

**Representing Clusters:  
K-Means Clustering, Self-Organizing  
Maps, and Multidimensional Scaling**

Michael W. Trosset<sup>a</sup>

February 20, 2008

**Technical Report 08-03**  
**Department of Statistics**  
**Indiana University**  
**Bloomington, IN**

---

<sup>a</sup>Department of Statistics, Indiana University. [mtrosset@indiana.edu](mailto:mtrosset@indiana.edu)

# Representing Clusters: K-Means Clustering, Self-Organizing Maps, and Multidimensional Scaling

Michael W. Trosset\*

February 20, 2008

## Abstract

It is well-known that self-organizing maps are intimately related to  $k$ -means clustering and to multidimensional scaling. These relations are explored and used to argue that a judicious combination of algorithms for  $k$ -means clustering and multidimensional scaling produces more easily interpreted results than do self-organizing maps, and at comparable computational expense. The proposed methodology is applied to 760 gene expression profiles.

**Key words:** Cluster analysis, unsupervised learning, visualization, parallel coordinates, gene expression, microarray experiments.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>K-Means Clustering</b>	<b>2</b>
<b>3</b>	<b>Self-Organizing Maps</b>	<b>7</b>
<b>4</b>	<b>Multidimensional Scaling</b>	<b>9</b>
<b>5</b>	<b>Example: Gene Clustering</b>	<b>12</b>
<b>6</b>	<b>Discussion</b>	<b>14</b>

---

\*Department of Statistics, Indiana University, Bloomington, IN (e-mail: [mtrosset@indiana.edu](mailto:mtrosset@indiana.edu))

# 1 Introduction

Cluster analysis and multidimensional scaling (MDS) are methodologies that are familiar to most statisticians. Clustering methods partition data into subsets (clusters) that are thought to exhibit internal cohesion and/or external isolation. For data that lie in Euclidean space,  $k$ -means clustering tries to find a partition that minimizes the sums of squared errors about the cluster means, which represent their respective clusters. MDS embeds objects in a Euclidean space of specified dimension in such a way that the resulting set of Euclidean interpoint distances approximates the given set of interobject dissimilarities. When MDS is used to visualize data, the specified dimension will be small, typically two or three. If the interobject dissimilarities are high-dimensional Euclidean distances, then MDS can be used for dimension reduction.

Both  $k$ -means clustering and MDS are intimately related to self-organizing maps (SOMs), a relatively recent but wildly popular methodology for unsupervised learning. A typical SOM displays representatives of similar objects in contiguous locations on a regular grid in two dimensions. SOMs are widely praised by practitioners for their interpretability; ironically, theoreticians have been unable to characterize precisely what SOMs actually do. Because  $k$ -means clustering and MDS are well understood, it is natural to inquire if analyses performed by SOMs might not be better performed by  $k$ -means clustering and MDS. We will argue that a judicious combination of algorithms for  $k$ -means clustering and MDS produces more easily interpreted results than do SOMs, and at comparable computational expense.

Section 2 describes  $k$ -means clustering. MacQueen's (1967) algorithm is derived and contrasted with a special case of SOM methodology that happens to perform  $k$ -means clustering. That this special case of SOM methodology is an algorithm for  $k$ -means clustering is widely known, but the performance of the SOM algorithm for  $k$ -means clustering has not been studied extensively. A simple numerical experiment demonstrates that it should be.

Section 3 explores the implications of imposing a *topological ordering* on the set of representatives. The topological ordering is the signature feature of a SOM, and the mechanism whereby a SOM maps similar objects to contiguous grid locations; however, a simple example demonstrates that the representatives returned by such a SOM may be difficult to interpret.

Section 4 challenges the notion that cluster representatives are best displayed on a regular grid. MDS is used to embed the representatives (not the individual objects) in two-dimensional Euclidean space. A simple example demonstrates that this technique conveys more information than does a regular grid.

Section 5 applies the proposed methodology,  $k$ -means clustering followed by embedding, to a set of 760 gene expression profiles.

Section 6 concludes with a discussion of computational efficiency.

## 2 K-Means Clustering

Given  $x_1, \dots, x_N \in \mathbb{R}^p$  and  $k \in \{1, \dots, N\}$ , we consider the popular clustering criterion

$$W(C_1, \dots, C_k; m_1, \dots, m_k) = \sum_{i=1}^k \sum_{x_j \in C_i} \|x_j - m_i\|^2,$$

where the *clustering*  $\{C_1, \dots, C_k\}$  is a partition of  $\{x_1, \dots, x_N\}$  and  $m_1, \dots, m_k \in \mathbb{R}^p$  are the *representatives* of the *clusters*  $C_1, \dots, C_k$ . The problem of  $k$ -means clustering is the problem of minimizing  $W$ . It is more commonly stated as the problem of minimizing the variable projection

functional

$$\bar{W}(C_1, \dots, C_k) = W(C_1, \dots, C_k; \bar{x}_1, \dots, \bar{x}_k) \leq W(C_1, \dots, C_k; m_1, \dots, m_k), \quad (1)$$

where  $\bar{x}_i = \bar{x}(C_i)$  is the mean of the  $x_j \in C_i$ ; however, for the purpose of this investigation, we prefer to regard minimizing  $\bar{W}$  as one possible approach to minimizing  $W$ . Notice that the arguments of  $\bar{W}$  are discrete.

When  $N$  is large, the problem of  $k$ -means clustering poses both computational and graphical challenges. There exist a number of algorithms that monotonically decrease  $\bar{W}$  and converge to a locally optimal partition, but algorithms that search for global solutions are overwhelmed by several hundred  $x_i$ . When  $N$  is very large (and especially when  $p$  is high), it may also be difficult to display information about the partition, in which case a natural compromise is to display information about the representatives.

The following notation will facilitate subsequent exposition. Given clusters  $C_1, \dots, C_k$ , representatives  $m_1, \dots, m_k$ , and  $x_j \in \{x_1, \dots, x_N\}$ , let  $C(x_j)$  denote the cluster to which  $x_j$  belongs and let  $m(x_j)$  denote the representative of  $C(x_j)$ . Let  $m_*(x_j)$  denote any representative that is nearest  $x_j$ , i.e.,

$$\|x_j - m_*(x_j)\| \leq \|x_j - m_i\|$$

for  $i = 1, \dots, k$ , and let  $C_*(x_j)$  denote the cluster whose representative is  $m_*(x_j)$ .

