

Improving the efficiency of multidimensional scaling in the analysis of high-dimensional data using singular value decomposition

Christophe Bécavin¹, Nicolas Tchitchek¹, Colette Mints-Eya^{1,2}, Annick Lesne^{1,3} and Arndt Benecke^{1,2,*}

¹Institut des Hautes Études Scientifiques, 35 route de Chartres, 91440 Bures sur Yvette, ²Institut de Recherche Interdisciplinaire, CNRS UMR3078, Université de Lille I, II, 50 avenue de Halley, 59658 Villeneuve d'Ascq and

³Laboratoire Physique Théorique de la Matière Condensée, CNRS UMR7600, Université Pierre et Marie Curie Paris 6, 4 place Jussieu, 75252 Paris Cedex 05, France

Associate Editor: Jonathan Wren

ABSTRACT

Motivation: Multidimensional scaling (MDS) is a well-known multivariate statistical analysis method used for dimensionality reduction and visualization of similarities and dissimilarities in multidimensional data. The advantage of MDS with respect to singular value decomposition (SVD) based methods such as principal component analysis is its superior fidelity in representing the distance between different instances specially for high-dimensional geometric objects. Here, we investigate the importance of the choice of initial conditions for MDS, and show that SVD is the best choice to initiate MDS. Furthermore, we demonstrate that the use of the first principal components of SVD to initiate the MDS algorithm is more efficient than an iteration through all the principal components. Adding stochasticity to the molecular dynamics simulations typically used for MDS of large datasets, contrary to previous suggestions, likewise does not increase accuracy. Finally, we introduce a *k* nearest neighbor method to analyze the local structure of the geometric objects and use it to control the quality of the dimensionality reduction.

Results: We demonstrate here the, to our knowledge, most efficient and accurate initialization strategy for MDS algorithms, reducing considerably computational load. SVD-based initialization renders MDS methodology much more useful in the analysis of high-dimensional data such as functional genomics datasets.

Contact: arndt@ihes.fr

Received on November 4, 2010; revised on February 6, 2011; accepted on March 13, 2011

1 INTRODUCTION

The appropriate and faithful visualization of high-dimensional data is often a prerequisite for their analysis as the human visual cortex is still one of the most powerful tools to detect and conceptualize structure in data (Holmes, 2006). Furthermore, communication of numerical and statistical results is greatly aided by the intuition arising from appropriate representations of data. Different methods

for the required dimensionality reduction have been developed (Berthold and Hand, 2003).

An entire family of approaches, such as principal component analysis (PCA) finds the minimal orthonormal basis using a mathematical tool called singular value decomposition (SVD). These methods, using different similarity or dissimilarity measures such as covariance or correlation, order the *ensemble* of components by their statistical deviation, and for visualization only the first two or three components are retained. Thereby, the statistical information in the first components are entirely retained, whereas one of the subsequent components is entirely lost. Today's high-dimensional biological datasets can easily contain thousands of instances (number of measures) with 10^5 – 10^9 variables (number of parameters measured). The repartition of information is usually homogeneous over the entire number of variables. In consequence, considering only the first components given by SVD-based techniques is not necessarily the best choice.

Multidimensional scaling (MDS) is a methodology that reduces dimensionality using only the information of similarities or dissimilarities between instances, hereafter regrouped in the general term of 'distance'. The search for an optimal configuration, is reduced to finding the global minimum of a function evaluating the loss of distance information. To be sure to find an acceptable minima (i) an initial state for the optimization algorithm, and (ii) an optimization algorithm and the appropriate parameters have to be appropriately chosen. Recently, Andrecut (2009) has shown that the best choice for the second is a molecular dynamics multidimensional scaling approach.

We demonstrate here that the choice of the initial position is paramount to the quality of the representation and its computational efficiency. By using SVD for providing an initial configuration for MDS, we obtain a significantly increased computational efficacy. Interestingly, we also demonstrate that performing an iterative MDS or adding stochastic energy during the molecular dynamics, MDS execution do not increase performance or reproducibility of the algorithm. We also investigate the local structure of the geometric objects after dimensionality reduction with our different methodologies, and then evaluate the accuracy of the different approaches developed here on biological data. These investigations and the use of SVD to the initial state allow to better define and

*To whom correspondence should be addressed.

control the dimensionality reduction process for high-dimensional data.

2 METHODS

2.1 SVD

Given a data matrix X with n rows and p columns and x_{ij} its value in row i and column j , we denote \tilde{X}_i the p components vector corresponding to row i of the matrix, and \tilde{X}_j the n components vector corresponding to column j of the matrix. A set of vector \tilde{X}_i is then a set of instances, whereas a set of vector \tilde{X}_j is a set of variables. In all the following, we will use this notation for vectors extracted from X .

It is known (Schmidt and Stewart, 1992) that every rectangular matrix can be decomposed using its singular values:

$$X = USV^t \quad (1)$$

where U (left singular vectors) and V (right singular vectors) are both square orthogonal matrices, and S is a rectangular matrix containing the singular values (s_i) which are positive ($S_{ij} = s_i$ and $S_{ij} = 0$). U , S and V are reorganized in order to have $s_1 > s_2 > \dots > s_r$, with r being the rank of S . Generally, before performing SVD X is centered, so the mean of each column is equal to zero. In this context, $\text{rank}(X) = \text{rank}(S) \leq \min(n-1, p)$ if X is n, p .

Singular value decomposition provides three major types of information:

- A new data matrix X_{new} , which represent the data points in a new orthonormal basis with a minimum number of components, and where distance between the instances is preserved.
- Inertia parameters $c_i = s_i / \sum_i s_i$ (with $\sum_i c_i = 1$) indicate the SD and relative contribution of the cloud of points on each principal component.
- The matrix V carrying the individual contributions to each principal component. These different types of information have already previously been used in the literature to infer biological knowledge in various settings [Alter *et al.* (2000, 2003); Fellenberg *et al.* (2001); Wall *et al.* (2003)]. The simplest way to find SVD, is to search first for the eigenvalues and the eigenvectors of the inner and outer products. As finding the eigenvalues of a matrix X with n rows and p columns, is hard to perform for objects with a high number of variables, this step is only feasible if either n or p are small (typically inferior to 1000, which is usually the case in biological datasets). If both n and p are large, one is obliged to use iterative SVD techniques as shown in Schmidt and Stewart (1992). One advantage of using SVD is its close link to classical techniques of dimensionality reduction such as PCA, classical scaling (cMDS), principal component correlation analysis (PCCA) and correspondence analysis. The different results of these techniques can be obtained using SVD and a proper normalization of the data, as shown below. SVD allows to demonstrate that the inner-product (XX^t) and outer-product (X^tX) of a data matrix X have the same eigenvalues λ_i , with $\lambda_i = s_i^2$. If $X = USV^t$ then:

