

# Stochastic Proximity Embedding

DIMITRIS K. AGRAFIOTIS

3-Dimensional Pharmaceuticals, Inc., 665 Stockton Drive, Exton, Pennsylvania 19341

Received 13 September 2002; Accepted 5 November 2002

**Abstract:** We introduce stochastic proximity embedding (SPE), a novel self-organizing algorithm for producing meaningful underlying dimensions from proximity data. SPE attempts to generate low-dimensional Euclidean embeddings that best preserve the similarities between a set of related observations. The method starts with an initial configuration, and iteratively refines it by repeatedly selecting pairs of objects at random, and adjusting their coordinates so that their distances on the map match more closely their respective proximities. The magnitude of these adjustments is controlled by a learning rate parameter, which decreases during the course of the simulation to avoid oscillatory behavior. Unlike classical multidimensional scaling (MDS) and nonlinear mapping (NLM), SPE scales linearly with respect to sample size, and can be applied to very large data sets that are intractable by conventional embedding procedures. The method is programmatically simple, robust, and convergent, and can be applied to a wide range of scientific problems involving exploratory data analysis and visualization.

© 2003 Wiley Periodicals, Inc. J Comput Chem 24: 1215–1221, 2003

**Key words:** stochastic proximity embedding; multidimensional scaling; nonlinear mapping; Sammon mapping; stochastic descent; self-organizing; dimensionality reduction; feature extraction; combinatorial chemistry; data mining; data analysis; pattern recognition; molecular descriptor; molecular similarity; molecular diversity

## Introduction

Converting distances to coordinates is a pervasive theme in many scientific domains. A prototypical example comes from the field of cartography—given a matrix of intercity distances such as those commonly found in the back page of a road atlas, the objective is to place the cities on a two-dimensional map that reflects their true geographical coordinates. Stated more generally, the problem is to arrange a set of objects in a space with a particular number of dimensions so as to reproduce the observed distances between them. This problem is known as multidimensional scaling (MDS)<sup>1</sup> or nonlinear mapping (NLM),<sup>2</sup> and has two primary applications: (1) reducing the dimensionality of high-dimensional observations, and (2) producing coordinate vectors from data supplied directly in the form of proximities, so that they can be analyzed with conventional statistical and data mining techniques. In addition to data compression and computational efficiency, these methods offer the ability to “explain” the observed proximities in terms of a few underlying dimensions, and thus more effectively reason about the data. Despite decades of intensive development, MDS remains a formidable problem that is considered intractable for large data sets.

More specifically, given a set of  $k$  objects, a symmetric matrix,  $r_{ij}$ , of relationships between these objects, and a set of images on a  $m$ -dimensional display plane  $\{\mathbf{x}_i, i = 1, 2, \dots, k; \mathbf{x}_i \in \mathfrak{R}^m\}$ , the

problem is to place  $\mathbf{x}_i$  onto the plane in such a way that their Euclidean distances  $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$  approximate as closely as possible the corresponding values  $r_{ij}$ . Although an exact projection is only possible when the metric matrix is positive semidefinite, meaningful projections can be obtained even when this criterion is not satisfied. The quality of the projection is determined using a sum-of-squares error function such as Kruskal’s stress:

$$S = \sqrt{\sum_{i<j} (d_{ij} - r_{ij})^2 / \sum_{i<j} d_{ij}^2} \quad (1)$$

or Sammon’s stress:

$$S = \sum_{i<j} \frac{(d_{ij} - r_{ij})^2}{r_{ij}} / \sum_{i<j} r_{ij} \quad (2)$$

which are numerically minimized in order to find the optimal configuration. The actual embedding is carried out in an iterative fashion by: (1) generating an initial set of coordinates  $\mathbf{x}_i$ , (2) computing the distances  $d_{ij}$ , (3) finding a new set of coordinates  $\mathbf{x}_i$  using a steepest descent algorithm, and (4) repeating steps 2 and 3

**Correspondence to:** D. K. Agrafiotis; e-mail: dimitris.agrafiotis@3dp.com

until the change in the stress function falls below some predefined threshold. There is a wide variety of MDS algorithms involving different loss functions and optimization heuristics, ranging from iterative majorization and Newton–Raphson minimization, to tabu search, simulated annealing, and genetic algorithms. The reader is referred to ref. 3 as well as the rich literature on a closely related technique known as distance geometry.<sup>4</sup>