If  $m_i = \bar{x}_i$  and the clustering is suboptimal, then there is an obvious way to decrease  $\bar{W}$ . Denoting the elements of  $C_1$  by  $\{y_1, \dots, y_{n_1}\}$  and the elements of  $C_2$  by  $\{z_1, \dots, z_{n_2}\}$ , suppose that

$$\|y_{n_1} - m_1\| > \|y_{n_1} - m_2\|,$$

in which case  $y_{n_1}$  is identified with the wrong representative. Let  $C'_1 = \{y_1, \dots, y_{n_1-1}\}$ , let  $C'_2 = \{y_{n_1}, z_1, \dots, z_{n_2}\}$ , and denote the means of  $C'_1$  and  $C'_2$  by

$$m'_1 = \frac{1}{n_1 - 1} \sum_{j=1}^{n_1-1} y_j = \frac{n_1}{n_1 - 1} m_1 - \frac{1}{n_1 - 1} y_{n_1}, \quad (2)$$

$$m'_2 = \frac{1}{n_2 + 1} \left( y_{n_1} + \sum_{j=1}^{n_2} z_j \right) = \frac{n_2}{n_2 + 1} m_2 + \frac{1}{n_2 + 1} y_{n_1}. \quad (3)$$

Then

$$\begin{aligned} \bar{W}(C_1, C_2) &= \sum_{j=1}^{n_1} \|y_j - m_1\|^2 + \sum_{j=1}^{n_2} \|z_j - m_2\|^2 \\ &> \sum_{j=1}^{n_1-1} \|y_j - m_1\|^2 + \|y_{n_1} - m_2\|^2 + \sum_{j=1}^{n_2} \|z_j - m_2\|^2 \\ &\geq \sum_{j=1}^{n_1-1} \|y_j - m_1\|^2 + \|y_{n_1} - m'_2\|^2 + \sum_{j=1}^{n_2} \|z_j - m'_2\|^2 \\ &> \sum_{j=1}^{n_1-1} \|y_j - m'_1\|^2 + \|y_{n_1} - m'_2\|^2 + \sum_{j=1}^{n_2} \|z_j - m'_2\|^2 \\ &= \bar{W}(C'_1, C'_2). \end{aligned}$$

These considerations suggest MacQueen's (1967) algorithm for  $k$ -means clustering, summarized in Figure 1.

```

initialize  $\{C_1, \dots, C_k\}$  and  $m_i = \bar{x}_i$ 
do until termination:
  choose  $x \in \{x_1, \dots, x_N\}$ 
  determine  $C(x)$  and  $C_*(x)$ 
  if  $C_*(x) \neq C(x)$ , then
    move  $x$  from  $C(x)$  to  $C_*(x)$ 
    update the  $m_i$ 

```

Figure 1: MacQueen's (1967) algorithm for  $k$ -means clustering.

Even if we are only interested in the representatives, MacQueen's algorithm requires us to know and update the clustering partition. Especially when  $N$  is large, it is natural to inquire if one can dispense with this bit of bookkeeping. Toward that end, let

$$\tilde{C}_i = \tilde{C}(m_i) = \{x_j \in \{x_1, \dots, x_N\} : m_*(x_j) = m_i\}$$

and consider the variable projection functional

$$\tilde{W}(m_1, \dots, m_k) = W(\tilde{C}_1, \dots, \tilde{C}_k; m_1, \dots, m_k) \leq W(C_1, \dots, C_k; m_1, \dots, m_k).$$

Just as minimizing  $\bar{W}$  is one approach to minimizing  $W$ , so is minimizing  $\tilde{W}$ . Notice that, whereas the arguments of  $\bar{W}$  are discrete, the arguments of  $\tilde{W}$  are continuous. Furthermore, the problem of minimizing  $\tilde{W}$  is *not* a continuous relaxation of the problem of minimizing  $\bar{W}$ ; rather, it is a continuous formulation of the problem of minimizing  $W$ .

It is not obvious how to vary  $m_1, \dots, m_k$  so as to decrease  $\tilde{W}$ . Recall, however, that the inequality in (1) follows from the well-known fact that, for a fixed partition  $C_1, \dots, C_k$ , the objective function

$$f(m_1, \dots, m_k) = W(C_1, \dots, C_k; m_1, \dots, m_k)$$

is minimized by setting each  $m_i$  equal to the mean of the  $x_j \in C_i$ . This fact is easily deduced from the separability of  $f$  into  $k$  convex summands,

$$f_i(m_i) = \sum_{x_j \in C_i} \|x_j - m_i\|^2,$$

and the stationary equations

$$\nabla f_i(m_i) = \sum_{x_j \in C_i} 2(m_i - x_j) = 0$$

for  $i = 1, \dots, k$ . Now, suppose that we eschew knowledge of the solution and try to minimize each  $f_i$  by the method of steepest descent. This is an iterative method that replaces the current iterate,  $m_i^{(n)}$ , with

$$\begin{aligned} m_i^{(n+1)} &= m_i^{(n)} - t_n \nabla f_i(m_i^{(n)}) \\ &= m_i^{(n)} - t_n \sum_{x_j \in C_i} 2(m_i^{(n)} - x_j) \\ &= [1 - 2t_n \#(C_i)] m_i^{(n)} + [2t_n \#(C_i)] \bar{x}_i \\ &= (1 - \alpha_n) m_i^{(n)} + \alpha_n \bar{x}_i, \end{aligned}$$

```

initialize  $m_1, \dots, m_k$  and  $\alpha$ 
do until termination:
  draw  $x \sim \text{Uniform}(\{x_1, \dots, x_N\})$ 
  determine  $m_*(x)$ 
  replace  $m_*(x)$  with  $(1 - \alpha)m_*(x) + \alpha x$ 
  update  $\alpha$ 

```

Figure 2: The SOM algorithm for  $k$ -means clustering.

where  $\#(C_i)$  is the cardinality of  $C_i$ . In numerical optimization, the step length control parameter  $t_n$  (equivalently  $\alpha_n$ ) is usually determined by performing a line search; however, one can also specify certain predetermined sequences, as in stochastic approximation. This may well impress the reader as an absurdly inefficient way to compute a mean (especially as the iteration formula requires knowing the mean!), but let us see where it leads.

