Clustering biological data with SOMs: on topology preservation in non-linear dimensional reduction

Diego H Milone[a], Georgina Stegmayer[a,b], Laura Kamenetzky[c], Mariana López[c], Fernando Carrari[c]

[a]*Research Center for Signals, Systems and Computational Intelligence, FICH-UNL, CONICET, Ciudad Universitaria UNL, Santa Fe, (3000), Argentina, d.milone@ieee.org*
[b]*Centro de Investigación en Ingeniería en Sistemas de Información, CONICET, Lavaise 610, Santa Fe, (3000), Argentina, gstegmayer@santafe-conicet.gov.ar*
[c]*Instituto de Biotecnología, Instituto Nacional de Tecnología Agrícola (IB-INTA), CONICET, PO Box 25, B1712WAA Castelar, Argentina (partner group of the Max Planck Institute for Molecular Plant Physiology, Potsdam-Golm, Germany)*

Abstract

Dimensional reduction is a widely used technique for exploratory analysis of large volume of data. In biological datasets, each object is described by a large number of variables (or dimensions) and it is crucial to perform their analyses in a smaller space, to extract useful information. Kohonen self-organizing maps (SOMs) have been recently proposed in systems biology as a useful tool for exploratory analysis, data integration and discovery of new relationships in -omics datasets. SOMs have been traditionally used for clustering in several data mining problems, mainly due to their ability to preserve input data topology and reduce a high dimensional input space into a 2-D map. In spite of this, the above-mentioned dimensional reduction can lead to counterintuitive results. Sometimes, maps having almost the same size, trained on the same dataset, and with identical learning algorithms and parameters, may find different clusters. However, one would expect that small changes in map sizes or another training condition would not result in an abrupt different location of any of the grouped patterns. The aim of this work is to analyze and explain this issue through a real case study involving transcriptomic and metabolomic data, since it might have an important impact when interpreting clustering results over a biological dataset.

Keywords: clustering, bioinformatics, dimensional reduction, topology preservation.

1. Introduction

Among clustering methods used in biological data mining today [1, 2, 3], self-organizing maps (SOMs) [4, 5] provide the dual property of finding clusters of patterns and relate them by similarity, preserving input data topology as well as reducing a high dimensional input space into a 2-D map [6, 7, 8]. This is possible because SOMs perform a non-linear projection of the input data onto the elements of a regular array, usually of low dimension [9]. The main characteristics of the projection is the preservation of neighborhood relations in the output space, which makes possible to see more clearly the structure hidden in the high-dimensional data, such as clusters [10, 11]. This capability of topology-preserving mapping [12] is one of its main advantages over other dimension reduction methods, although one of its shortcomings is the difficulty for non-expert users to interpret the SOM results [11].

SOMs have been applied with success to the analysis of expression profiles in several biological studies [13] and it is one of the first computational intelligence techniques that has been used for this kind of analysis [14, 15, 16, 17, 18, 19]. In fact, SOMs have been recently proposed in systems biology as a useful tool for exploratory analysis, integration and mining of different biological data [20, 21]. Biological data analysis usually consists of two phases: exploratory data inspection to generate hypothesis that have to be verified afterwards [22].

If a SOM is viewed from the perspective of its properties for dimension reduction, the non-linearities introduced in the mapping must be considered. These may lead to certain results whose interpretation might not be trivial. Thus, this phenomenon might have an important impact when interpreting clustering results over biological datasets. In order to guarantee the correct analysis of input data, there are different measures to quantify the goodness of a map in terms of the accuracy of the maps in preserving the topology (neighborhood relations). A widely used measure is the topographic error that determines how continuous the mapping from the input space to the map grid is [23]. However, it has been studied that it seems to have a tendency to depreciate rectangular maps [10] and it sometimes fails to effectively recognize mismatches between the input space and a map [12].