$$XX^t = USV^t(VS^tU^t) = USS^tU^t \quad (2)$$

$$X^tX = (VS^tU^t)USV^t = VS^tSV^t \quad (3)$$

Note also that missing values in data can be imputed using SVD [Brock *et al.* (2008); Candes and Recht (2008); Troyanskaya *et al.* (2001)]. If the number of missing values is relatively low, the Eckart Young theorem (Eckart and Young, 1936), which is the most commonly used theorem for matrix approximation, assures that the result of the SVD will change only in the value of the last singular values. Hence, for a rapid imputation, the row average method (Troyanskaya *et al.*, 2001) can be used which is generally sufficiently precise in most cases. Also, PCA is a very good choice for the initial state for K -means clustering (Ding and He, 2004). In the new representation given by SVD, cluster structure of the data will then naturally appear, and thus provide a natural interpretation of clusters.

2.2 SVD and classical techniques of dimensionality reduction

PCA relies on the search of the eigenvectors' covariance matrix. Hence, performing PCA reduces to finding the outer-product's eigenvectors. The singular values of X are the square root of the outer-product's eigenvalues. The link between PCA and SVD then becomes obvious (Wall *et al.*, 2003). Classical scaling (cMDS for classical Multidimensional Scaling) was invented to embed a set of instances in the simplest space possible, with the constraint of preserving the Euclidean distance between data points. Euclidean distance can be written as a sum of inner-products $\tilde{X}_i \cdot \tilde{X}_j$, one can pass from an Euclidean distance matrix to an inner product matrix by a simple matrix manipulation called double centering (Torgerson, 1952). Consequently, classical scaling consists in finding eigenvalue factorization of the inner-product matrix, so it can be performed using SVD. The link given by SVD between inner and outer product matrices implies that PCA and classical scaling give the same results, a fact reflected by classical scaling sometimes being referred to principal coordinate analysis. Principal component correlation analysis (PCCA) uses correlation between variables to find a minimal orthonormal basis. After a proper normalization of the data with their SD: $\tilde{X} = \left(\frac{x_{ij}}{\sigma(\tilde{X}_j)} \right)$, PCCA is performed by eigenvalue factorization of the outer-product matrix. Hence, after normalization of the data PCCA results are given by SVD. Correspondence analysis is used in the dimensionality reduction of contingency tables obtained after an operation of counting on categorical data (Berthold and Hand, 2003). This method can be used for microarray data analyses (Fellenberg *et al.*, 2001) as each value of gene expression is, in fact, a count of the number of RNAs produced. Generally speaking, this technique is used to compare two vectors in terms of their distribution profiles using the chi-square distance. When the distance is equal to zero, both vectors have the same statistical distribution. It can be shown (Cuadras and Fortiana, 1995) that χ^2 distance can be reduced to an Euclidean distance after normalization of the data $\tilde{X}_{ik} = \frac{x_{ik} \sqrt{W}}{(\sum_i x_{ik})(\sum_i x_{il})}$. Thus, to find the minimal space which embeds the data and conserves the information of χ^2 distance one performs a cMDS or PCA on the rescaled data matrix using SVD results.

2.3 MDS

MDS is a class of techniques to represent instances in an r dimensional space given an initial state and a similarity or dissimilarity matrix (Cox and Cox, 2000; Kruskal and Wish, 1978). Recently, molecular dynamics (MD) approaches have been used to perform MDS for high-dimensional objects drastically increasing quality of the dimensionality reduction (Andrecut, 2009). We have also developed a similar approach based on a spring analogy. Data points are connected to all other instances with virtual springs. The springs will tend to return to their equilibrium length during molecular dynamics simulation. The equilibrium length for the spring between point i and point j will be defined as the Euclidean distance $d(\tilde{X}_i, \tilde{X}_j)$ in the initial state. For each instance \tilde{X}_i , a force is defined $F(\tilde{X}_i)$, which is the sum of all spring interactions $F_{spr}(\tilde{X}_i, \tilde{X}_j)$ with the other instances \tilde{X}_j , minus a friction term to avoid oscillation of the spring network:

$$F_{spr}(\tilde{X}_i, \tilde{X}_j) = -k_{ij}(\delta(\tilde{X}_i, \tilde{X}_j) - d(\tilde{X}_i, \tilde{X}_j))(\tilde{X}_j - \tilde{X}_i) \quad (4)$$

$$F(\tilde{X}_i) = \sum_{j \neq i} F_{spr}(\tilde{X}_i, \tilde{X}_j) - \gamma m_i \dot{\tilde{X}}_i \quad (5)$$

with $\delta(\tilde{X}_i, \tilde{X}_j)$ being the distance between instances in the r dimensional space, k_{ij} the strength of spring ij , γ the friction parameter and m_i the mass given to each point. We consider that every spring and all instances are equal in strength and weight so k_{ij} and m_i are the same for every i and j ($k_{ij} = k$ and $m_i = m$). It is, however, possible to use different parameters—for instance, according to experimental precision—if different weights shall be considered for the different instances. A molecular simulation using the force vector is then executed. Following Newton's law it follows: $m_i \ddot{\tilde{X}}_i = F(\tilde{X}_i)$, with $\ddot{\tilde{X}}_i$ the double temporal derivation of vector $\tilde{X}_i(t)$. In order to find the new position

Table 1. The different datasets used in this study

| ID | Dataset name | No. of instances | No. of variables |
|-----|---------------------|------------------|------------------|
| d1 | 96Cell | 96 | 32878 |
| d2 | 96Cell_T | 96 | 1553 |
| d3 | Iris | 150 | 4 |
| d4 | Wine | 178 | 13 |
| d5 | Stochast 200 | 200 | 50 |
| d6 | CCYier | 516 | 12 |
| d7 | Pima | 768 | 9 |
| d8 | 96Cell_T transposed | 1553 | 96 |
| d9 | Secom | 1567 | 590 |
| d10 | Ozone | 2565 | 72 |
| d11 | Stochast 3000 | 3000 | 300 |
| d12 | Ecoli | 4288 | 7 |
| d13 | Wave | 5000 | 22 |

and velocity of and instance at the next time step, a Verlet integration is used:

$$\bar{X}_i(t + \Delta t) = 2\bar{X}_i(t) - \bar{X}_i(t - \Delta t) + A\Delta t^2 \quad (6)$$

$$\dot{\bar{X}}_i(t) = \frac{\bar{X}_i(t + \Delta t) - \bar{X}_i(t - \Delta t)}{2\Delta t} \quad (7)$$

with $\dot{\bar{X}}_i(t)$ the temporal derivation of vector $\bar{X}_i(t)$. The algorithm is run with simulation time t increasing. To avoid divergence of the Verlet algorithm parameters of the simulation $k, m, \gamma, \Delta t$ have to be well chosen. Here we used: $k=1, m=5, \gamma=0.1, \Delta t=0.02$ (cf. Fig. 2A). For the initial state, the data provided to the MDS algorithm were rescaled to fit in a hypercube with a diameter of 6 by multiplying the initial state matrix by a scalar α . To control the minimization process at each time step, a cost function termed the Kruskal stress is calculated according to Cox and Cox (2000):

$$e = \sqrt{\frac{\sum_i \sum_j (\delta(i,j) - d(i,j))^2}{\sum_i \sum_j d(i,j)^2}} \quad (8)$$

this global parameter is a direct evaluation of the amount of energy in the system and hence the loss of distance information.