To try to decrease  $\tilde{W}$ , we first set  $C_i = \tilde{C}_i$ . Then, for each  $i = 1, \dots, k$ ,  $m_i - \bar{x}_i$  is a descent direction for  $\tilde{W}$ . Unfortunately, identifying descent direction  $i$  requires knowing  $\bar{x}_i$ . We attempt to circumvent this difficulty by randomly drawing  $x \in \{x_1, \dots, x_N\}$  and determining  $m_*(x)$ . The choice of  $x$  thus determines which  $m_i$  is to be varied. Furthermore, we use  $x$  to estimate  $\bar{x}_*$ , resulting in an iterative algorithm that replaces  $m_i^{(n)} = m_*^{(n)}(x)$  with

$$\hat{m}_i^{(n+1)} = (1 - \alpha_n) m_i^{(n)} + \alpha_n x.$$

Under uniform random sampling of  $\{x_1, \dots, x_N\}$ ,

$$E\left(\hat{m}_i^{(n+1)}\right) = (1 - \alpha_n) m_i^{(n)} + \alpha_n E(x) = m_i^{(n+1)};$$

however,

$$E\left\|x - \hat{m}_i^{(n+1)}\right\|^2 \geq E\left\|x - m_i^{(n+1)}\right\|^2$$

and it may be that replacing  $m_i^{(n)}$  with  $\hat{m}_i^{(n+1)}$  actually increases  $\tilde{W}$ . Nevertheless, we have derived a crude but plausible algorithm for attempting to decrease  $\tilde{W}$  without explicit knowledge of the partition  $\tilde{C}_1, \dots, \tilde{C}_k$ . Anticipating material in Section 3, we remark that this algorithm is a special case of the on-line algorithm for self-organizing maps (SOM); it is summarized in Figure 2.

Let us compare the MacQueen and SOM algorithms for  $k$ -means clustering. MacQueen's algorithm decreases  $\tilde{W}$ , producing a partition  $(C_1^*, \dots, C_k^*)$ . If this partition minimizes  $\tilde{W}$ , then  $(C_1^*, \dots, C_k^*; \bar{x}_1, \dots, \bar{x}_k)$  minimizes  $W$ . The SOM algorithm attempts to decrease  $\tilde{W}$ , producing representatives  $(m_1^*, \dots, m_k^*)$ . If these representatives minimize  $\tilde{W}$ , then  $(\tilde{C}_1, \dots, \tilde{C}_k; m_1^*, \dots, m_k^*)$  minimizes  $W$ . Thus, both algorithms attempt to minimize the same objective function,  $W$ , and we can inquire (without any prejudices about self-organizing maps) how successfully they do so.

We conclude our discussion of  $k$ -means clustering with an example. Consider the  $N = 1000$  points in  $\mathfrak{R}^2$  displayed in Figure 3. For  $k = 20$ , we attempt to minimize  $W$  using  $n = 2000$  iterations of, respectively, the SOM algorithm and two variants of MacQueen's algorithm. Compared to traditional applications of  $k$ -means clustering, these are large values of  $N$  and  $k$ ; however, it is for large values of  $N$  and  $k$  that self-organizing maps are often recommended. One purpose of the present study is to interrogate that recommendation.

For  $(N = 1000, k = 20, n = 2000)$ , we do not expect to find global—or even local—minimizers of  $W$ . Instead, we investigate the effectiveness of each algorithm in finding small values of  $W$ . To

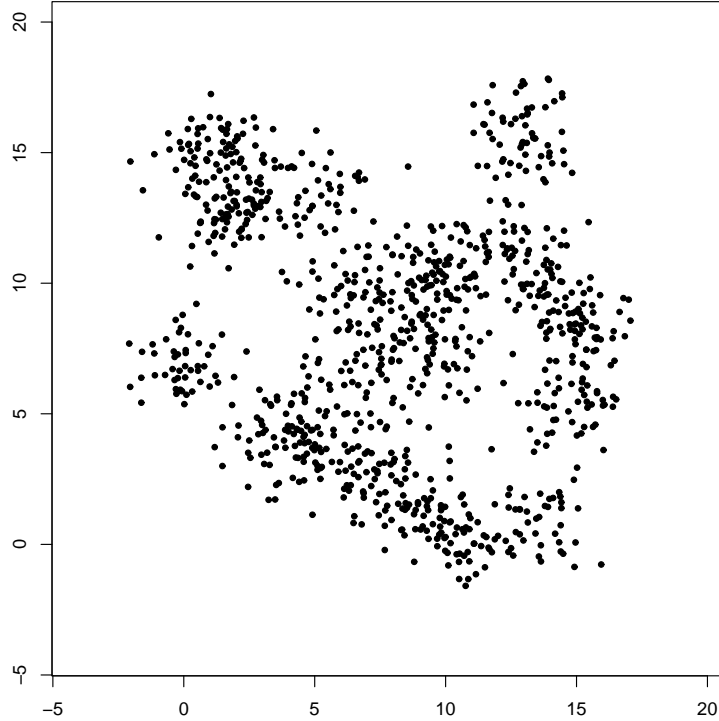


Figure 3: A scatter diagram of  $N = 1000$  points in  $\mathfrak{R}^2$ .

do so, we generate 1000 sets of initial values of  $m_1, \dots, m_{20}$ . Each set was generated by simple random sampling (without replacement) from  $\{x_1, \dots, x_{1000}\}$ . Each algorithm was started one time from each set of initial values. For MacQueen’s algorithm, which manipulates clusters rather than representatives, the clusters were initialized by taking  $C_i$  to be the  $x_j$  to whom the nearest representative is  $m_i$ . For the SOM algorithm, the values of  $\alpha$  followed Ripley’s (1996, p. 323) observation that “a typical specification is that  $\alpha$  might decline linearly from 1.0 to 0.04 over 1000 examples, then linearly to zero over the second thousand...”

Each algorithm modifies its current iterate in light of a single  $x_j$ . Our derivation of the SOM algorithm necessitates drawing  $x_j$  from a discrete uniform distribution on  $\{x_1, \dots, x_{1000}\}$ . Our *randomized MacQueen algorithm* does likewise; in contrast, our *cyclical MacQueen algorithm* begins each run by generating a random permutation of  $\{x_1, \dots, x_{1000}\}$ , then cycling through the  $x_j$  in the permuted order.

Each run of the SOM algorithm returns the value of  $\tilde{W}$  after 2000 iterations. Each run of each MacQueen algorithm returns the value of  $\bar{W}$  after 2000 iterations. The five-number summaries of these samples presented in Table 1 clearly suggest that the SOM algorithm is competitive with—if not superior to—the MacQueen algorithms. These results are provocative and invite further study of the  $\tilde{W}$  approach to  $k$ -means clustering; however, the present study takes no position on which algorithms for  $k$ -means clustering should be preferred.

Clustering Algorithm	min	$q_1$	$q_2$	$q_3$	max
SOM	1531	1610	1644	1705	2218
randomized MacQueen	1610	1956	2101	2309	4303
cyclical MacQueen	1529	1701	1799	1917	2887

Table 1: Five-number summaries of 1000 values of  $W$  returned by each of three algorithms for 20-means clustering the 1000 points in Figure 3. The  $q_i$  are the sample quartiles.

