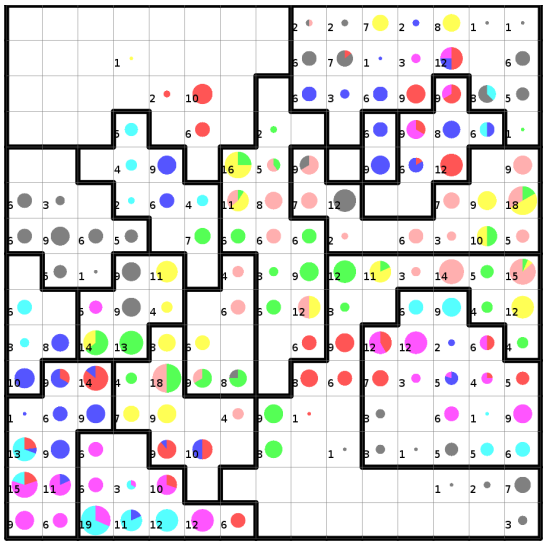


Figure 3: Class distribution and clusters in basic SOM, 16 clusters

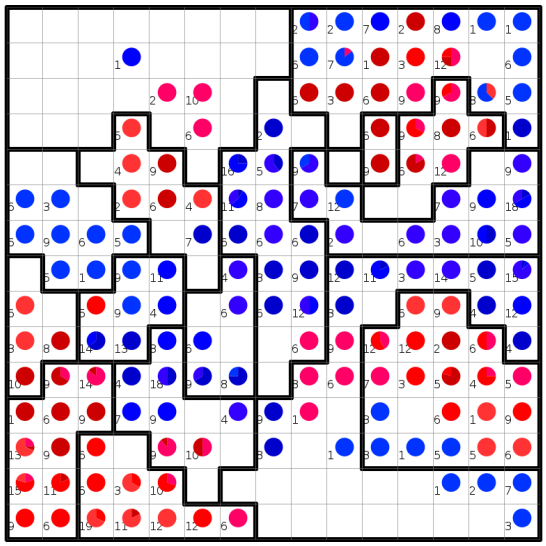


Figure 5: Behavior distribution and clusters in basic SOM (red = Context-Shock, blue = Shock-Context)

the behavior distribution, but still approximately half of the clusters are dominated by one treatment type (Figure 6).

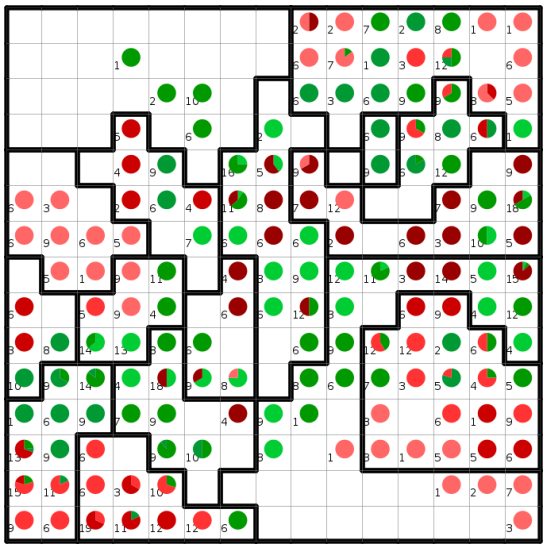


Figure 4: Genotype distribution and clusters in basic SOM (red = trisomic, green = control)

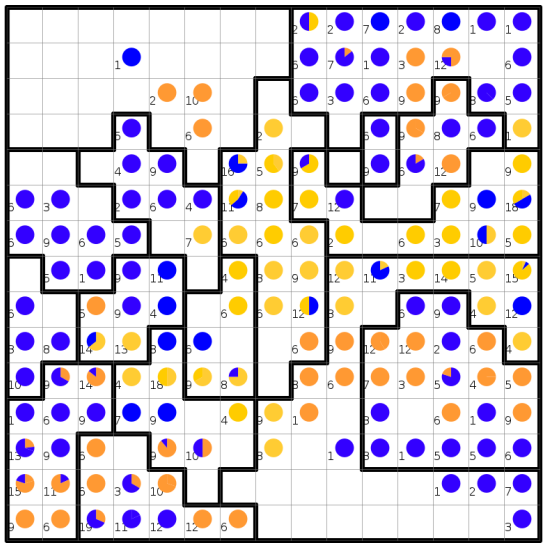


Figure 6: Treatment distribution and clusters in basic SOM (blue = saline, orange = memantine)

The mean quantization error is visualized in Figure 7. Most units have a small error (blue), with only a few units having a big error (red). This means that most units are close to their matched input vectors in the input space. There seems to be no correlation between cluster quality and quantization error. However, our trained map exhibits significant topology violations, as can be seen in Figure 8. This indicates that vectors that are close to each other in the input space are not close to each other in the output space, and therefore that the topology of the input space is not captured well by the map.

The border effect can be observed by the fact that there are more detailed clusters on the border areas of the map. This is especially visible with a smaller number of clusters (see Figure 2). The magnification factor is visualized clearly on Figure 9. Indeed, we can see that there are some units in very dense (blue) areas that have approximately the same

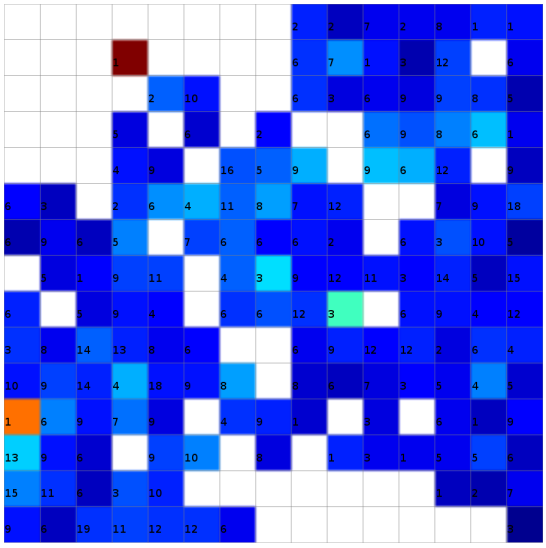


Figure 7: Mean quantization error in basic SOM (small error in blue, high error in red)

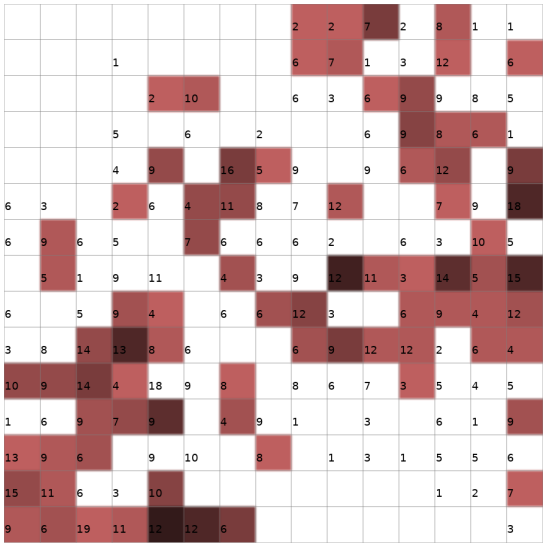


Figure 8: Topographic error in basic SOM

number of matching input vectors as units in less dense (red) areas. For this to happen, there has to be more units in these dense areas, therefore the denser zones of the input space are magnified.

2) Different initializations of the SOM

For this experiment, we trained two further SOMs, using the same parameters as before except for the random seed, which is set to 791 in the first map (SOM791) and to 398 in the second map (SOM398). For the newly generated maps, we notice that their class distribution has visibly changed in

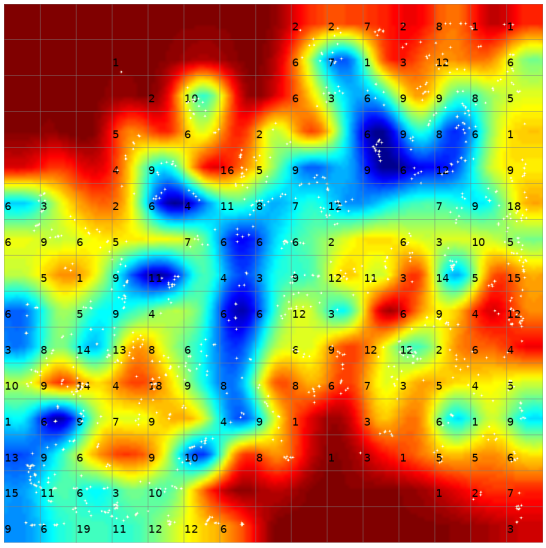


Figure 9: Interpolated P-matrix overlayed with sky metaphor

comparison to the original map (SOM42) with seed 42 (See Figures 3, 10, and 11).

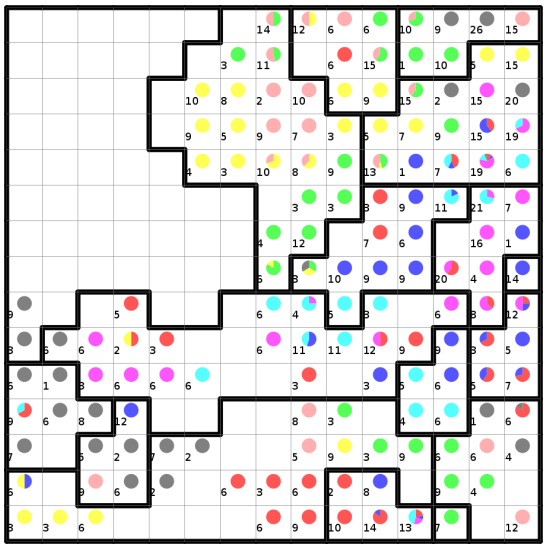


Figure 10: Class distribution and clustering on map with seed 791

In these maps, one can also see that the various SOMs have different structures. For example, SOM791 has a very large empty area (in the top left corner), while SOM42 has two empty areas of smaller size.