Unfortunately, these algorithms are impractical for data sets containing more than a few thousand items because the computation of the stress function [eqs. (1) and (2), and their variants] increases quadratically with the number of data points. Several methods have been devised to reduce the complexity of the task, either by restricting attention to a subset of objects or to a subset of distances. For example, Chang and Lee<sup>5</sup> proposed a heuristic relaxation approach in which a subset of the original objects (the frame) are scaled using a Sammon-like methodology, and the remaining objects are then added to the map by adjusting their distances to the objects in the frame. An alternative approach proposed by Pykett<sup>6</sup> is to partition the data set into a set of disjoint clusters, and map only the cluster prototypes, i.e., the centroids of the pattern vectors in each class. In the resulting two-dimensional plots, the cluster prototypes are represented as circles whose radii are proportional to the spread in their respective classes. Lee<sup>7</sup> proposed a triangulation method that positions each point on the plane in a way that preserves the distances from its two nearest neighbors already mapped. This idea was later elaborated by Biswas et al.,<sup>8</sup> who introduced a hybrid approach that combined the ability of Sammon's algorithm to preserve global information with the efficiency of Lee's triangulation method. Although the triangulation can be computed quickly compared to conventional MDS methods, it preserves only a small fraction of distances, and the projection may be difficult to interpret for large data sets.

A very promising technique is to use conventional MDS or NLM to project a small random sample, and then “learn” the underlying nonlinear transform using a multilayer perceptron.<sup>9</sup> Once trained, the neural network can be used in a feed-forward manner to project the remaining members of the population as well as new, unseen patterns with minimal distortion. This general strategy was subsequently extended to employ local learning techniques,<sup>10</sup> generalized to handle complex distance functions and input data supplied in nonvectorial form,<sup>11</sup> and tailored to allow the scaling of combinatorial libraries in a way that circumvents their explicit enumeration.<sup>12</sup> Still, this method does not address the fundamental embedding problem, which is a prerequisite to parametric learning. The algorithm described herein addresses the key limitation of classical methods—namely, quadratic complexity with respect to sample size—and can be applied on a broad range of problems in scientific data analysis and visualization.

## Methods

### Algorithm

Stochastic proximity embedding uses a self-organizing scheme that attempts to bring each individual stress  $(d_{ij} - r_{ij})^2$  rapidly to zero. The method starts with an initial configuration and iteratively

refines it by repeatedly selecting two points at random, and adjusting their coordinates so that their Euclidean distance on the map  $d_{ij}$  matches more closely their corresponding proximity  $r_{ij}$ .

The correction is proportional to the disparity  $\lambda \frac{|r_{ij} - d_{ij}|}{d_{ij}}$ , where  $\lambda$  is a learning rate parameter that decreases during the course of the refinement to avoid oscillatory behavior. The detailed algorithm is as follows:

1. Initialize the coordinates  $\mathbf{x}_i$ . Select an initial learning rate  $\lambda$ .
2. Select a pair of points,  $i$  and  $j$ , at random and compute their distance  $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ . If  $d_{ij} \neq r_{ij}$ , update the coordinates  $\mathbf{x}_i$  and  $\mathbf{x}_j$  by:

$$\mathbf{x}_i \leftarrow \mathbf{x}_i + \lambda \frac{1}{2} \frac{r_{ij} - d_{ij}}{d_{ij} + \varepsilon} (\mathbf{x}_i - \mathbf{x}_j)$$

and

$$\mathbf{x}_j \leftarrow \mathbf{x}_j + \lambda \frac{1}{2} \frac{r_{ij} - d_{ij}}{d_{ij} + \varepsilon} (\mathbf{x}_j - \mathbf{x}_i)$$

where  $\varepsilon$  is a small number to avoid division by zero.

3. Repeat (2) for a prescribed number of steps  $S$ .
4. Decrease the learning rate  $\lambda$  by prescribed decrement  $\delta\lambda$ .
5. Repeat (2)–(4) for a prescribed number of cycles  $C$ .

### Data Sets

The algorithm was tested on three data sets from the computer graphics and combinatorial chemistry literature.

1. Phone—a public domain data set for modeling and rendering programs. The data set consists of 6070 points in  $R^3$  derived from the file “phone.vort” found in the public domain of *net.land*. The original data was stripped of all its connectivity information (edges and triangles), and the centroids of all triangles were added to generate additional points.<sup>13</sup>
2. Diamine library—a three-component virtual combinatorial library containing  $10^6$  compounds (*diamine6*) derived by combining 100 diamine cores with two sets of 100 alkylating agents using the reductive amination reaction. Each of the products was described by 117 topologic descriptors including molecular connectivity indices, kappa shape indices, subgraph counts, information-theoretic indices, Bonchev-Trinajstić indices, and topologic state indices.<sup>14</sup> To eliminate redundancy in the data, the topologic descriptors were normalized and decorrelated using principal component analysis (PCA). This process resulted in an orthogonal set of 26 latent variables, which accounted for 99% of the total variance in the data. Molecular dissimilarity was defined as the Euclidean distance in the 26-dimensional latent variable space. Two smaller libraries containing  $10^4$  (*diamine4*) and  $10^5$  (*diamine5*) compounds were derived by random sampling from *diamine6* to test the scalability of the algorithm.