### 3 Self-Organizing Maps

As in Section 2, suppose that we seek  $m_1, \dots, m_k \in \mathfrak{R}^p$  to represent  $x_1, \dots, x_N \in \mathfrak{R}^p$ . Self-organizing maps (SOMs) are iterative algorithms that discover a set of representatives. The SOM algorithm for  $k$ -means clustering is a special case of a SOM; in this case, the representatives tend toward the means of clusters defined by a popular squared error criterion. More generally, it is not so clear what the  $m_i$  represent. Interpretation is complicated by the difficulty of identifying a specified SOM with an error criterion that it might be considered to minimize. Our derivation of the SOM algorithm for  $k$ -means clustering appears to be a special case of Růžička’s (1993) analysis of SOMs. In general, however, SOMs cannot be interpreted as algorithms that minimize a fixed error criterion; see, for example, Erwin, Obermayer, and Schulten (1992). Ripley (1996, p. 323) likened “the spirit of SOM” to the process of discretizing  $\mathfrak{R}^p$ , then representing each bin by averaging the  $x_i$  that lie within it. As we shall see, SOMs offer a compromise between discovering structure in the data (clustering the  $x_i$ ) and imposing structure upon the data (discretizing  $\mathfrak{R}^p$ ).

The signature feature of SOMs is the imposition of an external *topological ordering* on the set of representatives. This is usually accomplished by identifying the representatives with a regular grid in one or two dimensions. (The latter is more common; both square and hexagonal grids are used.) The topology of the grid is then used to define neighborhoods of representatives. As Ripley (1996, p. 323) explained, “the idea is that the representatives (called ‘weights’ by Kohonen) are spatially correlated, so that representatives at nearby points on the grid are more similar [than] those which are widely separated.” Generalizing the SOM algorithm for  $k$ -means clustering, “on-line” SOMs update all of the representatives in a neighborhood of  $m_*(x)$ . We illustrate this methodology, and explore some of its implications, with a trivial example.

Consider the problem of partitioning  $N = 3$  distinct points, say  $\{10, 15, 20\} \subset \mathfrak{R}$ , into  $k = 3$  clusters. The only possible clusterings comprise  $\{10\}$ ,  $\{15\}$ , and  $\{20\}$ ; and the only reasonable representatives of these clusters are 10, 15, and 20. Of course, these choices minimize the error criterion,  $W$ , for  $k$ -means clustering. In  $k$ -means clustering, the indexing of these choices is completely arbitrary, i.e., there is no reason to prefer the solution  $(m_1 = 10, m_2 = 15, m_3 = 20)$  to the solution  $(m_1 = 10, m_2 = 20, m_3 = 15)$ .

Now suppose that we impose the topological ordering  $m_1 \leftrightarrow m_2 \leftrightarrow m_3$ . This ordering expresses a preference for  $m_2$  to lie between  $m_1$  and  $m_3$ , in which case we do prefer the solution  $(m_1 = 10, m_2 = 15, m_3 = 20)$  to the solution  $(m_1 = 10, m_2 = 20, m_3 = 15)$ . We used the latter solution to initialize a SOM with the following neighborhoods:

$$N(m_1) = \{m_1, m_2\}, \quad N(m_2) = \{m_1, m_2, m_3\}, \quad N(m_3) = \{m_2, m_3\}. \quad (4)$$

An iteration of the SOM draws  $x \sim \text{Uniform}(\{10, 15, 20\})$ , determines  $m_*(x)$  and  $N(m_*(x))$ , then sets

$$m_i^{(n+1)} \leftarrow [1 - \alpha_n(*, i)] m_i^{(n)} + \alpha_n(*, i) x$$

for each  $m_i^{(n)} \in N(m_*(x))$ . As indicated by our notation, the choice of  $\alpha$  might depend on which representative is being updated (e.g., one might prefer smaller  $\alpha$  for representatives that are farther from  $m_*(x)$  in the prescribed topological ordering); for this example, however, we used the same schedule of  $\alpha_n(*, i) = \alpha_n$  that we used in Section 2.

$m_1$	$m_2$	$m_3$
12.66	14.84	17.24
12.47	15.19	17.68
12.51	14.94	17.43
12.60	15.15	17.52
12.58	14.97	17.39
12.37	14.82	17.50
12.35	15.06	17.62
12.49	14.94	17.51
12.35	14.89	17.55
12.61	14.78	17.26
12.50	14.96	17.47

Table 2: Ten replicates (and their mean) of  $k = 3$  representatives of  $\{10, 15, 20\}$  discovered by a self-organizing map with the fixed topological ordering  $m_1 \leftrightarrow m_2 \leftrightarrow m_3$ .

The results of 10 runs of the specified SOM are reported in Table 2. Two consequences of the topological ordering are immediately apparent. First, the topological ordering  $m_1 \leftrightarrow m_2 \leftrightarrow m_3$  successfully encouraged the SOM to place  $m_2$  between  $m_1$  and  $m_3$ , thereby illustrating how the signature feature of SOMs can affect global behavior. This effect was both anticipated and desired. Second, the representatives  $m_1$  and  $m_3$  discovered by the SOM do not correspond to any of the clusters. Rather, these representatives approximate the pairwise averages  $(10 + 15)/2 = 12.5$  and  $(15 + 20)/2 = 17.5$ . This illustrates how the topological ordering that a SOM imposes on the data can confound interpretation of the representatives. The topological ordering not only affects how a solution is labelled, but also what qualifies as a solution.

Perhaps for this reason, it is standard practice to shrink the neighborhoods defined by the topological ordering as the SOM progresses. With a judicious schedule of neighborhoods, this practice allows the topological ordering to influence global behavior while mitigating its effect on local behavior. To illustrate, we repeated the previous experiment, using the neighborhoods specified in (4) for the first 100 iterations and the trivial neighborhoods

$$N(m_1) = \{m_1\}, \quad N(m_2) = \{m_2\}, \quad N(m_3) = \{m_3\}.$$

thereafter. In each of 30 runs, this SOM returned  $(m_1 = 10, m_2 = 15, m_3 = 20)$ , the natural cluster representatives with the desired topological ordering.

Now, if we use a SOM for which the topological neighborhood of each  $m_i$  is eventually just  $\{m_i\}$ , then the SOM approximates  $k$ -means clustering in its limit. By affecting global behavior, the topological ordering may affect which locally optimal  $k$ -means clustering the SOM prefers. It will certainly affect how the clusters are labelled, and therefore how they are organized for subsequent display. The salient question becomes: if we desire representatives of a  $k$ -means clustering, then why should we allow an *a priori* topological ordering to determine how we display the cluster representatives?

One popular answer to the previous question appears to be something like the following: the topological orderings used in SOMs resemble artificial neural networks, which in turn resemble biological nervous systems. For example, the description on the back cover of Kohonen’s (2001) monograph concludes: “Last but not least, it should be mentioned that the SOM is one of the most realistic models of the biological brain function.” We are inclined to discount such rationales. Even if one were to concede that SOMs resemble brain function, it is hard to understand why the structure of (say) a large collection of text documents should. Indeed, if SOMs do resemble brain function and text documents do not, then would not the use of SOMs to cluster text documents be contraindicated?