Figure 12 shows the data shifts from SOM42 to SOM791. Most of the units are mapped to similar data-points across the two maps, which can be seen by the fact that there are

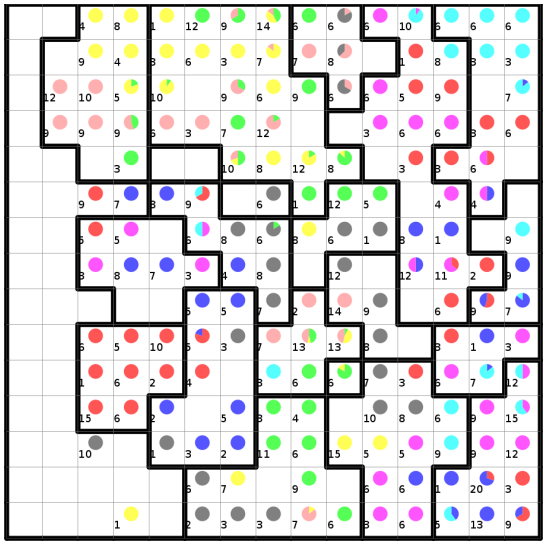


Figure 11: Class distribution and clustering on map with seed 398

more green than red arrows. In Figure 13, more red arrows can be observed compared to Figure 12, meaning that the data has shifted more between SOM42 and SOM398.

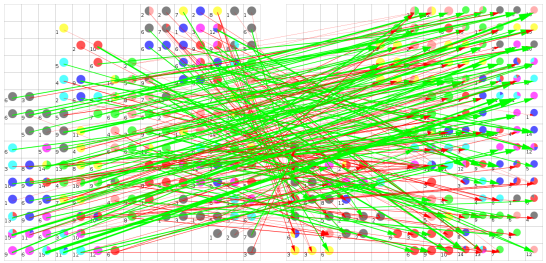


Figure 12: Data shifts from basic SOM (seed 42) to SOM with seed 791

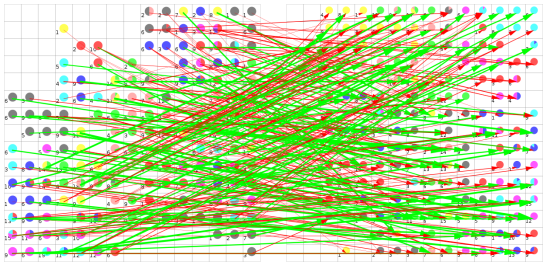


Figure 13: Data shifts from basic SOM (seed 42) to SOM with seed 398

When analyzing the data shifts in more detail, we discovered that clusters on the border seem to be mapped similarly across different maps, while clusters in the center experience more data shifts. This observation is exemplified by the two visualizations in Figures 14 and 15.

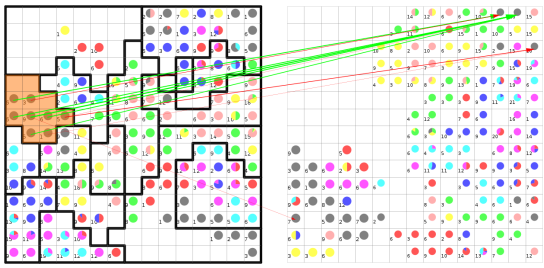


Figure 14: Cluster shift from map with seed 42 to map with seed 791

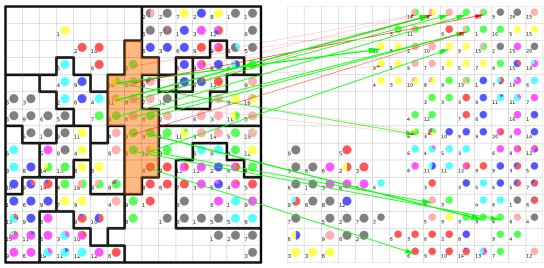


Figure 15: Cluster shift from map with seed 42 to map with seed 791

When examining topology violations, we compared the topographic error between the three maps (see Figures 8, 16 and 17). We observed that SOM791 has less units with extreme topographic error compared to the other two maps. However, the three maps show a similar level of topology violations overall.

3) Different map sizes

In this section, we are comparing SOMs of different sizes. To this end, we train a very small SOM and a very large SOM (see Table 3). The small map has an input vector to unit ratio of 1:16, which is approximately three times smaller than the basic map. This seemed reasonably small, but still big enough to be meaningful. The large map has 900 units (for 1080 unit vectors), i.e. almost a 1:1 ratio. When choosing the learning rate, we decided to use a value of 1.0 for both maps. As for sigma, the small map has a value of 1, which is rather large, considering that we used the same value for our regular SOM. The large map has a sigma value of 10.

When comparing class distribution of the small map to that of the large map (see Figures 18 and 19), we can see that on the bigger map, the classes are concentrated in more distinct groups. Observe for example the big group of grey

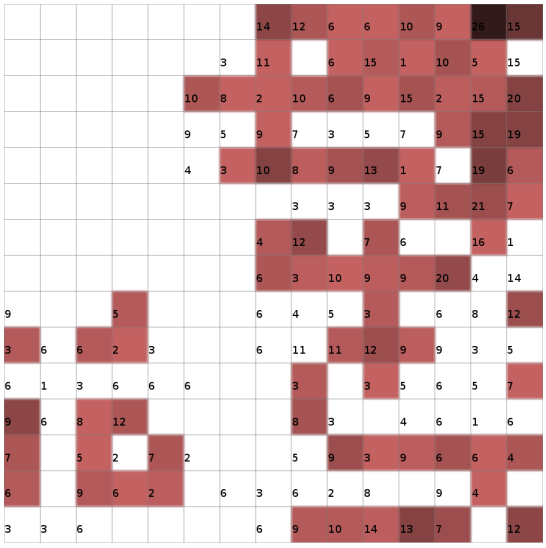


Figure 16: Topographic error on SOM with seed 791

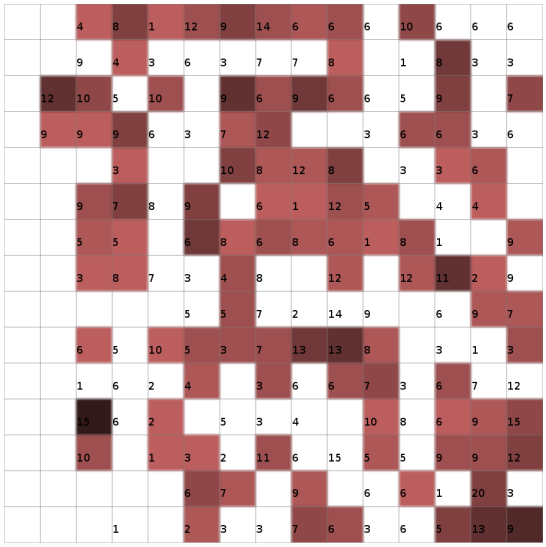


Figure 17: Topographic error on SOM with seed 398

Table 3: Parameters for very small SOM and very large SOM

Parameter	Small	Large
Random seed	42	42
x-size	8	30
y-size	8	30
Learning rate	1.0	1.0
Sigma	1	10
Num iterations	5400	5400

units in the center of the large map. However, even on the small map, there is at least some structure regarding class distribution. This indicates that on a bigger map, there is more space for the different classes to form distinct groups without interfering with each other. Interestingly, the quality of cluster structure is not better on the bigger map. For example, the grey group in the center of the large map is split into two clusters, while in the small map there are some clusters that consist of only one class. This might be due to the fact that the clusters contain less units the small map.

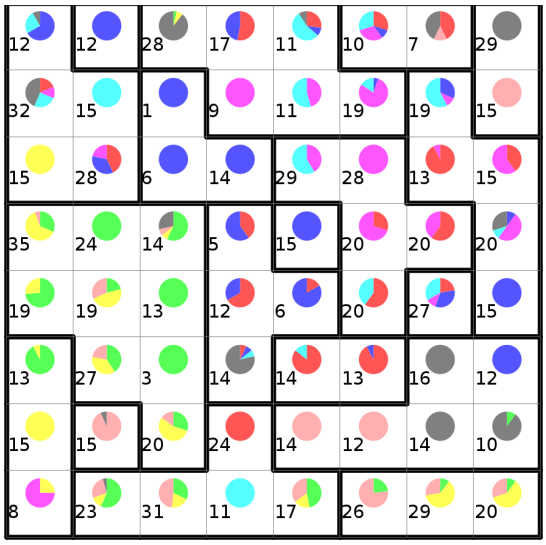


Figure 18: Class distribution and clusters for small SOM

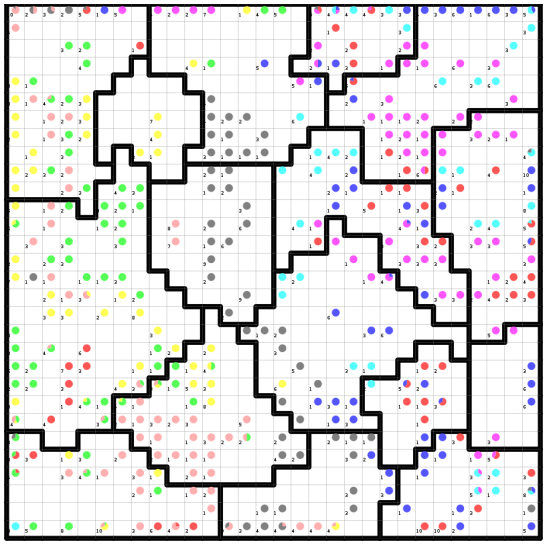


Figure 19: Class distribution and clusters for large SOM

Additionally, we compared how the data-points of single clusters in the regular map distribute to the large map (see Figures 20 and 21). We can not confirm our observation from the first experiment that clusters on the border exhibit less dramatic shifts than clusters in the center. This might be due to the difference in map size between the two SOMs.