2.4 Datasets used in this study

To test and illustrate the algorithm discussed here, we have used several publicly available datasets of different origin. We have used two different transcriptome datasets. Briefly, the cellular transcriptome is defined as the *ensemble* of RNA molecules resulting from gene expression in a cell. Using microarray technology, in the human case, some 30 000 different RNA species can be quantified simultaneously. The dataset here referred to 'd1—96Cell' includes 96 transcriptome measurements generated from 32 individual human tissues under non-pathological conditions. This dataset was initially published by Dezsó *et al.* (2008), and is available for download from: <http://mace.ihes.fr> using accession number: 2914508814. The dataset here called 'd6—CCYier' [Iyer *et al.*, 1999; mace access. no.: 2960354318] is composed of 12 human fibroblast transcriptome data points generated over 24 h during the cell cycle. Note that we eliminated 1 (Interleukin 8, IL8) of the 517 genes as an outlier from this dataset. The dataset 'd2—96Cell_T' (cf. Table 1) is a derivative of the initial dataset d1—96Cell', where only genes were retained that are specific to one and only one human tissue as provided in (Dezsó *et al.*, 2008), and removing again one outlier gene (Probe_ID: 162105). The dataset 'd8—96Cell_T' (cf. Table 1) is the transposed (Instances, Variables) dataset 'd2—96Cell_T'. All transcriptome datasets were median normalized in log2-space and processed according to standard procedures (Benecke, 2008; Noth *et al.*, 2006). Seven additional datasets with no relation to biology were used. Both originate from the Machine Learning Repository (Frank and

Asuncion, 2010): <http://archive.ics.uci.edu/ml> (i) 'Iris' here 'd3—Iris', (ii) 'Wine' here 'd4—Wine', (iii) 'Pima Indians Diabetes' here 'd7—Pima', (iv) 'SECOM' here 'd9—Secom', (v) 'Ozone Level Detection' here 'd10—Ozone', (vi) 'E.Coli Genes' here 'd12—Ecoli' and (vii) 'Waveform Database Generator (Version 1)' here: 'd13—Wave'. Please refer to the ML repository for details on these data. Finally, we generated two random datasets: (i) one with 200 instances and 50 variables between -6 and 6 here 'd5—Stochast 200', (ii) the other with 3000 instances and 300 variables between -6 and 6 here 'd11—Stochast 3000'. The number of instances and the number of variables for all 13 datasets is given in Table 1.

3 RESULTS

3.1 Comparison of different initialization methods for MDS

We postulated that the inconveniences associated with the combined molecular dynamics MDS techniques (hereafter simply: MDS) related to the dependence on the choice of the initial condition for the simulation leading to insufficient control and being trapped in local minima on the one hand, as well as the large information loss when SVD techniques are used for dimensionality reduction on the other hand, can be overcome when both methods are combined. We therefore created an SVD—MDS algorithm which uses SVD to compute the initial state of a molecular dynamics simulated MDS. This SVD—MDS approach was then compared to SVD and MDS on 13 different datasets (Table 1). Figure 1 well illustrates the shortcomings of SVD and MDS alone and how SVD—MDS overcomes those. The dataset 'd1—96Cell' containing 96 different instances was used to compute a 2D representation using SVD (Fig. 1A), our combined SVD—MDS approach (Fig. 1B) and two examples of MDS initialized by random positions defining a 12 unit hypercube (Fig. 1C and D). According to the Kruskal stress e , MDS techniques (Fig. 1B–D) better preserve the distances between the instances and their relationship. The data cloud is better resolved (see also blow ups) and the global distance information loss is lower than for SVD.

In order to demonstrate generality of our approach, we next analyzed the 12 remaining datasets (Table 1) using four different approaches: (i) SVD only, (ii) SVD—MDS, (iii) MDS initialized with all data points placed at zero with minimal random noise (zeroMDS), and (iv) MDS initialized with random positions (stochastMDS). The results are reported in Table 2. In all cases, we reduced the dimensions to two. It becomes again apparent from the Kruskal stress that the MDS-based techniques systematically outperform the SVD. While stochastMDS, zeroMDS and SVD—MDS give similar results in terms of the final information loss, the number of time-steps needed to identify a minimum stress is greatly reduced using SVD—MDS (Table 2 and for four examples Fig. 2). Therefore, SVD—MDS approaches the final state (here defined as a Kruskal stress value) faster than either of the MDS methods. We show an example of stress evolution in Figure 3A where stochastMDS and zeroMDS are slow due to the existence of local minima, and SVD—MDS clearly outperform them.

3.2 Iterative dimensionality reduction using iSVD—MDS

We next wondered whether the dimensionality reduction could be further improved by a step-wise reduction of one dimension after another. To this end, we compared again the performance of the three