There is another, less mystical explanation of how the imposition of a topological ordering benefits the display of cluster representatives: it is a device for ensuring that similar representatives will be displayed in contiguous locations. Indeed, Kohonen (2001, p. X) has remarked that this property provided the original motivation for SOMs:

“I just wanted an algorithm that would effectively map similar patterns (pattern vectors close to each other in the input signal space) onto contiguous locations in the output space.”

The undeniable appeal of displaying similar representatives in contiguous locations is illustrated in Kohonen’s (2001, p. 109) Figure 3.2, which is accompanied by the following caption:

“In this exemplary application, each processing element in the hexagonal grid holds a model of a short-time spectrum of natural speech (Finnish). Notice that neighboring models are mutually similar.”

The essential features of Kohonen’s exemplary application are present in our Figure 4, in which  $k = 9$  cluster representatives,  $m_i \in \mathbb{R}^5$ , are each displayed in parallel coordinates, organized in a  $3 \times 3$  rectangular grid. But this example also illustrates an unfortunate distortion: by insisting on contiguous locations, the display suppresses variation in the degrees of (dis)similarity between the  $m_i$ . For example, consider the vector displayed in the center of the grid: the Euclidean distance between the center vector and the one immediately above it is nine times the Euclidean distance between the center vector and the one immediately below it, yet the grid displays these vectors as though they were equidistant. In Section 4, we consider how to construct analogous displays that convey information about the distances between the  $m_i$ .

## 4 Multidimensional Scaling

There have been various attempts to incorporate information about the (dis)similarities of the  $m_i \in \mathbb{R}^p$  into a grid display of SOM representatives. For example, Ultsch (1993a,b) used greylevel shading in the spaces between contiguous grid locations to convey degrees of similarity between SOM representatives. Murtagh (1995) applied contiguity-constrained clustering to a grid of SOM representatives. Both of these strategies preserve the topological ordering imposed by the SOM. Less committed to SOM methodology, we drop the contiguity constraints and consider how to display  $k$  cluster representatives.

Multidimensional scaling (MDS) is a collection of techniques for embedding dissimilarity information in a (typically low-dimensional) Euclidean space. A set of objects is represented as a set of points in such a way that the Euclidean distances between the points approximate the dissimilarities between the objects. Different criteria for evaluating how well a set of pairwise distances

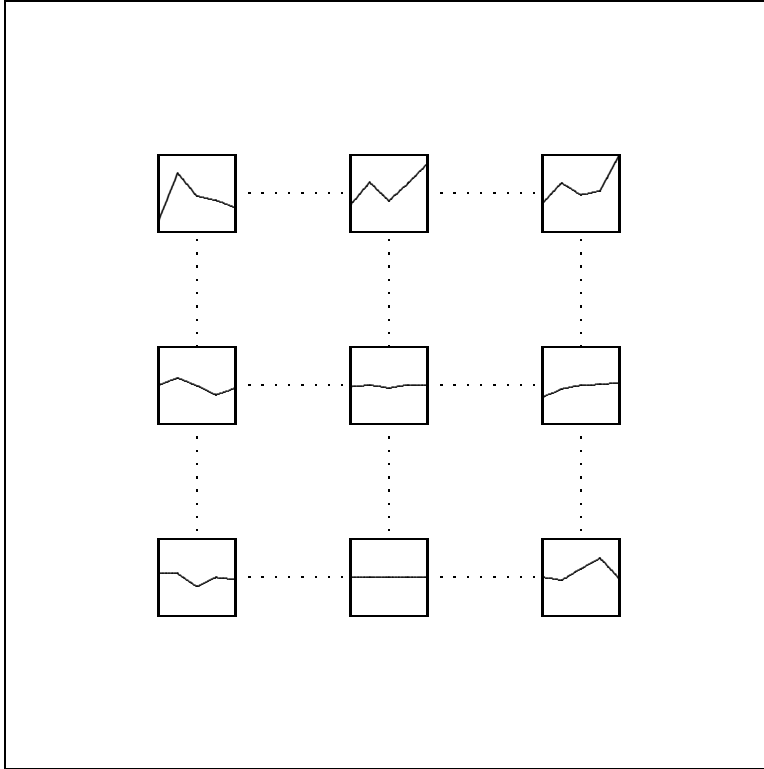


Figure 4: Nine cluster representatives,  $m_i \in \mathfrak{R}^5$ , displayed in parallel coordinates on a  $3 \times 3$  rectangular grid.

approximates a set of pairwise dissimilarities lead to different procedures for MDS. Examples include classical MDS (Torgerson, 1952; Gower, 1966), Kruskal’s (1964a,b) formulation of nonmetric MDS, and Sammon (1969) maps.

Ripley (1996, p. 323) described the process of constructing a SOM as “conceptually similar to multidimensional scaling” and observed that “it is often used to provide a crude version of multidimensional scaling.” Kohonen (2001, p. 37) suggested that “Sammon’s mapping... is always recommended for a preliminary test of the data to be used for Self-Organizing Maps.” (This suggestion is somewhat curious, in that the computational expense of MDS is greater than the computational expense of SOM, so that the “crude version” of MDS provided by SOM is more easily rationalized when the number of objects is large.) Both of these remarks envision applying MDS to the set of individual data points,  $x_1, \dots, x_N \in \mathfrak{R}^p$ , which may be prohibitively expensive when  $N$  is large. In contrast, we propose applying MDS to the set of cluster representatives,  $m_1, \dots, m_k \in \mathfrak{R}^p$ , as a means of displaying them. In most applications, the magnitude of  $k$  will be well within the capabilities of existing algorithms for MDS.

Two examples of the proposed methodology are exhibited in Figures 5 and 6. Both figures display the same cluster representatives,  $m_i \in \mathfrak{R}^5$ , that were previously displayed in Figure 4. Whereas Figure 4 displayed the  $m_i$  on a rectangular grid, Figures 5 and 6 convey additional information about the spatial relations between the  $m_i$ . Each solid dot in these figures corresponds to one cluster representative. The dots were positioned by MDS, so that the 2-dimensional Euclidean distances between the dots approximate the 5-dimensional Euclidean distances between the  $m_i$ .