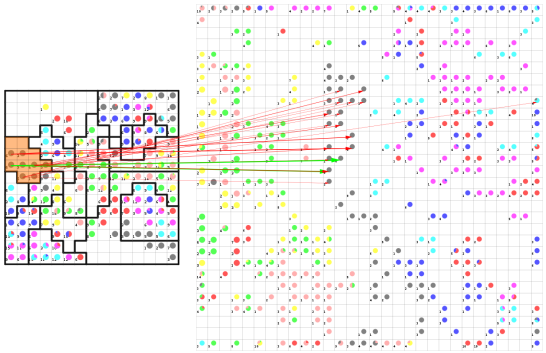


Figure 20: Cluster shift from regular map to large map

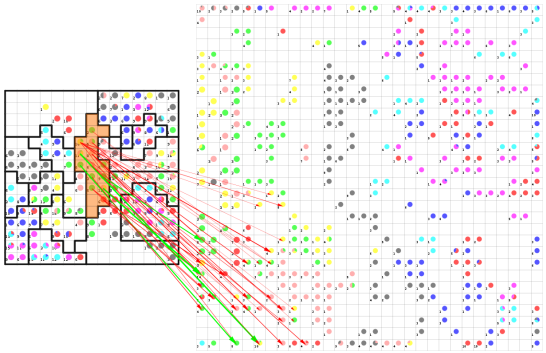


Figure 21: Cluster shift from regular map to large map

Regarding the mean quantization error, we can see that both the small map and the large map exhibit a small error (see Figures 22 and 23). In fact, on the large map, there is a higher percentage of units with a bigger quantization error (yellow/green).

We also compared the topographic error of the small map and the large map (see Figures 24 and 25). In the small map, we can see a lot of topology violations, while in the large map only a few units exhibit topology violations. This might be because with more units, the topology of the input space can be represented more accurately.

To compare the magnification factor between the small and large SOM, we used the P-matrix visualisation (see Figures 26 and 27). We observed that on the small map, the magnification factor is far more present than on the large map. For example, in the center of the small map, there is a very dense area, where 2 units cover 21 input vectors, while in a less denser area on the left, 2 units cover 50 input vectors. In the large map, this effect is not as prominent. This might be due to fact that in the large map there are enough units to show even less dense areas in detail.

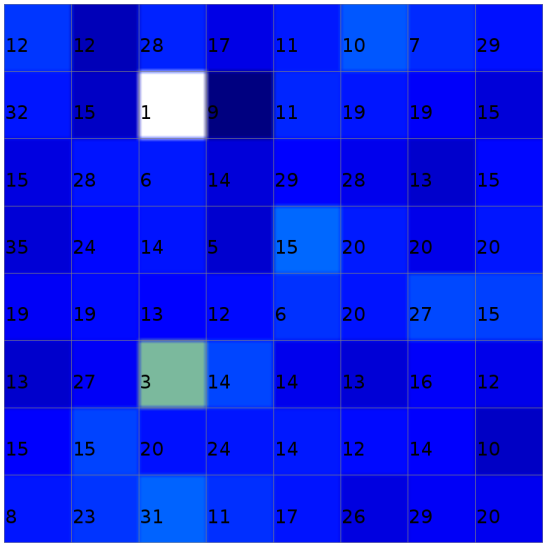


Figure 22: Mean quantization error for small SOM

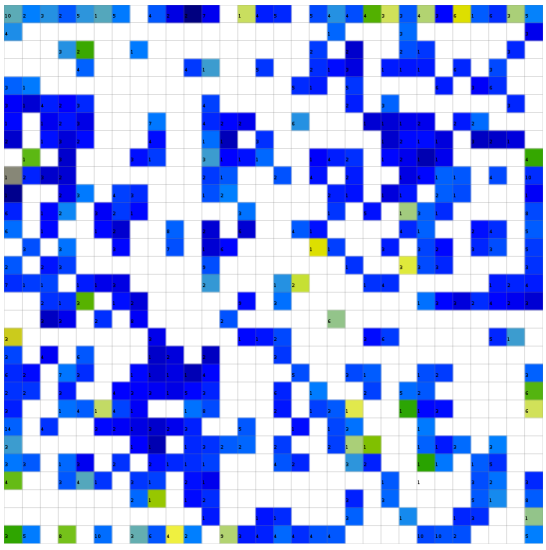


Figure 23: Mean quantization error for large SOM

4) Different initial radius setting

To analyze the impact of different initial radius settings (sigma), we trained two different maps: a regular map with much too large sigma and a large map with a much too small sigma (see Table 4).

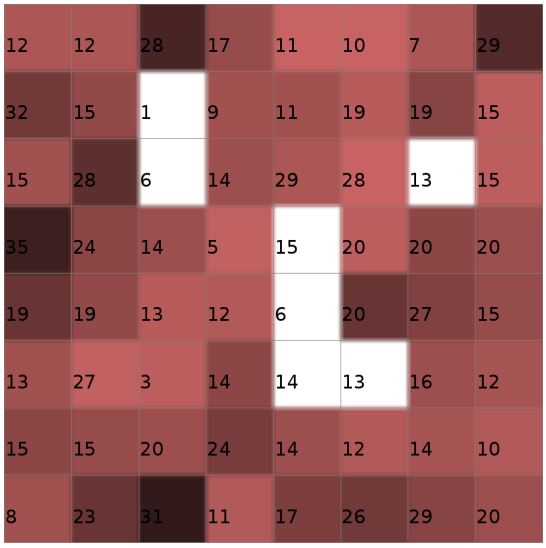


Figure 24: Topographic error for small SOM

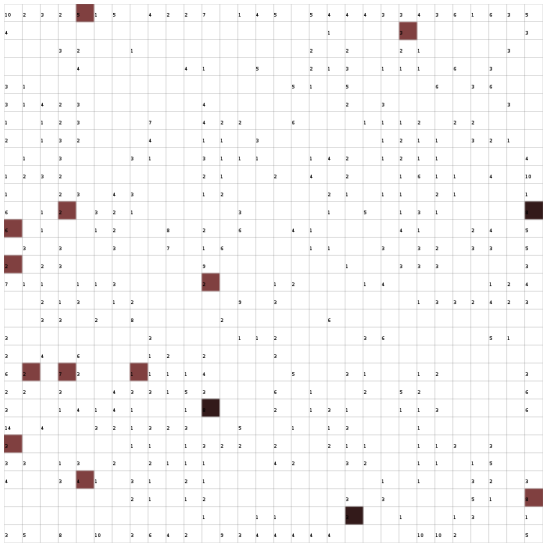


Figure 25: Topographic error for large SOM

Table 4: Parameters for regular map with large sigma and large map with small sigma

Parameter	Regular	Large
Random seed	42	42
x-size	15	30
y-size	15	30
Learning rate	0.75	0.75
Sigma	30	0.1
Num iterations	5400	5400

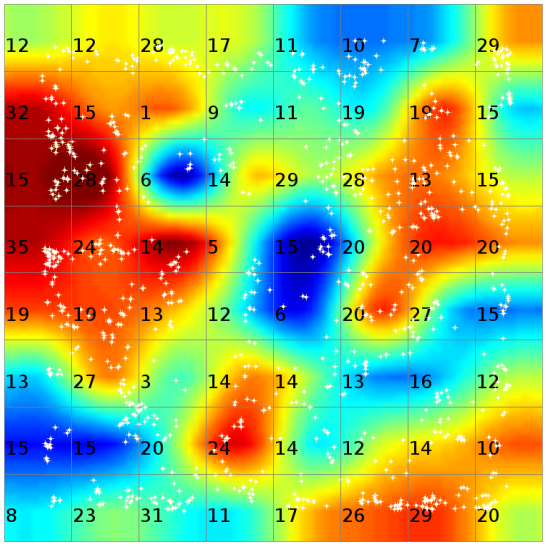


Figure 26: P-matrix with sky metaphor for small SOM

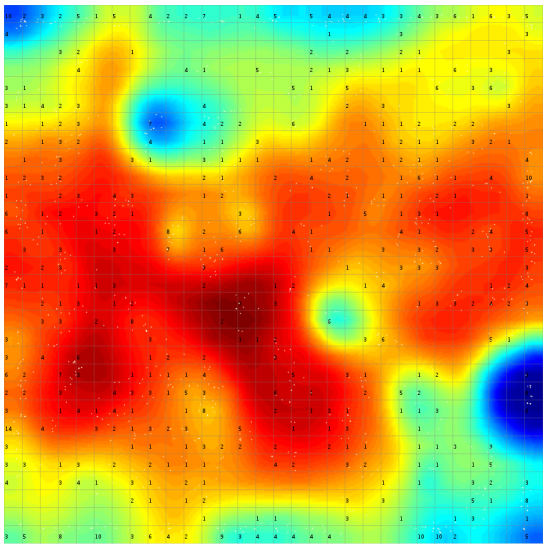


Figure 27: P-matrix with sky metaphor for large SOM

For the large SOM, we can see that we have clusters consisting of only one unit, and indeed all of the data-points are matched to a small number of isolated units (see Figure 29). This might be because dense areas in the input space are initially matched to a single unit. Since there is a small neighborhood radius, the data-points in these dense regions do not influence their second best unit, and are not magnified. We used 16 clusters for the large map, but we expect every unit to form its own cluster, if enough clusters would be used. In the regular SOM, we see a very different pattern. The clusters have very similar shapes and sizes, and are very well distributed over the map (see Figure 28).

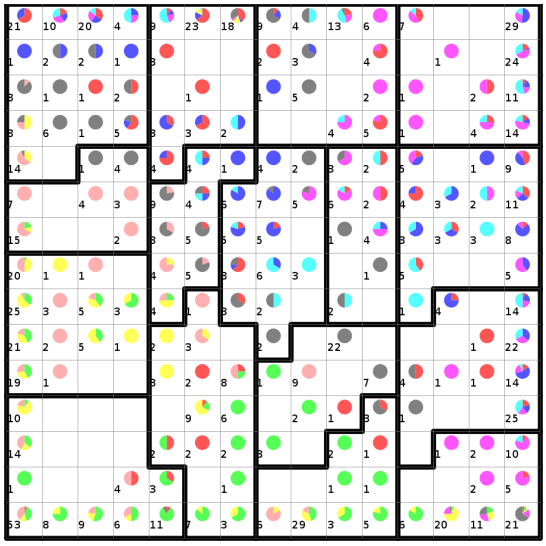


Figure 28: Class distribution and clusters for regular map with large sigma

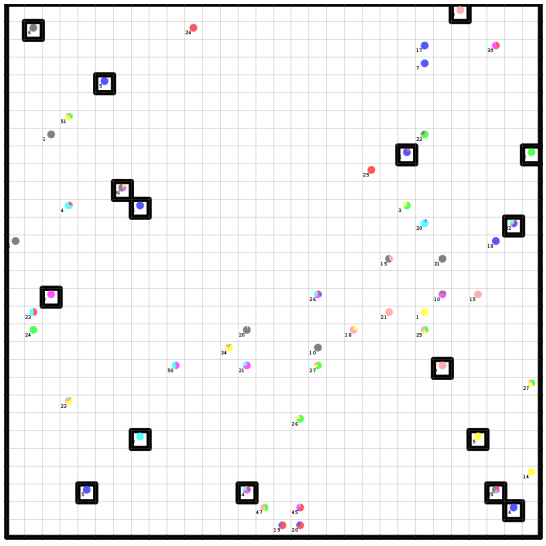


Figure 29: Class distribution and clusters for large map with small sigma

Secondly, we used the quantization and topographic error visualizations to analyze the quality of the SOMs. For the regular map (Figure 30), we observe a slight increase in the quantization error compared to the first experiment. However, we see a real improvement of the topographic error (see Figure 31), since only a few units display errors. This could be because with a large neighborhood radius, an input vector influences not only its best matching unit but also its neighbors. Therefore, it is unlikely that the 2nd, 3rd... best matching units end up far away in the SOM structure.