In any case, a continuous mapping and a good resolution are desired. A mapping is considered continuous if the data vectors that are very close in the input manifold are mapped to the same or adjacent neural units. A good map resolution is obtained when vectors that are distant in input space are not mapped closed in output space. However, the topology cannot be perfectly preserved and there is always a trade-off between the continuity and the resolution, which results in discontinuities in the mapping [24]. This behavior was named automatic selection of feature dimensions by Kohonen himself [25]. The aim of this work is to analyze and explain this issue through a real case study involving transcriptomic and metabolomic data, since it might have an important impact when interpreting clustering results over a biological dataset. Section 2 describes the materials and methods used in this study. Section 3 presents the jumping-pattern behavior issue through a real case. In Section 4, a simple artificial example is used for detailed analysis and explanation and, afterwards, a real case study involving transcriptomic and metabolomics data is shown. Finally, the conclusions can be found on Section 5.

2. Materials and methods

The biological dataset used in this study has 2458 values obtained from transcriptional profiles and metabolic accumulation of *S. lycopersicum* x *S. pennellii* introgression lines (ILs) [26]. The ILs harbor, in certain chromosome segments, introgressed portions of the wild Solanum species (*Solanum pennelli*). The use of ILs allows the study and creation of new varieties of such species by introducing exotic traits and thus constitutes a useful tool in crop domestication and breeding [27, 28]. The interest in comparing the cultivated *Solanum lycopersicum* with the different ILs lies on the fact that some wild *Solanum lycopersicum* fruits have proven to be the source of several specific agronomic traits, which could be used for the improvement of *Solanum lycopersicum* commercial lines. After log-transforming the green/red ratio values over the entire dataset, genes whose expression did not change significantly were discarded from further analysis. As a result of the pre-processing and selection steps, 70 metabolites and 1159 genes were selected [29].

Let us suppose having a SOM map trained with the standard batch-

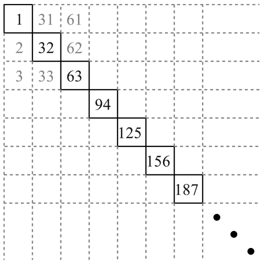


Figure 1: Neurons of interest in a SOM map.

training algorithm [4] provided by the somtoolbox made available by the Helsinky University of Technology[1]. The map has been linearly initialized by the two eigenvectors with the greatest eigenvalues computed from training data [30]. The maps were trained with the default training parameters and a Gaussian neighborhood function. Each neuron of the map is given a consecutive number that starts with 1 in the upper left corner, numbering continues consecutively in columns, until it finishes in 900 in the right down corner of the map (see an example in Figure 1).

3. Results: the jumping-pattern behavior issue

If a 30x30 SOM map is trained for 100 epochs with the previously mentioned dataset, the amino acid *alanine* is clustered in neuron 1. If a SOM of 31x31 neurons is used for the same dataset and training conditions, after 100 epochs *alanine* is located in neuron 445, a cluster far away from neuron 1. Nevertheless, all other transcripts and metabolites in neuron 1 of the 30x30 map are consistently clustered in neuron 1 of the new 31x31 map.