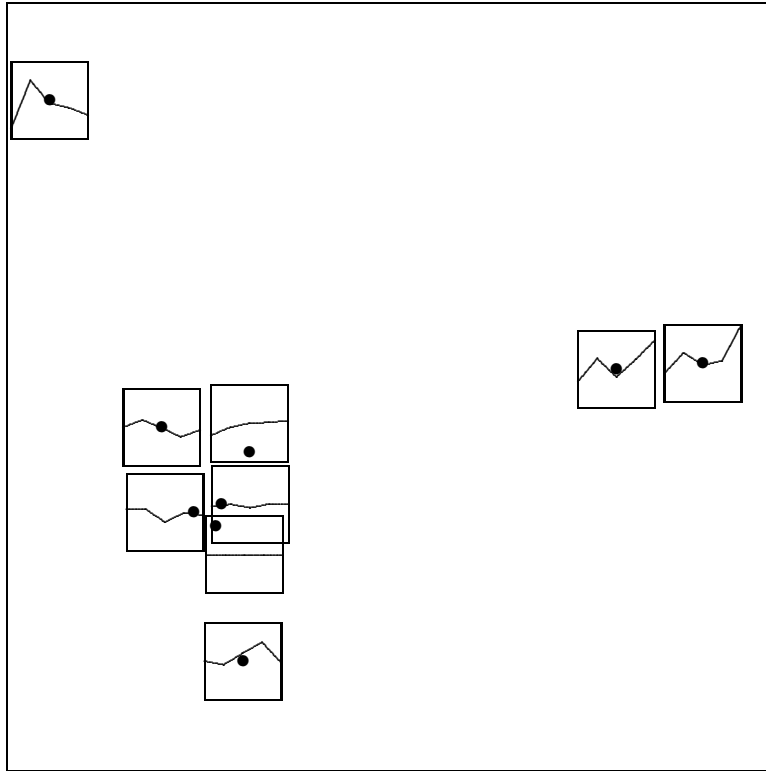


Figure 5: Nine cluster representatives,  $m_i \in \mathbb{R}^5$ , from Figure 4. The  $m_i$  were embedded in  $\mathbb{R}^2$  by classical multidimensional scaling (solid dots); each  $m_i$  is displayed, in parallel coordinates, in its own square.

(The locations were obtained by classical MDS, which is equivalent to projecting the  $m_i$  into the plane spanned by their first two principal components. Other MDS methods are better able to capture nonlinear structure in the data. Our use of classical MDS is merely illustrative and should not be construed as an endorsement of a particular method.) It is obvious that the grid display in Figure 4 suppresses valuable information.

An appealing feature of grid display is that the grid elements can be used to display additional information about the corresponding cluster representatives, as in our use of parallel coordinates in Figure 4. We have attempted to preserve this feature in Figures 5 and 6, in which we implemented two variations of a simple idea: associating with each dot in the 2-dimensional map a square that contains a parallel coordinates display of the corresponding cluster representative.

Figure 5 contains one square for each dot. To avoid distorting how one perceives the spatial relations between the dots, we would prefer to center each square at its corresponding dot. Unfortunately, several dots are tightly clustered and centering each square results in a rather busy display. To provide a clear view of each set of parallel coordinates, we repositioned several squares. An alternative visualization strategy was implemented in Figure 6. Here, each square is centered, but one square accommodates three tightly clustered dots. (It is centered at their mean.) The three corresponding displays in parallel coordinates are not easily distinguished, but this difficulty only emphasizes the similarity of these  $m_i$ . Alternatively, they might be displayed in different colors.

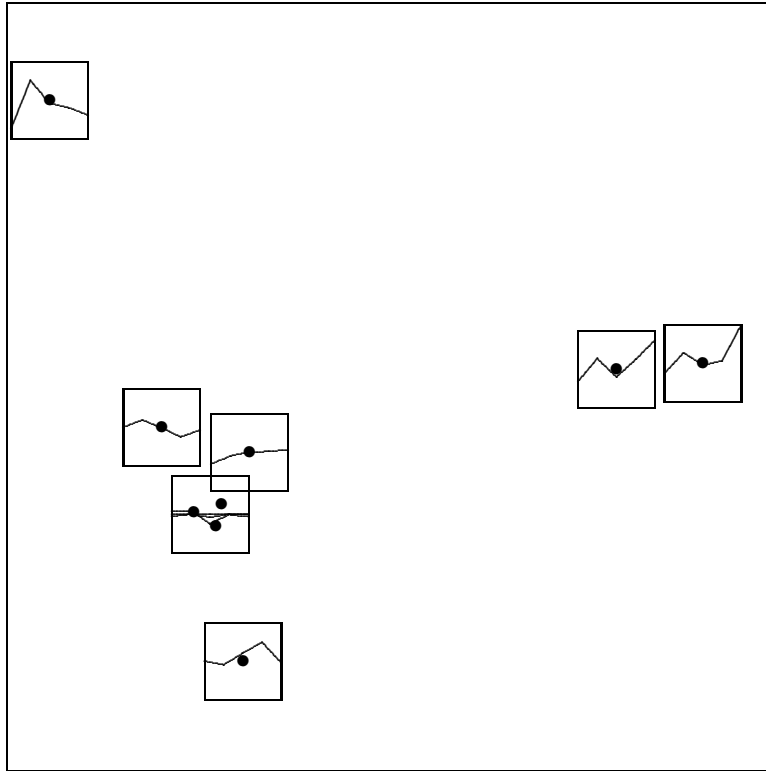


Figure 6: Nine cluster representatives,  $m_i \in \mathbb{R}^5$ , from Figure 4. The  $m_i$  were embedded in  $\mathbb{R}^2$  by classical multidimensional scaling (solid dots); each  $m_i$  is displayed, in parallel coordinates, in a corresponding square. Three tightly clustered  $m_i$  are displayed in one square.

## 5 Example: Gene Clustering

Cho et al. (1998) used high density oligonucleotide microarrays to measure mRNA transcript levels in synchronized yeast cells at regular 10-minute intervals during the cell cycle. These data were downloaded from <http://genomics.stanford.edu> by Tamayo et al. (1999), who analyzed their structure using SOMs, and by Yan (2004), who included them in a contributed package, `som`, for R, a free software environment for statistical computing and graphics. (Certain discrepancies should be noted. Cho et al. monitored 6220 transcripts and the Saccharomyces Cell Cycle Expression Database webpage refers to 6220 yeast open reading frames, whereas Tamayo et al. refer to 6218 yeast ORFs. Yan’s `yeast` data frame contains 6601 rows and appears to match the data in the Full Data in Downloadable Format webpage, which purports to contain “the full data for every gene in yeast.”) We analyzed the data in Yan’s `yeast` data frame, using the methods suggested in the `som` documentation for preprocessing. These methods approximate, but do not exactly match, the preprocessing performed by Tamayo et al.

The paper by Tamayo et al. (1999) provides a fascinating case study in how users regard SOMs and why they are drawn to them. First, it is clear that Tamayo et al. regarded SOMs as a clustering algorithm:

“This paper describes the application of self-organizing maps, a type of mathematical cluster analysis that is particularly well-suited for recognizing and classifying features

in complex, multidimensional data.” (p. 2907)

“These [other clustering] techniques include Bayesian clustering, k-means clustering, and self-organizing maps (SOMs).” (p. 2907)

Significantly, Tamayo et al. named the software package that they developed GENECLUSTER.