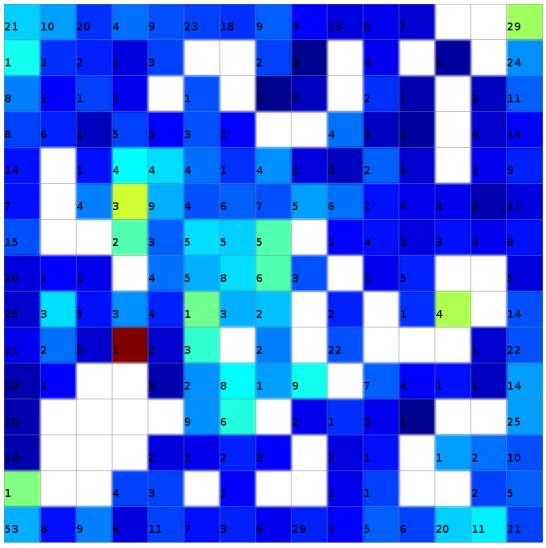


Figure 30: Quantization error for regular map with large sigma

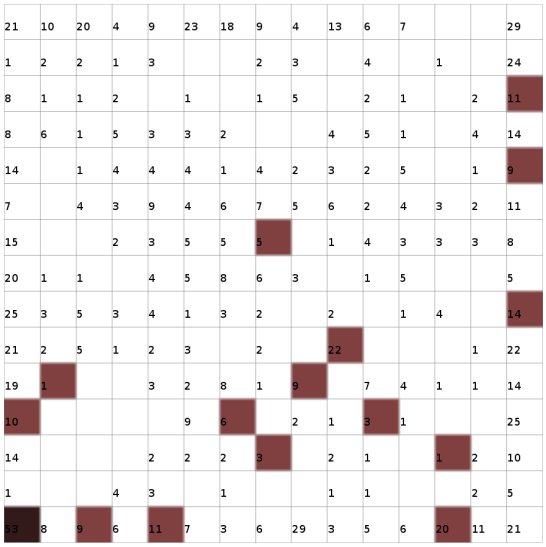


Figure 31: Topographic error for regular map with large sigma

For the large map, the quantization error is quite low (see Figure 32). In contrast, there are a lot of topology violations (see Figure 33). Every unit that is matched with at least some input vector has a large topographic error. Since the neighbors of the best matching unit will not be influenced as much using the small neighborhood radius, the topology can not be learned from the random initialization. We can conclude that when the neighborhood radius is too small, very few units are matched to input vectors.

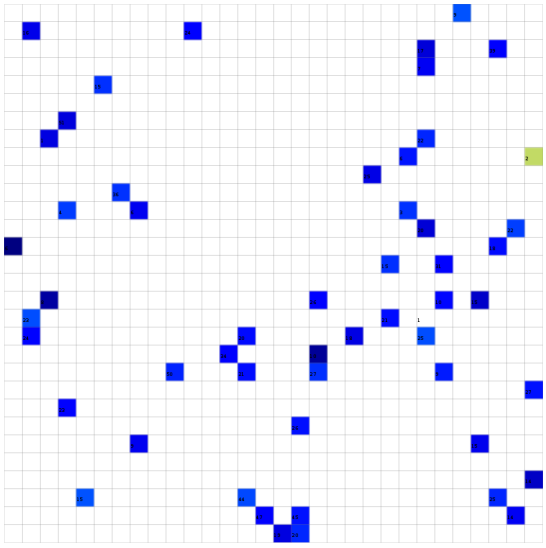


Figure 32: Quantization error for large map with small sigma

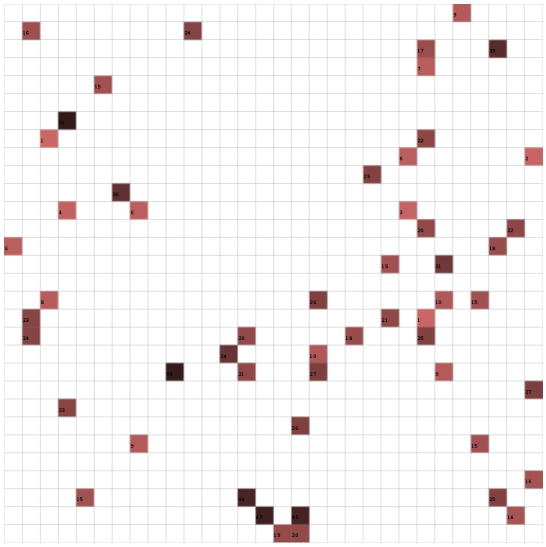


Figure 33: Topographic error for large map with small sigma

5) Different initial learning rates

To analyze the impact of different initial learning rates, we trained two different maps: a regular map with a big learning rate and a large map with a small learning rate (see Table 5).

The cluster structure of the regular map can be seen in Figure 34. Overall, it is not very different from the cluster structure in our first experiment. However, we can notice that when using the bigger learning rate, we have less empty units on the regular map overall. As for the large map (Figure 35), we can see that all data-points are matched to a small number of units, and that these non-empty units are grouped together in four clusters. This differs from the large map with small sigma in the previous experiment, where the non-empty units were isolated.

Table 5: Parameters for regular SOM with big learning rate and large SOM with small learning rate

Parameter	Regular	Large
Random seed	42	42
x-size	15	30
y-size	15	30
Learning rate	1.0	0.1
Sigma	1	1
Num iterations	5400	5400

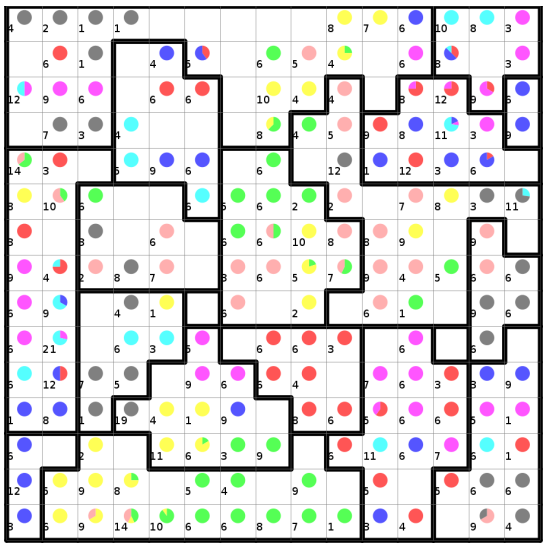


Figure 34: Class distribution and clusters for regular map with big learning rate

We again analyzed the quality metrics on the regular map. The quantization error (Figure 36) is comparable to previous experiments with the regular map size. The topographic error (Figure 37) is comparable to the base case, but a lot higher than for the regular map with a large sigma.

Again, the quantization error is quite low for the large map (Figure 38), and the topographic error is quite high (Figure 39).

We conjectured that for the large map, the learning rate is too low only with respect to the number of iterations used for training. We tried to confirm this hypothesis by using 1 million training iterations on the large map while keeping the small learning rate. As expected, we observed that the data is distributed more evenly over all units. Hence, a too

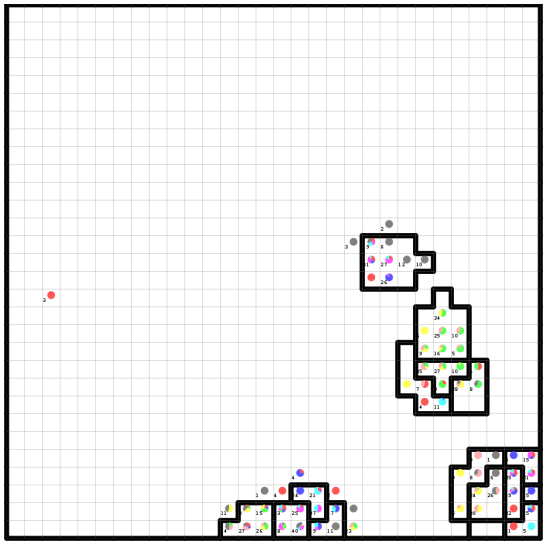


Figure 35: Class distribution and clusters for large map with small learning rate