This fact can be quite confusing. If an input vector is clustered in a

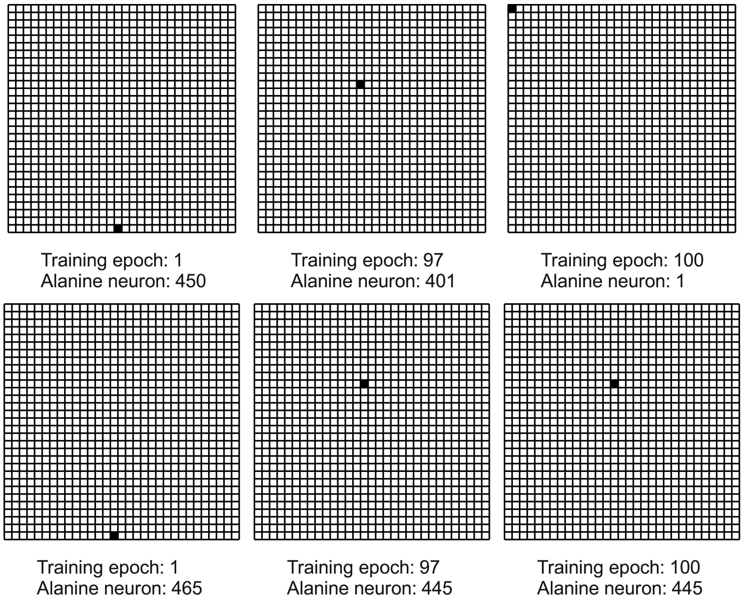
[1] http://www.cis.hut.fi/somtoolbox/

neuron, it is assumed that the variation pattern of this input vector is more similar to the other elements within this neuron than to any others in the map. Furthermore, given that neighbourhood neurons are similar among them and due to the topology preservation property, it should be expected that if there is a change in a given data point allocation, it would be to any neighbourhood neuron, but never to a neuron located at almost half-of the map of distance.

The images in Figure 2.a) show results after training a 30x30 SOM map for the amino acid *alanine*[2]. A black point is depicted in the position where this metabolite is located at different training epochs. The images show that this pattern, near the end of the training stage (at epoch 98), *jumps* from neuron 401 towards neuron 2. That is to say, on epoch 97 the neuron centroid more similar to the *alanine* is the 401, and in the epoch 98 it is the neuron centroid 2. Finally, at epoch 100, *alanine* ends located at neuron 1. According to the Euclidean distances, this jumping is correct. However, neurons 2 and 401 are quite distant in the map and this large jump was unexpected.

On the other hand, Figure 2.b) shows results from a 31x31 SOM, trained exactly with the same training parameters (and with the same dataset) as the previous map. When both trained maps are compared, a clear inconsistency in the clusters is found. As explained above, if the 30x30 map is taken as the reference map, all the data patterns that were grouped in neuron 1 in this map are consistent with the data points found in the neuron 1 of the 31x31

[2] Video animations of the figures in this paper can be found on: http://sourcesinc.sourceforge.net/ostm/



Training epoch: 1
Alanine neuron: 450

Training epoch: 97
Alanine neuron: 401

Training epoch: 100
Alanine neuron: 1

Training epoch: 1
Alanine neuron: 465

Training epoch: 97
Alanine neuron: 445

Training epoch: 100
Alanine neuron: 445

Figure 2: The pattern metabolite *alanine* (black dot) location at different training epochs in a) a 30x30 SOM map; b) a 31x31 SOM map.

map. The exception is the *alanine*, which has been located in a neuron that is more than 15 neurons away in the 31x31 map (neuron 445).

A natural question that may arise here is: what happens if the maps training is continued for more epochs? The result is that after training 50 additional epochs, both maps (30x30 and 31x31) are consistent, but then they become inconsistent again when epoch 200 is reached.

After this initial analysis we can conclude that, given the difference on map sizes, the data patterns move at a different speed, and by cutting the training at a given epoch number, there is no visible reason why they should all stay in the same cluster in both maps. However, there is still an issue to address: how is it possible that a data point could jump 15 neurons apart in only one training epoch? Stated differently: how is it possible that there are some patterns which are similar to a group of neurons located in a position of the map, but they can also be similar to another neuron far away of this position?

4. Discussion: a 3-D perspective of a 21-D problem

Let us suppose now that we have a 30x30 SOM map trained with a hypothetical experiment involving genes, composed by only three gene expression levels on a microarray. Therefore, in this simple example, each data point has 3 dimensions and we will consider only the following neurons on the map: 1, 32, 63, 94, 125, 156 and 187 (see Figure 1).

Suppose we have a given gene $\mathbf{x}$, whose corresponding expression values are $[4, 2, 3]$. To further simplify the analysis, only the following neuron centroids over all the map will be considered: $\mathbf{w}_1 = [2, 2, 3]$; $\mathbf{w}_{32} = [2, 2, 2]$; $\mathbf{w}_{63} = [2, 3, 2]$; $\mathbf{w}_{94} = [2, 4, 2]$; $\mathbf{w}_{125} = [2, 4, 3]$; $\mathbf{w}_{156} = [3, 4, 3]$ and $\mathbf{w}_{187} = [4, 4, 3]$. Figure 3 shows these centroids and their location in a 30x30 SOM model, as well as the centroids values and their Manhattan distance[3] to the data pattern $\mathbf{x}$, indicated as $\mathrm{m}(\mathbf{x}, \mathbf{w}_\mathrm{i})$. This distance is calculated here as the sum of the absolute error between $\mathbf{x}$ and $\mathbf{w}_i$ on each dimension[4]. It should be noted that the distance between each analyzed centroid (from 1 to 187) in the diagonal line is always 1.