Tamayo et al. (1999) preferred SOMs to  $k$ -means clustering precisely because they admired the structure imposed by topological ordering:

“k-means clustering is a completely unstructured approach, which proceeds in an extremely local fashion and produces an unorganized collection of clusters that is not conducive to interpretation.” (p. 2907)

“SOMs... are ideally suited to exploratory data analysis, allowing one to impose partial structure on the clusters... and facilitating easy visualization and interpretation.” (p. 2907)

These passages reveal a slightly idiosyncratic vision of exploratory data analysis, and the tension between discovering patterns *in* the data (“SOMs... expose the fundamental patterns in the data”) and imposing patterns *on* the data (“SOMs impose structure on the data”) persists through their paper.

Finally, Tamayo et al. (1999, p. 2909) asserted that “each cluster is represented by its average expression pattern...” As discussed in Section 3, this interpretation is problematic—if not completely misleading—when topological ordering is used to define neighborhoods of the representatives. The assertion is (essentially) correct when the neighborhoods are trivial, but in that case SOMs are just algorithms for  $k$ -means clustering. In the SOM implemented by Tamayo et al., the radius used to define the neighborhoods “eventually” became zero.

We seek to rehabilitate  $k$ -means clustering, using MDS to facilitate interpretation of the “unorganized collection of clusters” produced by  $k$ -means clustering. For this example, there are  $N = 760$  objects (genes) in  $p = 16$  dimensions to be clustered after preprocessing. Because Tamayo et al. (1999) reported results for a  $6 \times 5$  rectangular grid, we present results for  $k = 30$  clusters.

The results displayed in Figure 7 were obtained as follows:

1. K-means clustering. Cluster representatives,  $m_1, \dots, m_{30} \in \mathfrak{R}^{16}$ , were obtained using the SOM algorithm for  $k$ -means clustering, described in Figure 2. The  $m_i$  were initialized by drawing a simple random sample (without replacement) from the set of  $N = 760$  genes. The step length control parameter  $\alpha$  was decreased linearly from  $\alpha = 1$  to  $\alpha = 0$  in increments of 0.00005, for a total of 20,000 iterations.
2. Multidimensional scaling. Embedded cluster representatives,  $\hat{m}_1, \dots, \hat{m}_{30} \in \mathfrak{R}^2$ , were obtained from the  $m_i \in \mathfrak{R}^{16}$  by classical MDS. Each solid dot in Figure 7 corresponds to one  $\hat{m}_i$ . Each  $m_i$  is also displayed in parallel coordinates. Where possible,  $m_i$  is displayed in a square centered at  $\hat{m}_i$ ; in several cases of tightly clustered  $\hat{m}_i$ , several  $m_i$  are displayed in one square centered at the mean of the corresponding  $\hat{m}_i$ .

Figure 7 does not display 30 vectors in parallel coordinates as efficiently as does a  $6 \times 5$  rectangular grid. The spatial inefficiencies that accompany Figure 7 are the price of using space to convey valuable information about how the vectors are related. We submit that this is a small price to pay, that one can more easily discern relations between cluster representatives by embedding them in  $\mathfrak{R}^2$  than by forcing them onto a regular grid. It is quite clear that the clusters displayed in Figure 7 do not lie on a regular grid. To paraphrase Tamayo et al. (1999), grids impose structure on the data, whereas embedding discovers structure in the data.

## 6 Discussion

Contrasting SOMs with the *curvilinear component analysis* method of Demartines and Héroult (1997), Kohonen (2001, p. 39) observed that in curvilinear component analysis “the data items are first clustered and then the clusters are mapped,” whereas “the Self-Organizing Map... combines the clustering and projection operations.” However, in the case of SOMs, we believe that conflating clustering and projection confounds interpretation, so that these operations are best performed separately.

Previous authors have observed that SOMs are intimately related to  $k$ -means clustering (a classical clustering method) and multidimensional scaling (a classical projection method). What do SOMs offer that these methods do not? One often hears that SOMs can be used on large data sets, whereas  $k$ -means clustering and MDS cannot. In our view, this contention is based on false and misleading comparisons.

A fair comparison of the computational efficiencies of SOMs and  $k$ -means clustering requires careful consideration of what each is expected to accomplish. For  $k$ -means clustering, there are three plausible levels of expectation:

1. Find the partition (or the corresponding set of cluster means) that is globally optimal with respect to the squared error criterion  $W$ . Indeed, it is quite expensive to find globally optimal partitions and  $N = 150$  is considered large for this problem. See, for example, Du Merle et al. (2000).
2. Find a partition that is locally optimal with respect to the squared error criterion  $W$ . Finding locally optimal partitions is considerably less expensive than finding globally optimal partitions, but doing so requires running an iterative method for  $k$ -means clustering, e.g., MacQueen’s algorithm (Figure 1), until convergence. If  $N$  is large, then this may be prohibitively expensive.
3. Decrease the squared error criterion  $W$  as much as possible subject to a specified budget for computation. Our inclination is to say that  $N$  is large when one cannot afford to find a locally optimal partition and therefore must impose a budget. This is the situation that we anticipated in designing the numerical experiments performed in Section 2. Any iterative method for  $k$ -means clustering can be adapted for this purpose.

Now, in what sense are SOMs less expensive than  $k$ -means clustering? They are no more scalable than (say) MacQueen’s algorithm for  $k$ -means clustering; the difference is that one usually runs an algorithm for  $k$ -means clustering to convergence, whereas one usually runs a SOM for a fixed number of iterations and accepts the result. Fair comparison requires comparable tasks, e.g., investigating how much can be accomplished with a fixed budget of computational resources.

In fact, the SOM algorithm for  $k$ -means clustering (Figure 2) outperformed MacQueen’s algorithm in the simple experiment reported in Section 2. Although MacQueen’s algorithm is not the state of the art, this result is provocative and the SOM algorithm for  $k$ -means clustering surely warrants further investigation. But this is not the algorithm that has generated so much enthusiasm for SOMs.

It is certainly true that it is more expensive to submit  $N$  objects to MDS than to submit  $N$  objects to a SOM. However, as explained in Section 4, this is not a fair comparison. Submitting  $N$  objects to MDS results in a map of  $N$  objects, whereas a SOM produces a map of  $k$  objects. The expense of submitting  $k$  objects to MDS is rarely prohibitive.

In summary, a judicious combination of algorithms for  $k$ -means clustering (quite possibly involving the SOM algorithm for  $k$ -means clustering) and multidimensional scaling produces more easily interpreted results than do SOMs, and at comparable computational expense.