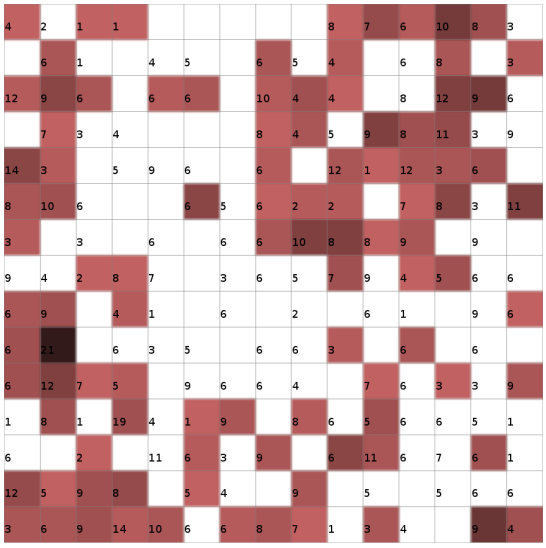


Figure 37: Topographic error for regular map with big learning rate

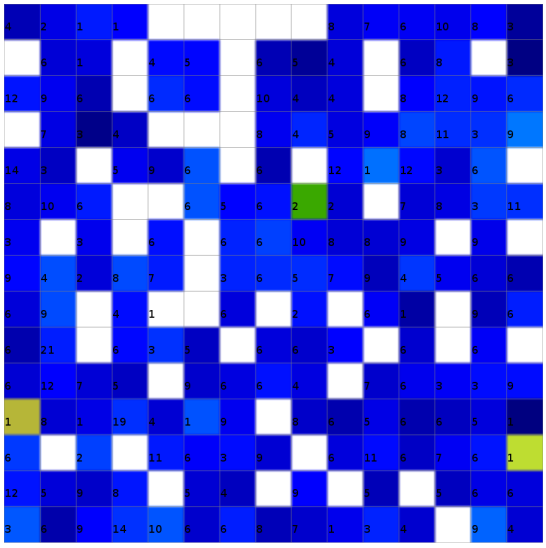


Figure 36: Quantization error for regular map with big learning rate

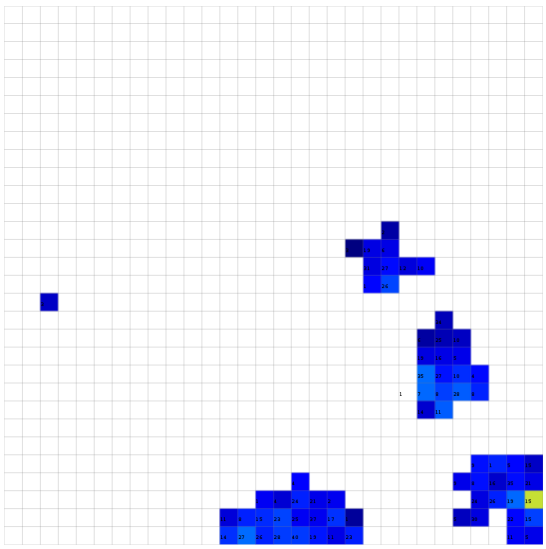


Figure 38: Quantization error for large map with small learning rate

small learning rate is characterized by small concentrated clusters, with input vectors poorly distributed across the SOM.

6) Different scalings

For this experiment, we trained a map with erroneous scaling: Instead of applying the z-score method, we multiplied each column (attribute) with a random number between 1 and 100. Otherwise, we used the same parameters as in the first experiment.

The class distribution and cluster structure can be seen in Figure 40. The input vectors are grouped together in the bottom left corner of the map. Interestingly, some of the clusters consist of predominately one class, i.e. the classes are distributed fairly well over the SOM.

When looking at the quantization error (Figure 41) and the topographic error (Figure 42), the problems of this erroneous scaling approach become apparent. We see the biggest quantization error so far across all conducted experiments and a very high topographic error. This indicates that the

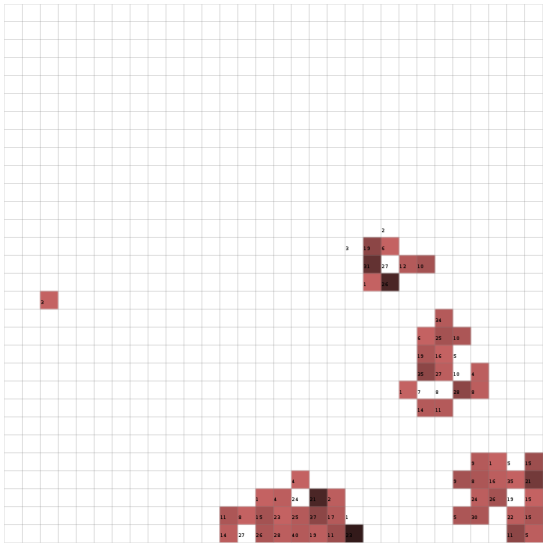


Figure 39: Topographic error for large map with small learning rate

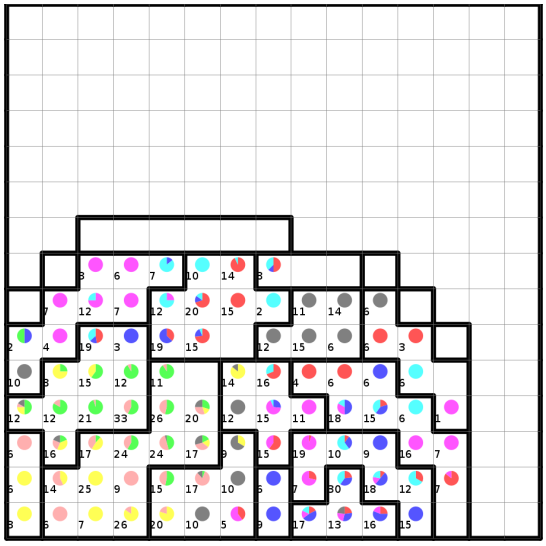


Figure 40: Class distribution and clusters for SOM with wrongly scaled data

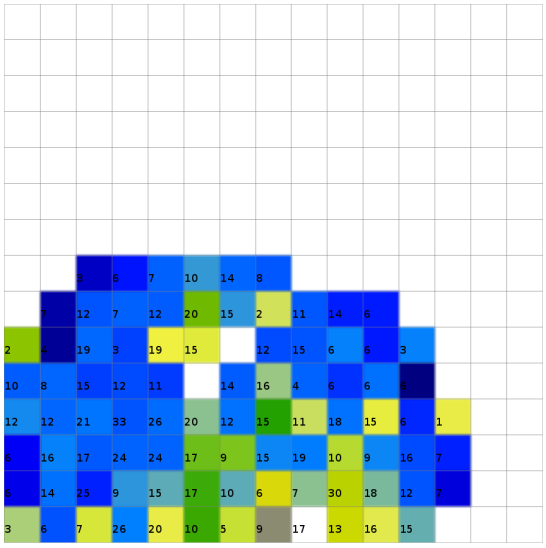


Figure 41: Quantization error for SOM with wrongly scaled data

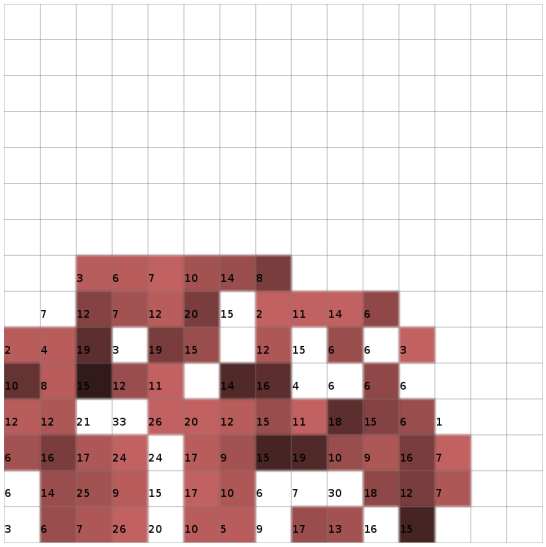


Figure 42: Topographic error for SOM with wrongly scaled data

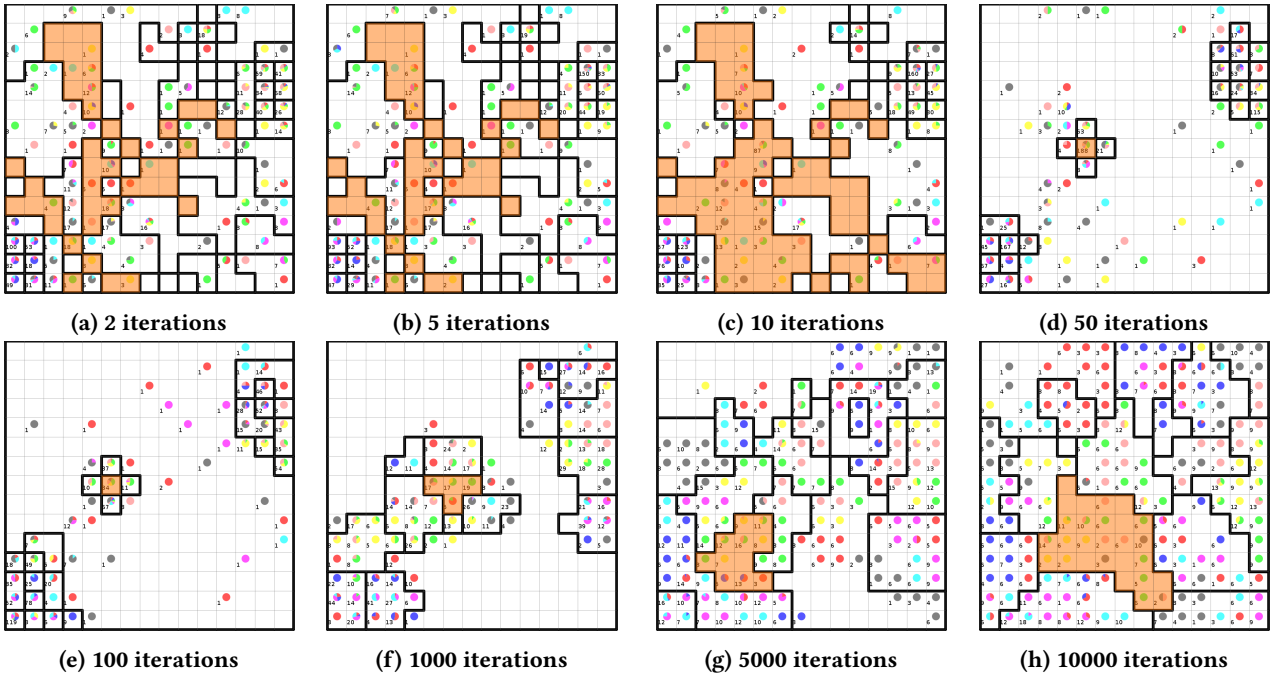
SOM does not represent the data well, which is an argument in favor of correct scaling methods, such as z-score.