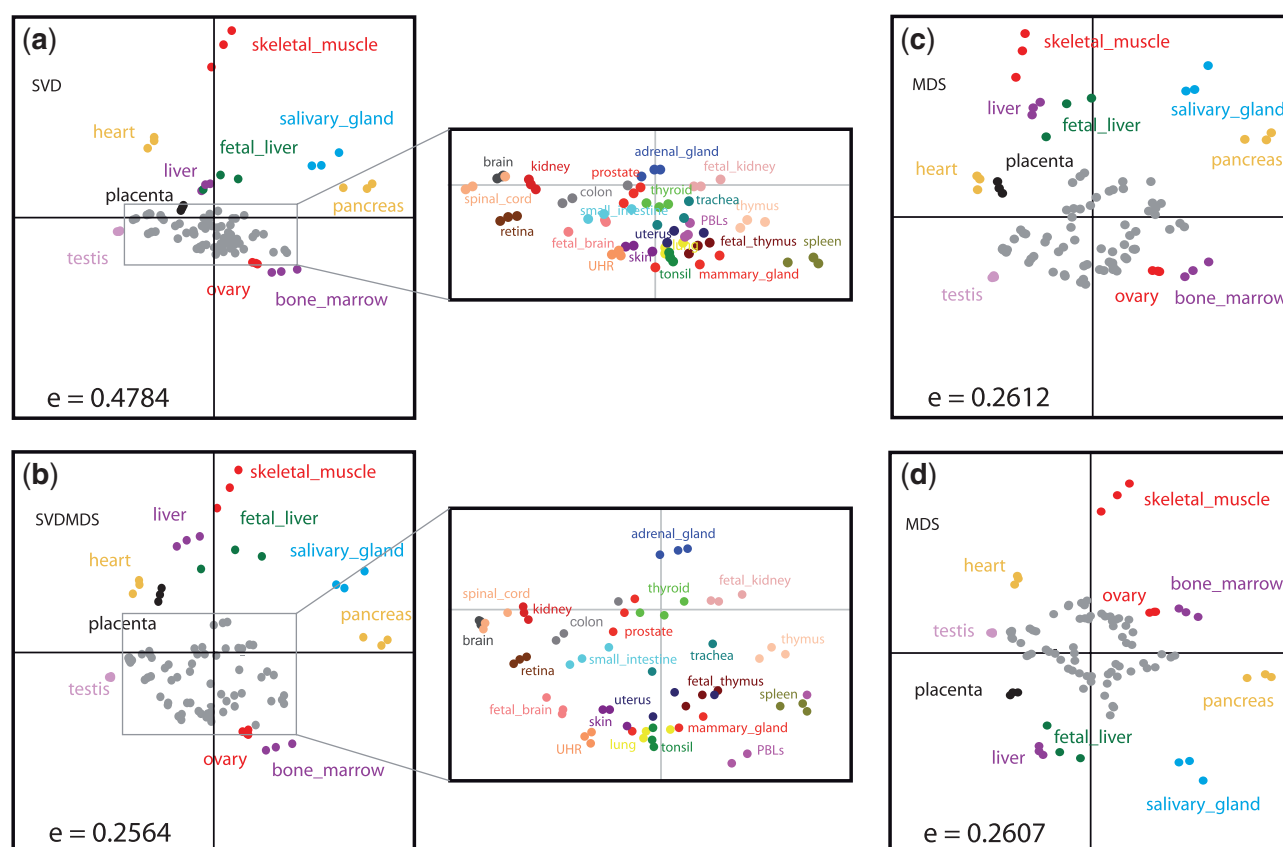


Fig. 1. Comparison of the results of different dimensionality reduction techniques on the same dataset. The dataset ‘d1—96Cell’, composed of 96 individual transcriptome profiles generated from 32 different human tissues (cf. Table 1 and Section 2.4) was represented in 2D space using: (a) SVD based on covariance, (b) SVD-initialized multidimensional scaling; (c) random initialized multidimensional scaling, and (d) as in (c) using the same algorithm and leading to a different random position matrix. The peripheral data points were color coded and labeled according to the human tissue analyzed. For (a) and (b) the central cloud of points has been zoomed into at the same scaling factor. The resulting Kruskal Stress e for each of the dimensionality reductions is indicated. Similar computations were used to generate Table 2.

techniques SVD–MDS, MDS and iterative SVD–MDS (iSVD–MDS) on the different datasets. In iSVD–MDS, for each successive round a SVD followed by a subsequent molecular dynamics MDS is performed. As can be seen in Figure 3A, SVD–MDS rapidly approaches a minimal Kruskal stress configuration over the simulation time. The previously described MDS procedure which uses stochastic initiation for the molecular dynamics simulation requires much more simulation time to find the same minimal stress configuration as the SVD–MDS algorithm. Finally, the iterative iSVD–MDS approach will also converge to the identical minimum obtained by the other methods; however, as for each component a separate simulation is performed the convergence time is greatly increased. Albeit many different simulations on the different datasets, we have never obtained a final configuration using iSVD–MDS where the Kruskal stress would allow to conclude on an improved performance when compared to SVD–MDS. Therefore, the iterative method does not allow for improved accuracy, but rather prolongs simulation time with no immediate gain (Table 3 summarizes the results). We next compared iSVD and iSVD–MDS methods to determine how the loss of information is distributed during iterative dimensionality reduction. As can be

seen in Figure 3B for both procedures, the amount of stress or lost information increases both relatively and absolutely with the number of components removed. Note also, that the iSVD–MDS method better preserves at every consecutive iteration the distance information of the object (Fig. 3B).

3.3 Molecular dynamics dimensionality reduction with added stochasticity

In Andreut (2009), an approach reminiscent of simulated annealing was used to avoid getting trapped in local minima during the molecular dynamics simulation. This combination of methods is equivalent to adding a stochastic force to all data points $F_{\text{stochastic}}(\vec{X}_i) = -T * s(t)$ where $s(t)$ is a random number given by a generalized Gaussian stochastic distribution, and T is the temperature of the system. By decreasing T exponentially during the simulation, one expects to reach the global minimum. Adding stochasticity to the molecular dynamics-driven MDS is, after Andreut (2009), required to insure reproducibility of the algorithmic performance.

Table 2. Results from the different MDS algorithms applied to the various datasets (c.f. Table 1)

| ID | Dataset name | Metric | SVD | SVD-MDS | | zeroMDS | | stochastMDS | |
|-----|---------------------|--------|--------|---------|------|---------|------|-------------|------|
| | | | | e | t | e | t | e | t |
| d1 | 96Cell | R^2 | 0.6472 | 0.3409 | 2500 | 0.352 | 2500 | 0.3478 | 2500 |
| d2 | 96Cell_T | Cov | 0.5001 | 0.1401 | 4500 | 0.146 | 4500 | 0.1503 | 4500 |
| d3 | Iris | Cov | 0.0421 | 0.0344 | 509 | 0.0343 | 3554 | 0.0344 | 4059 |
| d4 | Wine | Cov | 0.0010 | 0.0010 | 0 | 0.0064 | 4500 | 0.0061 | 4500 |
| d5 | Stochast 200 | Cov | 0.7513 | 0.4088 | 1500 | 0.4169 | 1500 | 0.4157 | 1500 |
| d6 | CCYier | Cov | 0.1634 | 0.0765 | 400 | 0.0932 | 3500 | 0.1079 | 4500 |
| d7 | Pima | Cov | 0.0964 | 0.0708 | 700 | 0.105 | 3500 | 0.1098 | 3500 |
| d8 | 96Cell_T transposed | R^2 | 0.6954 | 0.1498 | 4500 | 0.1572 | 4500 | 0.1715 | 4500 |
| d9 | Secom | Cov | 0.1801 | 0.1168 | 750 | 0.1217 | 4499 | 0.1283 | 4375 |
| d10 | Ozone | Cov | 0.1223 | 0.0935 | 712 | 0.0935 | 2587 | 0.0951 | 2143 |
| d11 | Stochast 3000 | Cov | 0.9067 | 0.4353 | 130 | 0.4382 | 130 | 0.438 | 130 |
| d12 | Ecoli | Cov | 0.1634 | 0.000 | 0 | 0.0202 | 4500 | 0.2484 | 4500 |
| d13 | Wave | Cov | 0.2922 | 0.2132 | 324 | 0.2132 | 2252 | 0.2132 | 1998 |

CoV, covariance; R^2 , correlation; e , Kruskal stress; t , time steps for MD simulation.