At a certain point during training, $\mathbf{x}$ is assigned to the neuron centroid

[3] For simplicity, the Manhattan distance is used in this example. The considerations for Manhattan distance hold for Euclidean distance as well.

[4] For example: the distance between $\mathbf{x} = [4, 2, 3]$ and $\mathbf{w}_{32} = [2, 2, 2]$ is calculated as $\mathrm{m}(\mathbf{x}, \mathbf{w}_{32}) = |4 - 2| + |2 - 2| + |3 - 2| = 2 + 0 + 1 = 3$.

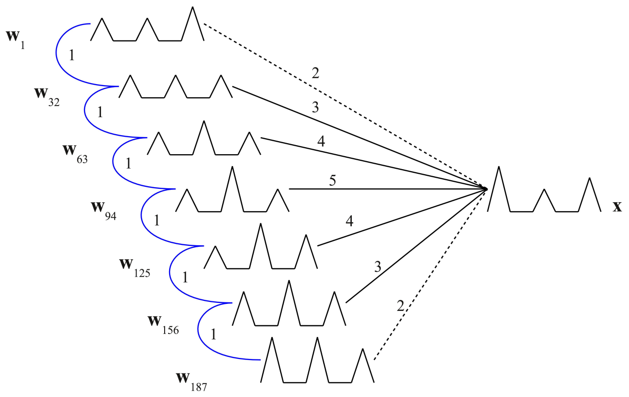


Figure 3: Simplified 3-D example: centroids location and values of the neurons of interest in a 30x30 SOM map and their Manhattan distance to a data pattern $\mathbf{x}$.

$\mathbf{w}_1$. Then, in the next training iteration, $\mathbf{x}$ *jumps* from neuron 1 to neuron 187, meaning that the shape of the pattern is more similar to the shape of $\mathbf{w}_{187}$ at this training step. As can be seen from Figure 2 these centroids are quite distant in the map. The fact that a data pattern is closer to distant centroids than to some of the intermediate neurons goes against intuition. In terms of neighbourhood neurons, this jumping implies a Von Neumann neighbourhood with radius 12, which is clearly a large distance in a 30x30 map.

However, Figure 3 shows that $\mathrm{m}(\mathbf{x}, \mathbf{w}_1) = 2$ and $\mathrm{m}(\mathbf{x}, \mathbf{w}_{187}) = 2$, while the distances between $\mathbf{x}$ and all the other intermediate centroids are greater than 2. For instance, if $\mathbf{x}$ is assigned to $\mathbf{w}_1$ and after a training epoch the centroid $\mathbf{w}_{187}$ assumes the value $[4, 3.9, 3]$, now $\mathrm{m}(\mathbf{x}, \mathbf{w}_{187}) = 1.9$ and therefore, $\mathbf{x}$ must jump to the $\mathbf{w}_{187}$ centroid. After some more training, it may happen that $\mathbf{w}_1$ takes the values $[2.2, 2, 3]$. In this case, now $\mathrm{m}(\mathbf{x}, \mathbf{w}_1) = 1.8$ and again, the data point $\mathbf{x}$ must jump-back to neuron 1, without being assigned, previously, to any of the intermediate centroids $\mathbf{w}_{156}$, $\mathbf{w}_{125}$, $\mathbf{w}_{94}$, $\mathbf{w}_{63}$ nor $\mathbf{w}_{32}$. This means that the data point is more different to the intermediate centroids and cannot be assigned to any of them, even when all the neighbour neurons are very similar among them. In other words, in this projected space, the shortest path between the neurons does not seem to be a straight line, since the distances are actually measured on the original space.