7) Different maximum iterations

For this experiment, we trained 8 SOMs, each with a different number of iterations (2, 5, 10, 50, 100, 1000, 5000, 10000). The other parameters are the same as in the first experiment).

In Figure 43, we can see how the cluster are formed and how they evolve through the iterations. One can see that before the 10 iteration mark they are very scattered. Observe for example the highlighted cluster in orange. The first noticeable improvement appears to happen at 50 iterations, when the clusters are very small but distinct. At 1000 iterations we can start to see bigger clusters, and at 5000 iterations the clusters are overall well formed. After 10000 iterations they are fully formed and have a comparable size. It is clear that a minimum number of 1000 iterations is needed.

In Figure 44 it is possible to see how the mean quantization error changes across iteration numbers. At the beginning



(a) 2 iterations (b) 5 iterations (c) 10 iterations (d) 50 iterations

(e) 100 iterations (f) 1000 iterations (g) 5000 iterations (h) 10000 iterations

Figure 43: Class distribution and clusters across different iterations

(until iteration 100), we observe a high error. At the 1000 iteration mark, we see a huge improvement. For 5000 and 10000 iterations, the error is comparable to that of 1000 iterations. If anything, the quantization error at 10000 iterations seems to be slightly bigger than for 1000 and 5000 iterations.

Finally, we analyzed the topographic error (Figure 45). Until 100 iterations, all units that are matched to data-points show some topographic error. Starting with 1000 iterations, the situation improves, and we start to see some non-empty units without error. For 5000 iterations and 10000 iterations, we can only see a slight improvement compared to 1000 iterations.

As soon as the data-points start to form groups in the map (iteration 1000), we can see an improvement in cluster structure, quantization error, and topographic error. This improvement is especially noticeable when looking at the quantization error, which indicates that it is useful to analyze whether or not a map is stable.

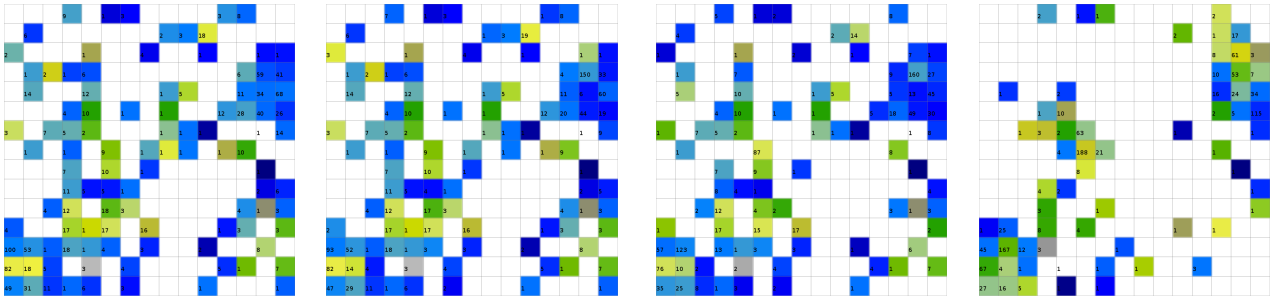
8) Optimal SOM analysis

The parameters for the SOM used in this last experiment are outlined in Table 6. The map size was chosen as 20×20, since using a larger map in the third experiment had some benefits, such as a lower topographic error. This gives us a unit to input vector ratio of approximately 1:3. We decided not to go for a 1:1 ratio, as the quantization error increased for the 30x30 map. Furthermore, we decided to increase the neighborhood radius (sigma) as it proved to decrease topographic error (as observed in experiment 4). As for the learning rate, we decided to go for a balance between the base case and the large learning rate from experiment 5, as they showed similar results. Finally, in experiment 7, we could see that a higher number of iterations is generally beneficial for stabilization of the clusters. For this reason, we decided to increase the number of iterations from 5400 to 20000. The only expected drawback from this is an increase in running time.

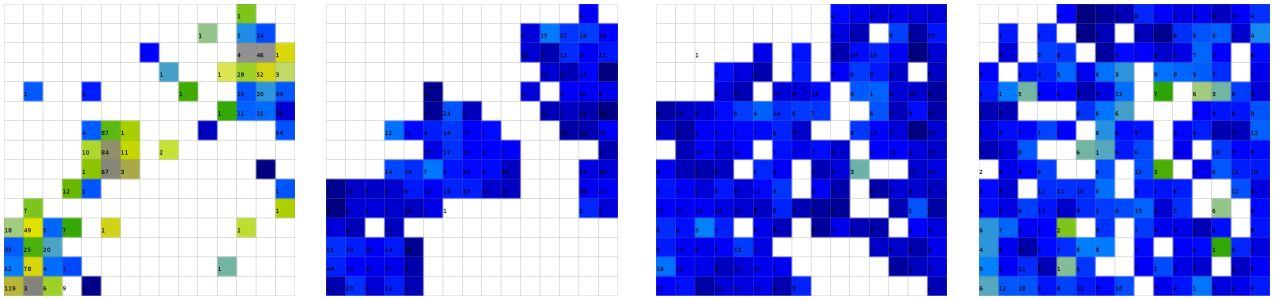
Table 6: Parameters for optimal SOM

Parameter	Value
Random seed	42
x-size	20
y-size	20
Learning rate	0.9
Sigma	4
Num iterations	20000

When looking at the class distribution of the SOM (Figure 46), there is not a big improvement compared to the basic SOM (Figure 1): Some classes are grouped together very clearly, while others are more spread out over the map. For instance, the yellow (cSCs) and green (cSCm) classes appear very close, which makes sense given their similarity

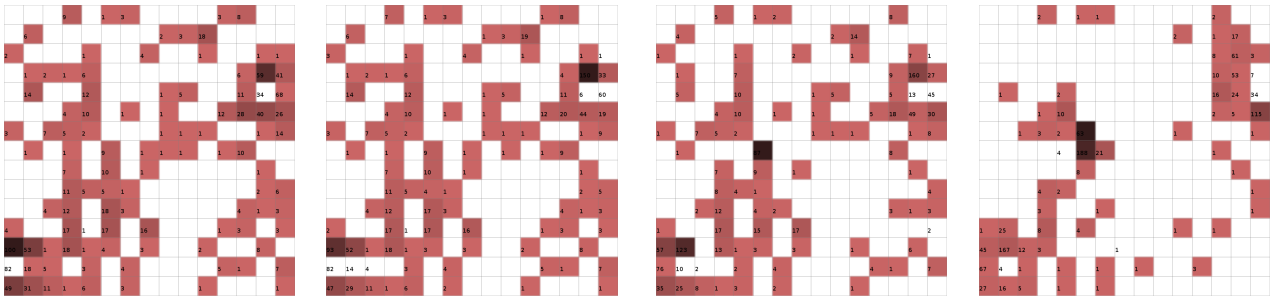


(a) 2 iterations (b) 5 iterations (c) 10 iterations (d) 50 iterations

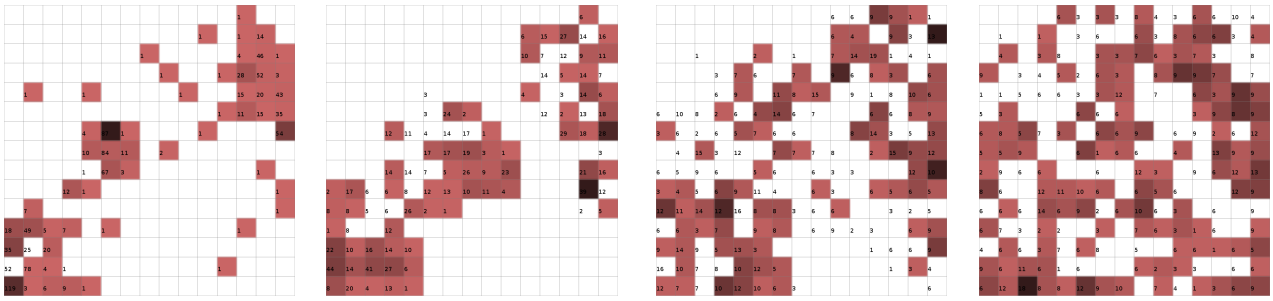


(e) 100 iterations (f) 1000 iterations (g) 5000 iterations (h) 10000 iterations

Figure 44: Quantization error across different iterations



(a) 2 iterations (b) 5 iterations (c) 10 iterations (d) 50 iterations



(e) 100 iterations (f) 1000 iterations (g) 5000 iterations (h) 10000 iterations

Figure 45: Topographic error across different iterations

in semantics. The same can be said for the class distribution among clusters (Figure 47): Some clusters consist of mainly one class (e.g. the mainly grey cluster in the center), while some clusters are very diverse (cluster at the top). However, a general observation is that the clusters distribution does not follow the class distribution. We can also see that the clusters are distributed and spaced fairly evenly across the map, and contain more or less the same number of data points. A difference to the basic map of experiment 1 is the absence of large empty areas.

Despite this, we do not observe any kind of hierarchical cluster relations.

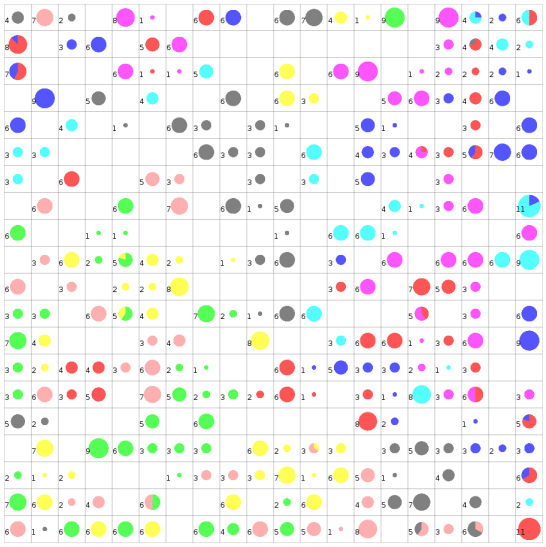


Figure 46: Class distribution of optimal SOM

As in experiment 1, we also grouping the classes by only one of their defining characteristics at a time (figures 49, 49, and 50). In this way, a clearer distribution emerges. The clearest distinction can be found when grouping by behavior, where the red and blue classes are very separated in the map, just as in experiment 1.