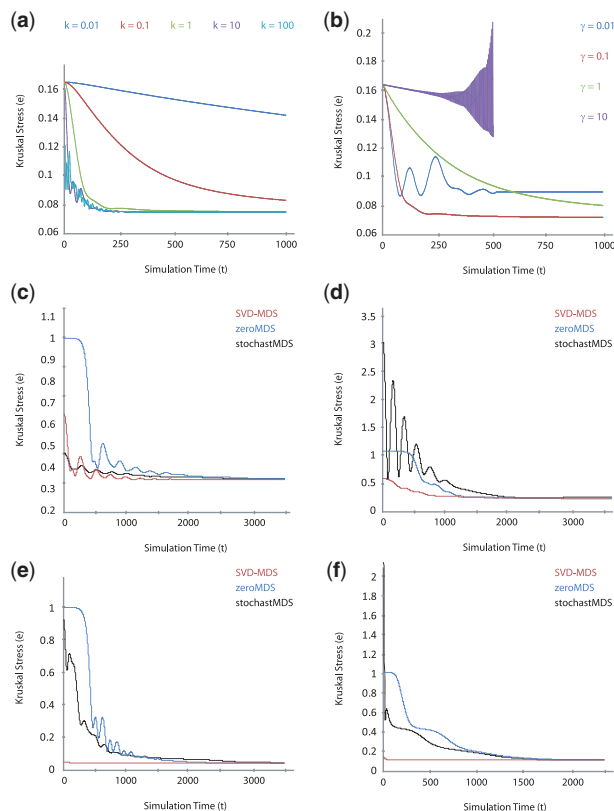


Fig. 2. Parameter optimization and Kruskal stress (e) evolution over the number of simulation iterations (t). Optimizing the choice of parameters k (a) and γ (b) using the dataset ‘d6—CCYier’ in covariance space. Comparison of the SVD-MDS, MDS initialized with all points in the center (zeroMDS), and MDS initialized by stochastic positions (stochastMDS) methods on different datasets (c) ‘d1—96Cell’ in correlation basis, (d) ‘d2—96Cell_T’ in covariance basis, (e) ‘d3—Iris’ in correlation basis, (f) ‘d10—Ozone’ in covariance basis.

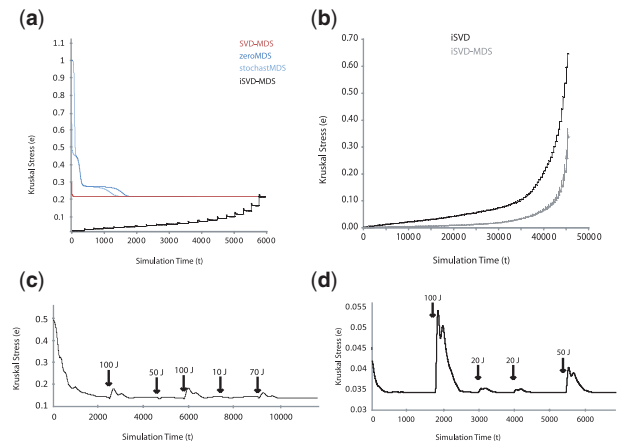


Fig. 3. Iterative SVD-MDS and robustness of SVD-MDS. (a) Comparison of the SVD-MDS, zeroMDS, stochastMDS and iterative SVD-MDS (iSVD-MDS) methods on dataset ‘d13—Wave’ in covariance basis. (b) Comparison of the iterative SVD (iSVD) and iSVD-MDS methods on dataset ‘d1—96Cell’ in correlation basis. Evolution of stress over number of simulation iterations with injection of energy, on different datasets (c) ‘d2—96Cell_T’ in covariance basis and (d) ‘d5—Iris’ in covariance basis.

To compare MD-MDS with our SVD-MDS algorithm, we have implemented different MD-MDS algorithms with stochastic energy. We used two types of temperature decrease, the first linear, beginning with a temperature of 100 J and decreasing linearly to 0 J during 3000 steps of simulation; we call this method MD-MDS linear. The second includes an exponential decrease from 100 J to below 0.1 J during 3000 steps of simulation; we call this method MD-MDS exponential. The function $s(t)$ uses random numbers generated uniformly between -0.5 and 0.5 .

As seen in Table 3, SVD-MDS as well as the two MD-MDS algorithms ‘linear’ and ‘exponential’ always identify final configurations with the same amount of residual energy. It can also be seen that SVD-MDS converges faster for these four examples than the MD-MDS methods. In conclusion, the two MD-MDS

Table 3. Results from SVD–MDS, iSVD–MDS and both MD–MDS algorithms applied to the various datasets (cf. Table 1)

| ID | Dataset Name | Metric | SVD–MDS | | iMDS | | MDMDSlinear | | MDMDSexpo | |
|-----|---------------------|--------|---------|------|--------|--------|-------------|------|-----------|------|
| | | | e | t | e | t | e | t | e | t |
| d1 | 96Cell | R^2 | 0.3409 | 2500 | 0.3381 | 232097 | 0.3453 | 5500 | 0.3421 | 2500 |
| d2 | 96Cell_T | Cov | 0.1401 | 4500 | 0.1494 | 92536 | 0.1465 | 4500 | 0.1542 | 4500 |
| d3 | Iris | Cov | 0.0344 | 509 | 0.0344 | 3008 | 0.0359 | 4500 | 0.0343 | 4000 |
| d4 | Wine | Cov | 0.0010 | 0 | 9.0E-4 | 10003 | 0.0089 | 4500 | 0.0067 | 4500 |
| d5 | Stochast 200 | Cov | 0.4088 | 1500 | – | – | 0.4092 | 4500 | 0.4089 | 4500 |
| d6 | CCYier | Cov | 0.0765 | 400 | 0.0753 | 22508 | 0.1346 | 5500 | 0.1162 | 5500 |
| d7 | Pima | Cov | 0.0708 | 700 | 0.0692 | 27005 | 0.1128 | 5500 | 0.0986 | 5500 |
| d8 | 96Cell_T transposed | R^2 | 0.1498 | 4500 | 0.1525 | 122059 | 0.1832 | 4224 | 0.1822 | 4500 |
| d9 | Secom | Cov | 0.1168 | 750 | – | – | 0.1511 | 5500 | 0.1396 | 4500 |
| d10 | Ozone | Cov | 0.0935 | 712 | 0.0935 | 66031 | 0.0944 | 4500 | 0.0951 | 3500 |
| d11 | Stochast 3000 | Cov | 0.4353 | 130 | – | – | 0.4353 | 200 | 0.4353 | 200 |
| d12 | Ecoli | Cov | 0.0 | 0 | – | – | 0.312 | 5500 | 0.2273 | 5500 |
| d13 | Wave | Cov | 0.2132 | 324 | – | – | 0.2132 | 3671 | 0.2132 | 2203 |