This analysis has been performed taking into account the 2-D dimensions of the SOM. If we would look at the problem in its original 3-D dimensional space, we would build a graph like the one presented in Figure 4. It shows all the analyzed centroids in blue and the data point $\mathbf{x}$ indicated with a red circle. The distance between each point can be seen clearly here. Now is possible to see how the data point $\mathbf{x}$ is as close to $\mathbf{w}_1$ as to $\mathbf{w}_{187}$, and it is more distant to the rest of the neurons. This means that, what can be seen as an abrupt jumping in a 2-D map, it is not counter-intuitive in the original dimensionality of the problem, because in a 3-D space, those points are clearly, the closest ones.

The jumping-pattern issue will now be seen in the dimensions of a real problem (the tomato dataset presented in Material and Methods). Figure 5 shows the detail of the jumping amino acid *alanine* and the centroids of the neurons 401 and 2 for a 30x30 SOM map. In this case, the distance between points has been calculated using the classical Euclidean distance. The left of the figure shows the results after 90 training epochs with a 30x30 SOM map. At this stage of the training process, the Euclidean distance $\mathrm{d}(\mathbf{x}, \mathbf{w}_{401}) = 0.93$, while $\mathrm{d}(\mathbf{x}, \mathbf{w}_2) = 0.95$, therefore $\mathbf{x}$ is clustered in neuron 401 (indicated with an arrowhead). The right part of the figure shows that,

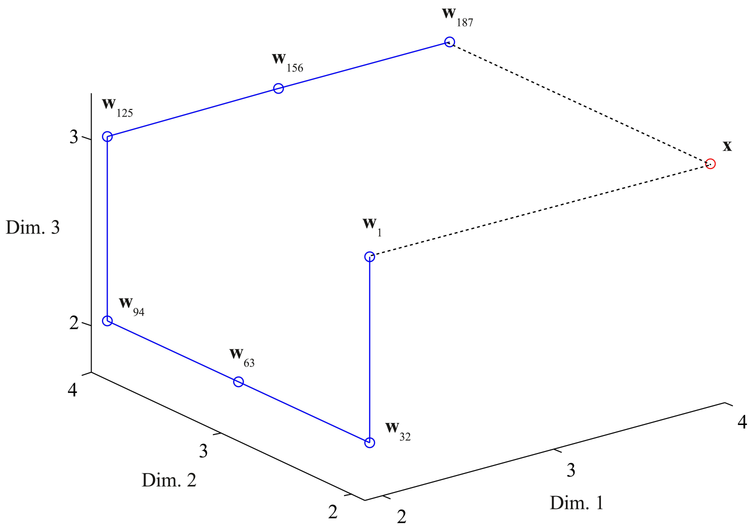


Figure 4: Centroids of a 30x30 SOM map in a 3-D problem.

after 98 training epochs, the distance between the pattern $\mathbf{x}$ and $\mathbf{w}_2$ is now the shortest, and therefore $\mathbf{x}$ has to be assigned to the neuron 2 (arrow indication).

Each of the curves in the real case has 21 dimensions and it is very hard to analyze which are those that make the data point more similar to one centroid over another one. In any case, the assignments of the pattern to each centroid on each epoch are correct and coherent. Analogously to the 3-D example, only these two neuron centroids (at training epoch 98) are more similar to the data point $\mathbf{x}$ than any other neuron in the whole map, due to the fact that the distances are measured in the original 21 dimensions of the problem.

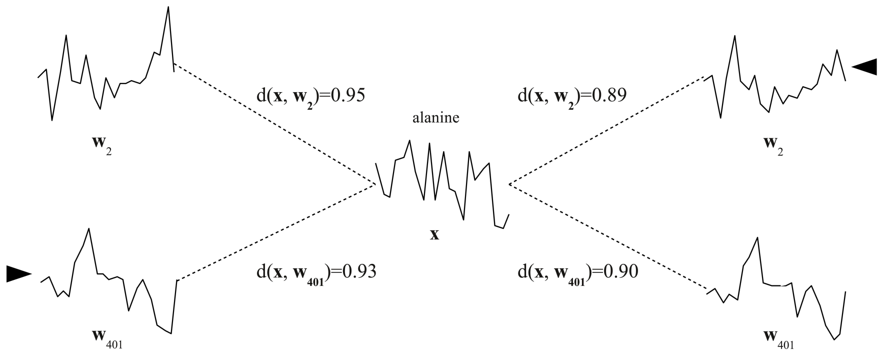


Figure 5: Centroids of a 30x30 SOM map in a real 21-D problem.