## Acknowledgments

This research was partially supported by a subcontract to a Phase II SBIR grant for *Proteomics Software for Cancer Diagnostics*, awarded to Incogen by the National Institutes of Health.

## References

- [1] R. J. Cho, M. J. Campbell, E. A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. G. Wolfsberg, A. E. Gabrielian, D. Landsman, D. J. Lockhart, and R. W. Davis. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2:65–73, 1998.
- [2] P. Demartines and J. Héroult. Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8:148–154, 1997.
- [3] O. Du Merle, P. Hansen, B. Jaumard, and N. Mladenović. An interior point algorithm for minimum sum-of-squares clustering. *SIAM Journal on Scientific Computing*, 21:1485–1505, 2000.
- [4] E. Erwin, K. Obermayer, and K. Schulten. Self-organizing maps: Ordering, convergence properties and energy functions. *Biological Cybernetics*, 67:47–55, 1992.
- [5] J. C. Gower. Some distance properties of latent root and vector methods in multivariate analysis. *Biometrika*, 53:325–338, 1966.
- [6] T. Kohonen. *Self-Organizing Maps*. Springer, New York, 2001. Third Edition.
- [7] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964.
- [8] J. B. Kruskal. Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29:28–42, 1964.
- [9] J. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, Berkeley, CA, 1967. University of California Press.
- [10] F. Murtagh. Interpreting the Kohonen self-organizing feature map using contiguity-constrained clustering. *Pattern Recognition Letters*, 16:399–408, 1995.
- [11] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, 1996.
- [12] P. Růžička. On the convergence of learning algorithm for topological maps. *Neural Network World*, 4:413–424, 1993.

- [13] J. W. Sammon. A non-linear mapping for data structure analysis. *IEEE Transactions on Computers*, 18:401–409, 1969.
- [14] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. Lander, and T. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Science*, 96:2907–2912, 1999.
- [15] W. S. Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17:401–419, 1952.
- [16] A. Ultsch. Knowledge extraction from self-organizing neural networks. In O. Opitz, B. Lausen, and R. Klar, editors, *Information and Classification*, pages 301–306. Springer, Berlin, 1993.
- [17] A. Ultsch. Self-organizing neural networks for visualization and classification. In O. Opitz, B. Lausen, and R. Klar, editors, *Information and Classification*, pages 307–313. Springer, Berlin, 1993.
- [18] J. Yan. The som package. Comprehensive R Archive Network, September 2004. Available at <http://www.r-project.org>.

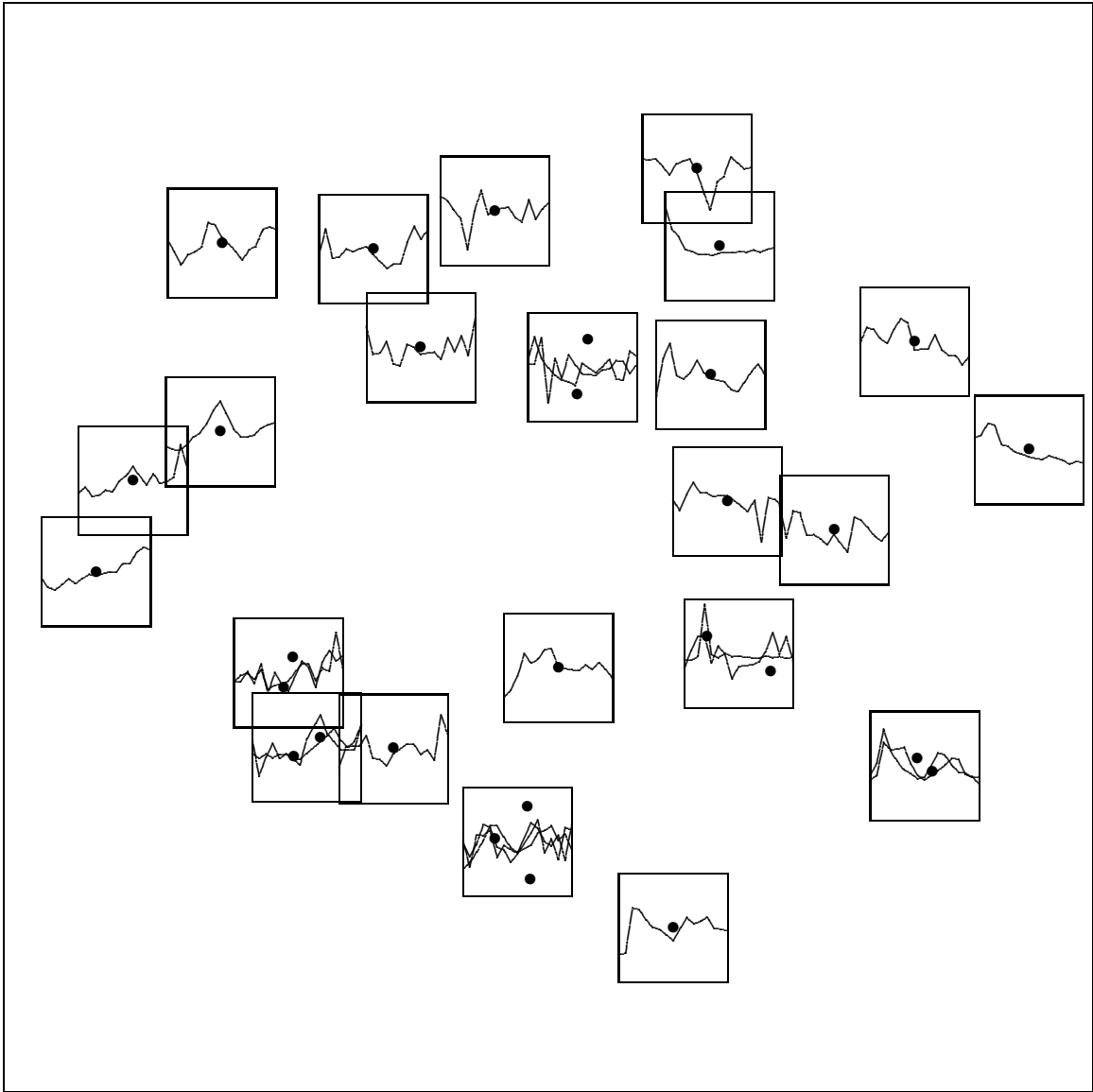


Figure 7: Thirty cluster representatives,  $m_i \in \mathbb{R}^{16}$ , discovered by applying the SOM algorithm for  $k$ -means clustering to 760 gene expression profiles from the **yeast** data set. The  $m_i$  were embedded in  $\mathbb{R}^2$  by classical multidimensional scaling (solid dots); each  $m_i$  is displayed in parallel coordinates.