CoV, covariance; R^2 , correlation; e , Kruskal stress; t , time steps for MD simulation.

algorithms do not improve MDS, on the contrary they converge slower.

We next asked whether or not similarly adding stochasticity to the SVD–MDS algorithm would improve its performance. Figure 3C and D illustrates that indeed adding different amounts of energy at different times of the simulation (arrows) does not lead to lower energy minima. The SVD–MDS algorithm, similarly as the MD–MDS algorithms (Table 3), always converges to the same energy state. This has also been confirmed using other datasets (data not shown). Taken together, the results using MD–MDS-lin and MD–MDS-exp and SVD–MDS strongly suggest that only a single ground state is present. While we do not have any formal proof, we believe that the detailed analysis of the geometric structure of the data objects presented below also strongly argues in favor of a global energy minimum.

3.4 Geometric structure

Kruskal stress directly evaluates the distance information deformation. Graef *et al.* demonstrated in 1979 (Graef and Spence, 1979), that it rather evaluates global deformation of the cloud of instances. To gain information on local distances deformation, we define a new parameter, *Entourage*. For any one instance \tilde{X}_i in the reference distribution obtained through SVD (undistorted representation), we consider its k nearest neighbors: N_i^{ref} . In the new distribution obtained after dimensionality reduction, we also compute the k nearest neighbors for the same instance \tilde{X}_i , and obtain a list: N_i^{new} . We then search for $G_i = \text{card}(N_i^{\text{ref}} \cap N_i^{\text{new}})$, which will be the number of instances common to both. This operation is repeated for all instances i , and one obtains the *Entourage* parameter:

$$Ent_k = \frac{\sum_{i=1}^n G_i}{G} \quad (9)$$

with $G = nk$ a normalization parameter ($Ent \in (0, 1)$).

If $G_i = \text{card}(N_i^{\text{ref}} \cap N_i^{\text{new}}) \approx 0.01 \text{card}(N_i^{\text{ref}}) = 0.01k$ for every i then $Ent_k \approx \frac{0.01 \sum_{i=1}^n k}{nk} = 0.01$, a difference of 1% between two values of *Entourage* corresponds to an average deformation of 1% in the local organization. This parameter has more signification for

a small number of neighbors k compared to the total number of points n .

The geometric properties of the data objects are analyzed using the *Entourage* parameter. We have plotted the relationship of *Entourage* and k for six different methodologies: zeroMDS, stochastMDS, SVD–MDS, iSVD–MDS, MD–MDS-lin, MD–MDS-exp in Figure 4 for eight different datasets. From the selected examples, it becomes clear that again the SVD–MDS method outperforms the different types of MDS over a wide array of structures analyzed as the *Entourage* value is consistently higher no matter how many different k nearest neighbors are considered. The iterative iSVD–MDS method, due to the accumulation of small residual errors during the molecular dynamics simulation, and the MDS method give similar results. At the cost of increasing computational load, the iSVD–MDS better and better approximates the SVD–MDS method. In conclusion, the SVD–MDS method, under all conditions tested, better represents the geometric structure of the datasets in low-dimensional space when compared to the input object with $\text{rank}(S)$ components (given by SVD). Note that this holds even for objects with equal stress.

Figure 1 illustrates the problem of rotational variance when using stochastically initiated molecular dynamics simulations for MDS. When comparing Figure 1C and D as well as comparing them to Figure 1A and B that stochastMDS results produces near-optimal solutions (with respect to the Kruskal stress), the resulting orientation of the instances, however, is different (focus, for instance, on the relationship between ‘skeletal muscle’ and ‘fetal liver’). SVD–MDS on the contrary only produces a single result. This observation, taken together with the results on the relevance of stochasticity in the simulation obtained above, argues for the existence of different equivalent energy minima that only differ in the rotational orientation of the object and (at best) only minimally in the Kruskal-stress; a fact predicted by mathematical consideration. Hence, SVD–MDS not only reduces significantly the computational load, but also insures uniqueness of the resulting representation. The quality of this final and unique representation can be demonstrated using the *Entourage* parameter. The increase in fidelity in the representation of data should not be underestimated (see also Fig. 5).