5. Conclusions

To conclude, we can say that, although the jumping-pattern behavior in a SOM map may seem hard to understand, it is possible that some data points, similar among them, are grouped in distant neurons when considering two maps with slightly different sizes or variants in the training process. This is not the general case, however. In a 900 neurons map that clusters more than 2400 data points, there may be around 10 patterns in total that can eventually jump between distant neurons in the map.

In general, this behavior can be considered as inherent to any method for non-linear dimension reduction. Particularly in this case, it is well-known that there is some loss of information when the original problem is reduced and projected into 2-D. This is especially true when the original space has a high dimensionality. There might be some cases that cannot be projected perfectly and therefore, while they are correct in the original dimension, they may seem incorrect in the reduced one. However, whilst some information might be lost due to the dimension reduction technique, from the original dimensions of the problem it might be impossible to visualize, analyze or extract any useful information.

6. Acknowledgements

This work was supported by the National Scientific and Technical Research Council [PIP 1122008, PIP 2072008, PIP 1142009]; the National Agency for the Promotion of Science and Technology [PICT 2008 00100, PICT 2008 00140, PAE 37122, PICT 2009 00152] and the National Institute for Agricultural Technology [PE 243542] of Argentina.

References

[1] G. Fogel, D. Corne, Y. Pan, Computational Intelligence in Bioinformatics, Wiley-IEEE Press, 2007.

[2] H. Wang, Z. Wang, X. Li, B. Gong, L. Feng, Y. Zhou, A robust approach based on weibull distribution for clustering gene expression data., Algorithms for Molecular Biology 6 (1) (2011) 14.

[3] J. A. Castellanos-Garzon, C. A. Garcia, P. Novais, F. Dias, A visual analytics framework for cluster analysis of dna microarray data, Expert Systems with Applications 40 (2) (2013) 758 – 774.

[4] T. Kohonen, Self-organized formation of topologically correct feature maps, Biological Cybernetics 43 (1982) 59–69.

[5] T. Kohonen, M. R. Schroeder, T. S. Huang, Self-Organizing Maps, Springer-Verlag New York, Inc., 2005.

[6] E. Bonabeau, F. Henaux, Self-organizing maps for drawing large graphs, Information Processing Letters 67 (4) (1998) 177–184.

[7] N. Chen, B. Ribeiro, A. Vieira, A. Chen, Clustering and visualization of bankruptcy trajectory using self-organizing map, Expert Systems with Applications 40 (1) (2013) 385 – 393.

[8] A. Segev, J. Kantola, Identification of trends from patents using self-organizing maps, Expert Systems with Applications 39 (18) (2012) 13235 – 13242.

[9] M. Cottrell, J.-C. Fort, G. Pagès, Theoretical aspects of the som algorithm, Neurocomputing 21 (1-3) (1998) 119–138.

[10] E. A. Uriarte, F. D. Martín, Topology preservation in som, International Journal of Mathematical and Computer Sciences 1 (2005) 19–22.

[11] S.-L. Shieh, I.-E. Liao, A new approach for data clustering and visualization using self-organizing maps, Expert Systems with Applications 39 (15) (2012) 11924 – 11933.

[12] T. Villmann, R. Der, J. M. Herrmann, T. Martinetz, Topology preservation in self-organizing feature maps: exact definition and measurement, IEEE Transactions on Neural Networks 8 (2) (1997) 256–266.

[13] P. Meinicke, T. Lingner, A. Kaever, K. Feussner, C. Gobel, I. Feussner, P. Karlovsky, B. Morgenstern, Metabolite-based clustering and visualization of mass spectrometry data using one-dimensional self-organizing maps., Algorithms for Molecular Biology 3.