In Figure 51, we can see that the center of the map is less dense than the border regions. This could be an indication of border effect. We can also see that dense regions have a similar number of data-points in them compared to less dense regions, indicating a magnification factor.

The SOM shows an overall low quantization error (Figure 52), and a low topographic error (Figure 53). This indicates that generally, the units are close to their matched input vectors in the input space, and that the topology of the input data is conserved. The biggest topographic errors can be seen on the border of the map, especially on the bottom. To confirm this hypothesis, we decided to have a look at another

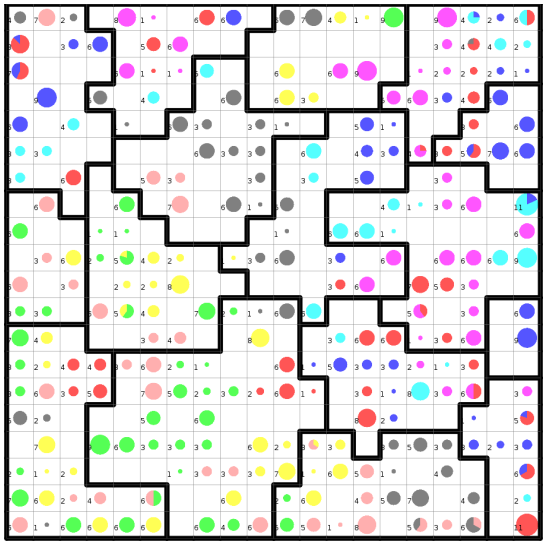


Figure 47: Clusters and class distribution of optimal SOM

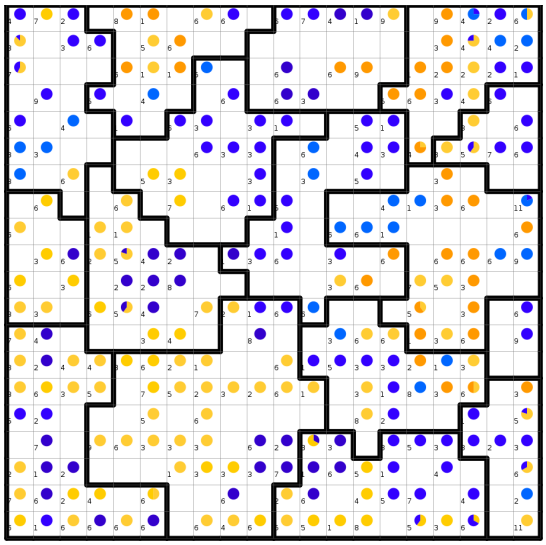


Figure 48: Treatment distribution and clusters of optimal SOM (blue = saline, orange = memantine)

visualization for topology violations: the knn-Neighborhood-Graph (Figure 54). We can see that there are indeed many long arcs going from units at one border to units at another border (e.g. from the left top corner to the bottom right corner). In contrast, there are more short arcs in the middle of the map, which seems to confirm our hypothesis.

D) SUMMARY OF FINDINGS

1) General observations

The first thing that we learned during this project is that it is not necessary to understand the dataset to run it through

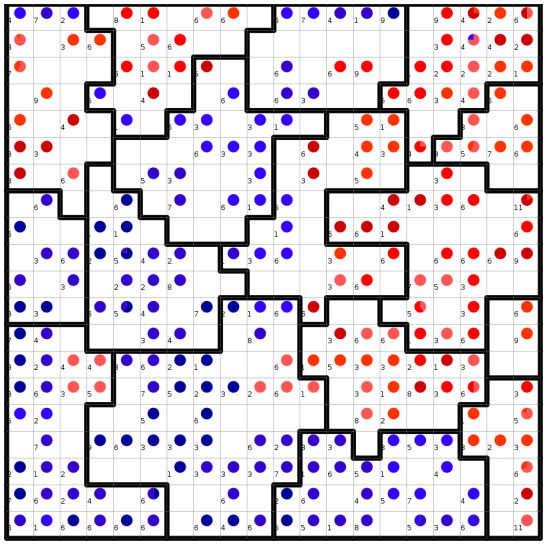


Figure 49: Behavior distribution and clusters of optimal SOM (blue = Shock context, red = Context shock)

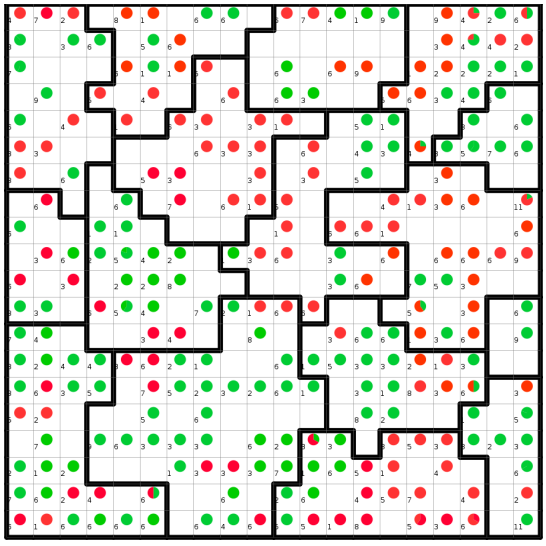


Figure 50: Genotype distribution and clusters of optimal SOM (red = trisomic, green = control)

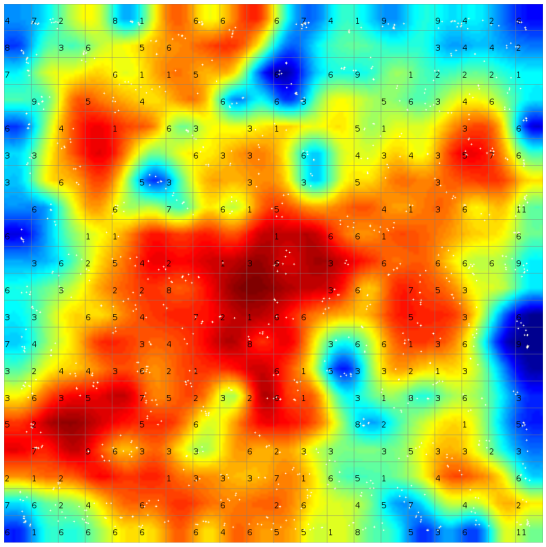


Figure 51: Interpolated P-matrix overlayed with sky metaphor

Figure 52: Quantization Error of the optimal SOM

a SOM, but knowing it could provide very useful insights to understand the results. At the same time, knowing the dataset could provide a starting bias that could lead to actively pursue a specific result. It is anyway clear that a basic understanding of the dataset is beneficial when interpreting a SOM.

A really interesting observation about the clusters and the class distribution is that they rarely overlap. However, when aggregating the classes with respect to a specific feature (such as behavior), they overlap much better with the clusters and it is very easy to find patterns and to make observations.

For example, Behaviour seems to be the feature that relates better with clusters, with almost every cluster being dominated by one behaviour type. The Genotype feature is the second best, while the Treatment feature seems to be the most spread out. This could mean that on trisomic mice the Shock-Context approach is more effective that the memantine treatment. It is also important to notice that we have very limited information on this experiment and this is our interpretation of the data without any technical knowledge.

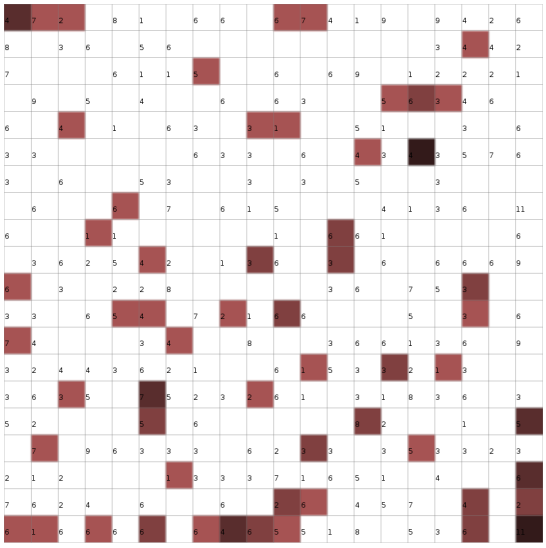


Figure 53: Topographic error of the optimal SOM

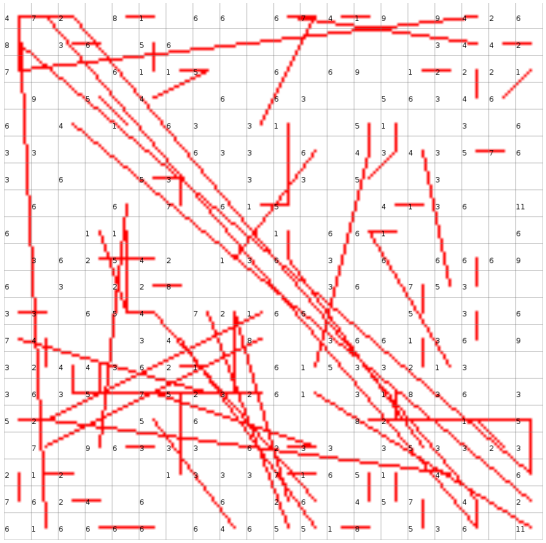


Figure 54: Neighborhood Graph (kNN) of the optimal SOM (k=1)

2) Preprocessing

As expected, preprocessing proved to be crucial when dealing with SOMs. Handling missing data in the wrong way or not dealing with outliers could pollute the results of an otherwise promising dataset and obfuscate the final results. In particular it is very important to perform scaling correctly, since an incorrect scaling could have a bad impact on the quality of the SOM, as seen in experiment 6.

3) Impact of parameters

The number of iterations proved to be a very important parameter for training a SOM. The quality of the SOM improves significantly with a higher iteration number, until it eventually reaches a plateau. The drawback of a high number of iterations is that the running time significantly increases. In our case this was not a problem, since the dataset was relatively small. With a bigger dataset this would be of bigger concern.

Another crucial factor is the dimension of the map. If the map is too small, then no observations are possible, since the units are overloaded with input vectors. With a larger map however, the training time increases significantly. As we know from the lecture, a good map size is strongly dependent on the dataset. The size of the map influences the impact of the parameter sigma. The same sigma value could be too big or too small depending on the size of the map.