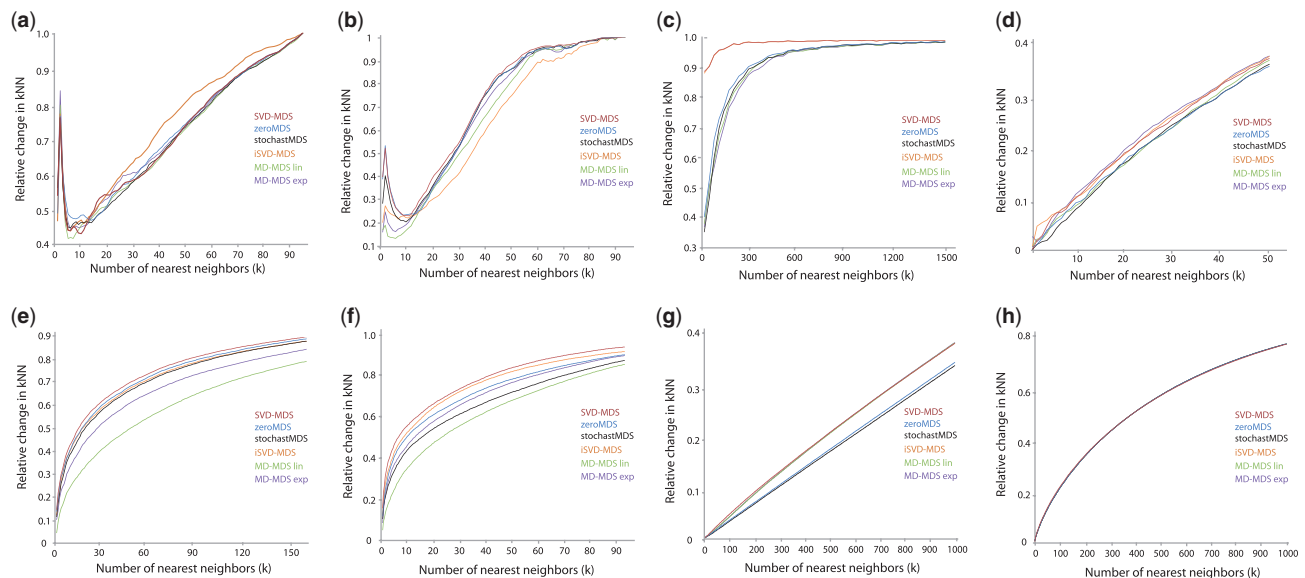


Fig. 4. Relative changes in k nearest neighbors (*Entourage*) are local, structural measures of dimensionality reduction and thus assess quality of the procedure. As a function of the number of nearest neighbors k considered, the relative change in kNN between the initial high-dimensional space and 2D space is plotted for the methods: SVD-MDS, zeroMDS, stochasticMDS, iSVD-MDS, MD-MDS linear and MD-MDS exponential. The datasets used are in: (a) 'd1—96Cell' in correlation basis, (b) 'd2—96Cell_T' in covariance basis, (c) 'd4—Wine' in covariance basis, (d) 'd5—Stochast 200' in covariance basis, (e) 'd6—CCYier' in covariance basis, (f) 'd7—Pima' in covariance basis, (g) 'd11—Stochast 3000' in covariance basis and (h) 'd13—Wave' in covariance basis.

This is reminiscent to techniques of principal manifold searches (Gorban *et al.*, 2008) where parameters describing topology, local organization or other geometric characteristics are used.

A major advantage of using SVD to define the initial state is that it provides the inertia of each principal component. The comparison of the different internal structures of the studied datasets showed a vast variety of profiles. A good dimensionality reduction technique would ideally account for these differences. Taking into account the inertia, the stress and the *Entourage* during the MDS process will help to have an even more accurate representation of the data matrix in low-dimensional space.

3.5 Data analysis

In order to demonstrate the applicability of the SVD-MDS methodology and its superior performance, we reanalyzed a previously published biological dataset not yet used here (Prakash *et al.*, 2006). The dataset consists of quantitative measures for 10 selected cytokines in a cohort of human malaria patients from central India displaying different severeness of disease as well as endemic and non-endemic control subjects. A total of 98 patients were included in the original study by Prakash *et al.* (2006). The main objective is to determine whether individual or combinations of cytokine measurements can be used to determine whether an individual is affected by cerebral malaria (CM), the most severe form of the disease, and how to distinguish CM from severe malaria (SM). Both forms of the disease require early detection and prognosis which are pressing matters for health caretakers. We have computed from the entire dataset [including the controls and patients with mild malaria (MM)] SVD-based and SVD-MDS-based representations of the cytokine activity measurements in covariance space (Fig. 5A and B). It becomes immediate evident

that whereas the representation by SVD-MDS identifies TNF α as having a major contribution to one of the higher principal components, SVD alone does not reveal this prominent role for TNF α leading to the conclusion that the main variability in the patient samples is due to IL2, IL6 and TGF β [Fig. 5C as opposed to 5D (SVD-MDS)]. The combination of IL2 and TNF α measurements alone suffices, however, to separate SM (red) from CM (blue) patients in single linkage hierarchical clustering based on Euclidean distances (Fig. 5E). The combination of IL2 and TNF α would unlikely have been identified as effective by SVD alone (Fig. 5A). The role of TNF α in CM has been also clarified when investigating the auto-immune component of CM in Bansal *et al.* (2009).

4 CONCLUSION

Dimensionality reduction of complex, high-dimensional data is an important problem which becomes ever more complicated due to the increase of data concomitant with an increase in their dimensionality. This is particularly true for data from modern genomics analyses where more and more data with thousands of instances each over millions of variables are generated. We demonstrate here how a combined molecular dynamics simulation multidimensional scaling approach for dimensionality reduction of high-dimensional data can be improved by better defining the initial conditions. We have shown that singular value decomposition is most effective to create an initial condition for MDS. Using links between SVD and different standard data analysis methods, we demonstrate how our combined SVD-MDS method can be used to improve geometric representation in low-dimensional space that are generally obtained with standard analysis methods (PCA, classical scaling, PCCA, correspondence analysis). We also show that the use of iterative reduction or