[14] R. Xu, D. W. II, Clustering, Wiley and IEEE Press, 2009.

[15] S. Haykin, Neural Networks: A Comprehensive Foundation (3rd Edition), Prentice-Hall, Inc., 2007.

[16] J. Quackenbush, Computational analysis of microarray data, Nat Rev Genet 2 (6) (2001) 418–427.

[17] M. Y. Hirai, M. Yano, D. B. Goodenowe, S. Kanaya, T. Kimura, M. Awazuhara, M. Arita, T. Fujiwara, K. Saito, Integration of transcriptomics and metabolomics for understanding of global responses to nutritional stresses in arabidopsis thaliana, Proceedings of the National Academy of Sciences of the United States of America 101 (2004) 10205–10.

[18] M. Yano, S. Kanaya, M. Altaf-Ul-Amin, K. Kurokawa, M. Y. Hirai, K. Saito, Integrated data mining of transcriptome and metabolome based on BL-SOM, Journal of Computer Aided Chemistry 7 (2006) 125–136.

[19] K. Saito, M. Y. Hirai, K. Yonekura-Sakakibara, Decoding genes with coexpression networks and metabolomics - majority report by precogs, Trends in Plant Science 13 (2008) 36–43.

[20] D. H. Milone, G. Stegmayer, L. Kamenetzky, M. Lopez, J. M. Lee, J. J. Giovannoni, F. Carrari, *omesom: a software for clustering and visualization of transcriptional and metabolite data mined from interspecific crosses of crop plants, BMC Bioinformatics 11 (2010) 438.

[21] G. Stegmayer, D. H. Milone, L. Kamenetzky, M. G. Lopez, F. Carrari, A biologically-inspired validity measure for comparison of clustering methods over metabolic datasets, IEEE ACM Transactions in Computational Biology and Bioinformatics DOI: 10.1109/TCBB.2012.10.

[22] S. Kaski, J. Nikkilä, M. Oja, J. Venna, P. Törönen, E. Castrén, Trustworthiness and metrics in visualizing similarity of gene expression, BMC Bioinformatics 4 (2003) 48.

[23] H.-U. Bauer, K. Pawelzik, T. Geisel, A topographic product for the optimization of self-organizing feature maps, in: NIPS, 1991, pp. 1141–1147.

[24] K. Kiviluoto, Topology preservation in self-organizing maps, in: Int. Conference on Neural Networks, 1996, pp. 294–299.

[25] T. Kohonen, Self-organization and Associative Memory, Vol. 8 of Springer Series in Information Sciences, Springer-Verlag, Berlin Heidelberg, 1984.

[26] F. Carrari, C. Baxter, B. Usadel, E. Urbanczyk-Wochniak, M.-I. Zanor, A. Nunes-Nesi, V. Nikiforova, D. Centero, A. Ratzka, M. Pauly, L. J. Sweetlove, A. R. Fernie, Integrated analysis of metabolite and transcript levels reveals the metabolic shifts that underlie tomato fruit development and highlight regulatory aspects of metabolic network behavior, Plant Physiol. 142 (2006) 1380–1396.

[27] L. Rieseberg, J. Wendel, Introgression and its consequences in plants, Vol. 1, Oxford University Press, 1993.

[28] S. Y. Lippman, Z.B., Z. D., An integrated view of quantitative trait variation using tomato interspecific introgression lines, Current Opinion in Genetics and Development 17 (2007) 1–8.

[29] G. Stegmayer, D. Milone, L. Kamenetzky, M. Lopez, F. Carrari, Neural network model for integration and visualization of introgressed genome and metabolite data, in: IEEE International Joint Conference on Neural Networks (IJCNN), Atlanta, USA, 2009, pp. 2983–2989.

[30] J. Vesanto, J. Himberg, E. Alhoniemi, J. Parkankangas, Som toolbox for matlab 5 (technical report a57)., Neural Networks Research Centre, Helsinki University of Technology, Espoo, Finland (2000) 1–59.