Another observation can be done on the random nature of the SOM training. Changing the random seed produces maps that appear very different on a first look, but the core characteristics like the class and cluster properties remain similar. The mean quantization error and the topographic error are still comparable across maps with a different random seed. This consistency is very important because it allows us not to worry about whether the map was trained with a good seed or not (the presence of a "luck factor" is generally a bad sign).

Concerning the neighborhood radius, we observed that a too low value leads to improper magnification of the dense areas of the map, while too high values led to very high border effect (since all units are attracted toward the center). Besides, discussion with other groups indicated that the optimal value for sigma varies between different datasets. In our case the optimal value seemed to be 4, but given an arbitrary dataset, one should experiment with different values for finding a sigma that works well.

Finally, the learning rate should be chosen in coordination with the number of training iterations, such that the training reaches an acceptable state. In our case, we observed that a learning rate closer to 1.0 was beneficial, coupled with a high number of iterations. However this might be relative to our specific data and should not be taken as a general rule.

4) Visualizations

Regarding visualization, we found that the p-matrix used in conjunction with the hit-histogram is useful to observe the magnification factor in our SOMs.

To judge the quality of the SOMs, we mainly used the mean-quantization-error (as the quantization error visualizer did not work in the JavaSOMToolbox), as well as the topographic error. To further observe the extent of topology

violations the neighbourhood graph proved useful. However, when using a large k (with the kNN method), neighborhood graphs tend to become hard to read.

Another visualization that we used extensively is the pie-chart to visualize class distribution. The thematic class map could have been also used for this purpose. However, after initial experiments, we concluded that it did not provide a clear advantage over the pie chart visualization. Therefore, we did not include it in the report.

A APPENDIX

In the appendix we provide the tables and the images too big to be included in the main body of the report.

Table 7: Details about Attributes: Part 1

Attribute name	mean	std	outliers
DYRK1A_N	0.4261	0.249	13
ITSN1_N	0.6168	0.2512	10
BDNF_N	0.3197	0.0505	0
NR1_N	2.2923	0.359	0
NR2A_N	3.8347	0.9476	0
pAKT_N	0.234	0.0443	4
pBRAF_N	0.1828	0.0325	3
pCAMKII_N	3.5287	1.3025	0
pCREB_N	0.2134	0.0364	3
pELK_N	1.4262	0.4685	13
pERK_N	0.5458	0.3447	14
pJNK_N	0.3141	0.0531	0
PKCA_N	0.3185	0.0533	0
pMEK_N	0.2757	0.0479	0
pNR1_N	0.825	0.1188	0
pNR2A_N	0.7264	0.188	0
pNR2B_N	1.5591	0.2757	0
pPKCAB_N	1.5226	0.4838	0
pRSK_N	0.4431	0.0667	0
AKT_N	0.6818	0.1275	0
BRAF_N	0.3789	0.2161	15
CAMKII_N	0.3638	0.0529	0
CREB_N	0.1814	0.0319	3
ELK_N	1.1648	0.3394	0
ERK_N	2.4691	0.6602	0
GSK3B_N	1.1708	0.2467	0
JNK_N	0.2424	0.037	3
MEK_N	0.2749	0.0489	4
TRKA_N	0.6927	0.1209	0
RSK_N	0.1694	0.0338	3

Table 8: Details about Attributes: Part 2

Attribute name	mean	std	outliers
APP_N	0.4051	0.0614	0
Bcatenin_N	2.1218	0.4719	0
SOD1_N	0.5426	0.28	0
MTOR_N	0.4527	0.0655	0
P38_N	0.4156	0.0893	0
pMTOR_N	0.7584	0.1228	0
DSCR1_N	0.585	0.1005	0
AMPKA_N	0.3688	0.063	0
NR2B_N	0.5652	0.0881	0
pNUMB_N	0.3576	0.0635	0
RAPTOR_N	0.3164	0.0552	0
TIAM1_N	0.4189	0.0674	0
pP70S6_N	0.3948	0.156	0
NUMB_N	0.1811	0.0293	0
P70S6_N	0.9431	0.1728	0
pGSK3B_N	0.1612	0.0193	0
pPKCG_N	1.7066	0.5782	0
CDK5_N	0.2924	0.0374	1
S6_N	0.4292	0.1374	0
ADARB1_N	1.1974	0.3616	0
AcetylH3K9_N	0.2165	0.1852	1
RRP1_N	0.1666	0.0319	6
BAX_N	0.1793	0.0188	0
ARC_N	0.1215	0.0143	0
ERBB4_N	0.1565	0.0151	0
nNOS_N	0.1813	0.0249	0
Tau_N	0.2105	0.069	0
GFAP_N	0.1209	0.0132	1
GluR3_N	0.2219	0.0349	0
GluR4_N	0.1266	0.0269	2
IL1B_N	0.5273	0.082	0
P3525_N	0.2913	0.03	0
pCASP9_N	1.5483	0.248	0
PSD95_N	2.2352	0.2543	0
SNCA_N	0.1598	0.0241	0
Ubiquitin_N	1.2393	0.1735	0
pGSK3B_Tyr216_N	0.8488	0.0943	0
SHH_N	0.2267	0.029	0
BAD_N	0.2624	0.2145	0
BCL2_N	0.2782	0.2422	0
pS6_N	0.1215	0.0143	0
pCFOS_N	0.1686	0.1403	0
SYP_N	0.4461	0.0664	0
H3AcK18_N	0.2559	0.2034	0
EGR1_N	0.2777	0.1977	0
H3MeK4_N	0.3307	0.2256	0
CaNA_N	1.3378	0.317	0

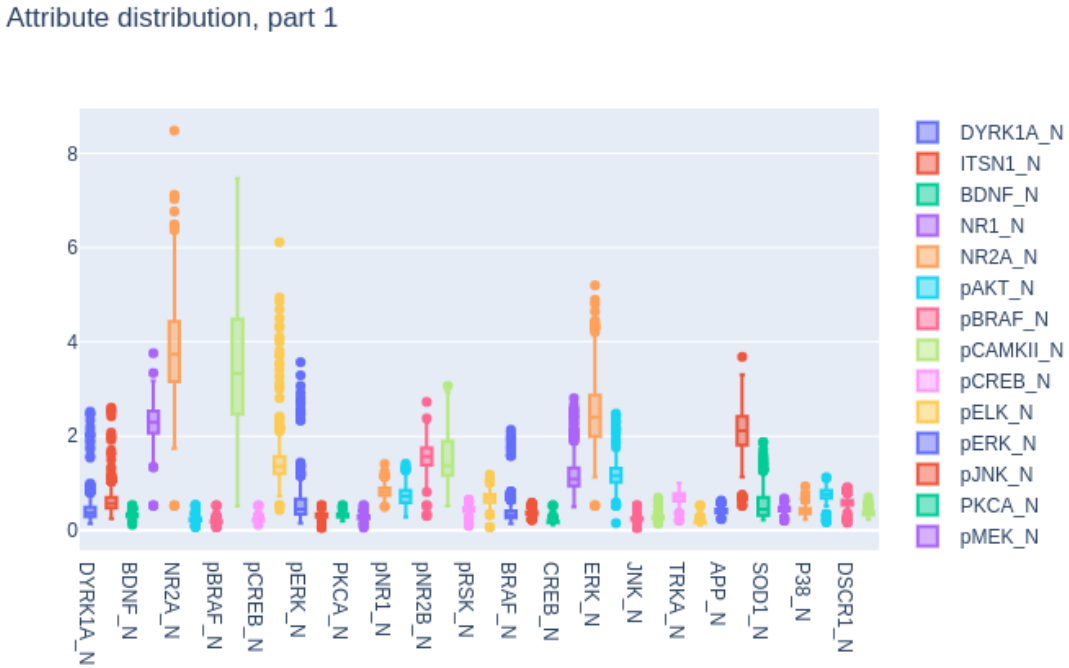


Figure 55: Attribute distribution, part 1

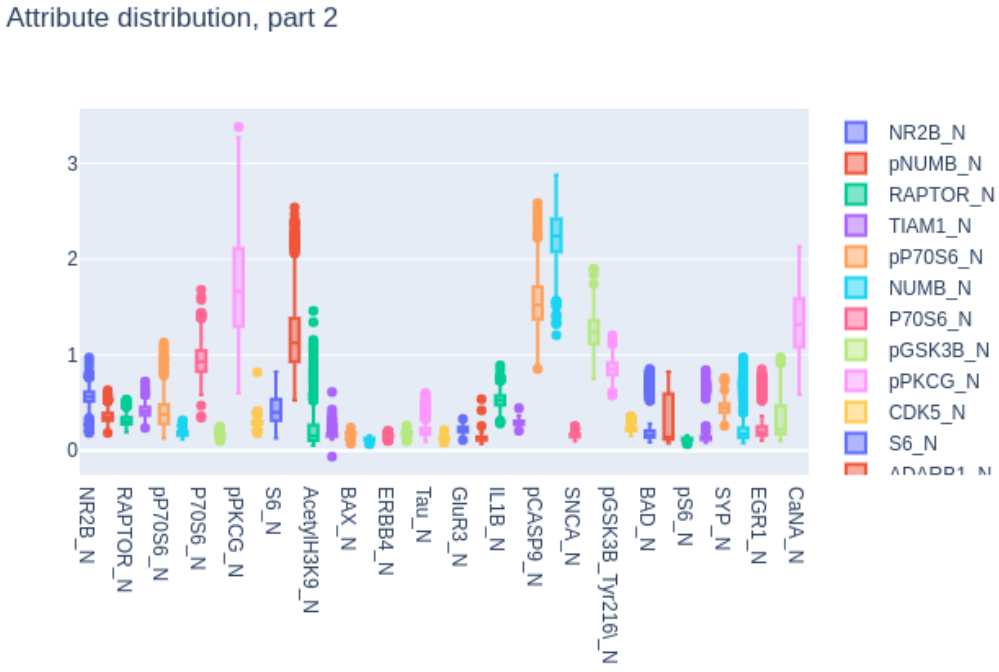


Figure 56: Attribute distribution, part 2