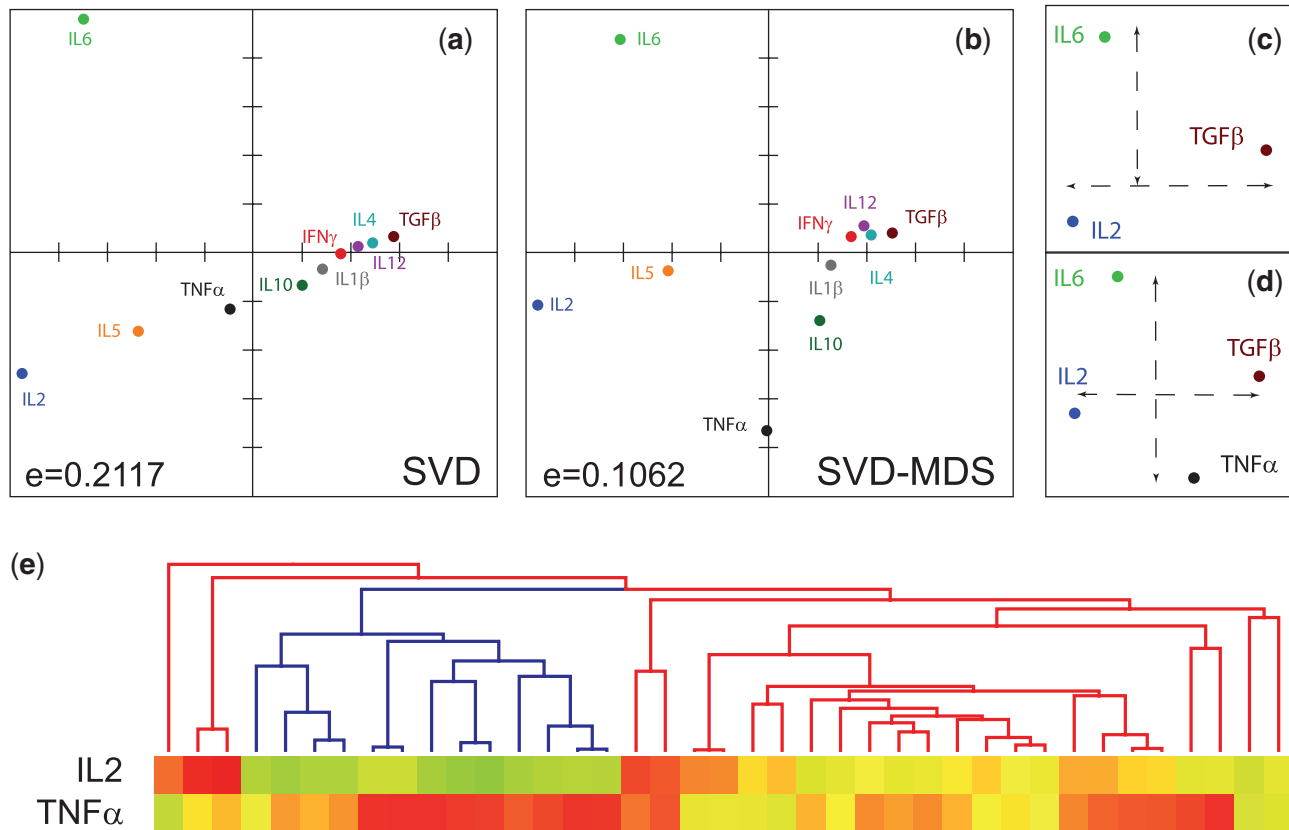


Fig. 5. Comparative analysis of cytokine activity measurements in an Indian malaria human patient cohort. The cytokine dataset from Prakash *et al.* (2006) was represented in covariance space using SVD (a) and SVD-MDS (b). A simplified representation for SVD and SVD-MDS is shown in (c) and (d), respectively. (e) Single linkage hierarchical clustering based on Euclidean distance of the severe malaria (SM, red) and cerebral malaria (CM, blue) patients according to IL2 and TNFα activity.

stochastic energy does not increase performance of the algorithms in terms of finding an optimal solution. Finally, we have investigated the local structure deformation induced by dimensionality reduction, and confirmed the superior accuracy of the SVD-MDS. Overall, the methodology developed here should further advance our capacity to analyze high-dimensional data such as the ones produced by functional genomics approaches.

Funding: Centre National de la Recherche Scientifique (CNRS); the *Agence Nationale pour la Recherche contre le SIDA et les hépatites virales* (ANRS); the *Agence Nationale pour la Recherche* (ANR, ISPA project); the *Genopole Evry*. C.B. is recipient of a PhD. fellowship from the ANRS.

Conflict of Interest: none declared.

REFERENCES

- Alter, O. *et al.* (2000) Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl Acad. Sci. USA*, **97**, 10101–10106.
- Alter, O. *et al.* (2003) Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms. *Proc. Natl Acad. Sci. USA*, **100**, 3351–3356.
- Andrecut, M. (2009) Molecular dynamics multidimensional scaling. *Phys. Lett. A*, **373**, 2001–2006.
- Bansal, D. *et al.* (2009) IgG autoantibody to brain beta tubulin iii associated with cytokine cluster-ii discriminate cerebral malaria in central india. *PLoS One*, **4**, e8245.
- Benecke, A. (2008) Gene regulatory network inference using out of equilibrium statistical mechanics. *HFSP J.*, **2**, 183–188.
- Berthold, M. and Hand, D. (2003). *Intelligent Data Analysis*, 2nd edn., Springer, Heidelberg/Berlin.
- Brock, G. *et al.* (2008) Which missing value imputation method to use in expression profiles: a comparative study and two selection schemes. *BMC Bioinformatics*, **9**, 12.
- Candes, E. and Recht, B. (2008) Exact matrix completion via convex optimization; eprint arXiv:0805.4471.
- Cox, T. and Cox, M. (2000) *Multidimensional Scaling*, 2nd edn., Chapman & Hall/CRC Press, Boca Raton, USA.
- Cuadras, C.M. and Fortiana, J. (1995) Metric scaling graphical representation of categorical data. *Penn State University*.
- Dezso, Z. *et al.* (2008) A comprehensive functional analysis of tissue specificity of human gene expression. *BMC Biol.*, **6**, 49.
- Ding, C. and He, X. (2004) K-means clustering via principal component analysis. In *Proceedings of the 21 st International Conference on Machine Learning*. ACM Press, NY, USA, pp. 225–232.
- Eckart, C. and Young, G. (1936) The approximation of one matrix by another of lower rank. *Psychometrika*, **1**, 211–218.
- Fellenberg, K. *et al.* (2001) Correspondence analysis applied to microarray data. *Proc. Natl Acad. Sci. USA*, **98**, 10781.
- Frank, A. and Asuncion, A. (2010) *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences. Available at <http://archive.ics.uci.edu/ml>.
- Gorban, N. *et al.* (2008) *Principal Manifolds for Data Visualization and Dimension Reduction*. Springer Publishing Company, Heidelberg, Germany.
- Graef, J. and Spence, I. (1979) Using distance information in the design of large multidimensional scaling experiments. *Psychol. Bull.*, **86**, 60–66.

- Holmes, S. (2006) Visualising data. In Lyons, L. and Unel, M.K. (eds) *Proceedings of Statistical Problems in Particle Physics, Astrophysics and Cosmology*, Imperial College Press, World Scientific Publishing Co., p. 197.
- Iyer, V. *et al.* (1999) The transcriptional program in the response of human fibroblasts to serum. *Science*, **283**, 83–87.
- Kruskal, J. and Wish, M. (1978) *Multidimensional Scaling*. SAGE Publications Inc., Thousand Oaks, CA, USA.
- Noth, S. *et al.* (2006) High-sensitivity transcriptome data structure and implications for analysis and biologic interpretation. *Genomics Proteomics Bioinformatics*, **4**, 212–229.
- Prakash, D. *et al.* (2006) Clusters of cytokines determine malaria severity in plasmodium falciparum-infected patients from endemic areas of central india. *J. Infect. Dis.*, **194**, 198–207.
- Schmidt, E. and Stewart, G. (1992) On the early history of the singular value decomposition. *Univeristy of Maryland*.
- Torgerson, W. (1952) Multidimensional scaling: I. theory and method. *Psychometrika*, **17**, 401–419.
- Troyanskaya, O. *et al.* (2001) Missing value estimation methods for DNA microarrays. *Bioinformatics*, **17**, 520–525.
- Wall, M. *et al.* (2003) Singular value decomposition and principal component analysis. In *A Practical Approach to Microarray Data Analysis*. Springer, Berlin/Heidelberg, Germany, pp. 91–109.