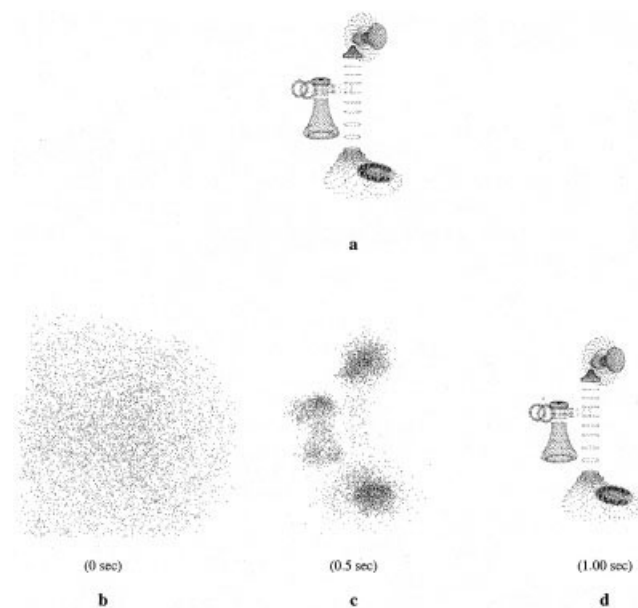
3. Ugi library—a four-component virtual combinatorial library containing  $10^6$  compounds (*ugi6*) derived by combining 100 carboxylic acids, 50 primary amines, 20 aldehydes, and 10 isonitriles using the Ugi reaction. Each of the products was described by a 166-dimensional binary fingerprint, where each bit encoded the presence or absence of a particular structural feature in the target molecule. The bit assignment was based on the fragment dictionary used in the ISIS chemical database management system. Molecular dissimilarity was based on the Tanimoto coefficient:

$$r_{ij} = 1 - |\text{AND}(a, b)| / |\text{IOR}(a, b)| \quad (5)$$

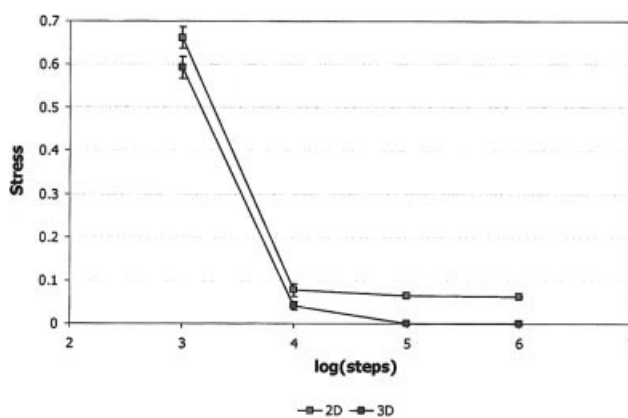
where *a* and *b* represent two binary encoded molecules, *AND* is the binary “and” operation (a bit in the result is set if both of the corresponding bits in the two operands are set), and *IOR* is the binary “inclusive or” operation (a bit in the result is set if either of the corresponding bits in the two operands are set). Two smaller libraries containing  $10^4$  (*ugi4*) and  $10^5$  (*ugi5*) compounds were derived by random sampling from *ugi6*.

#### Computational Details

All programs were implemented in the C++ programming language and are part of the DirectedDiversity® software suite.<sup>15</sup> All calculations were carried out on a Dell Inspiron 8000 laptop



**Figure 1.** Three-dimensional stochastic proximity embedding of the *phone* data set. The map was derived by computing the distance matrix of the 6070 points comprising the image, and using those distances as input to SPE. (a) Original data points. (b) Initial random configuration. (c) Intermediate configuration obtained after 50 refinement cycles. (d) Final configuration obtained after 100 refinement cycles. The CPU times required are shown in parentheses.



**Figure 2.** Mean and standard deviation of the stress obtained after 30 independent two- and three-dimensional SPE embeddings of the *phone* data set. Each embedding was carried out using 100 cycles and a linear learning schedule from  $\lambda_{\max} = 1$  to  $\lambda_{\min} = 0.01$ , and was based on a different initial configuration and random number seed.

computer equipped with a 733 MHz Pentium III Intel processor running Windows 2000 Professional.

#### Results and Discussion

A minimum requirement for a nonlinear mapping algorithm is to be able to rapidly and reliably discover the true configuration when the data is embedded in a space of the intrinsic dimension. As shown in Figures 1 and 2, SPE performs extremely well in this regard, as it is able to reconstruct the original three-dimensional coordinates of the *phone* data set (or its mirror image) given only the interpoint distances, regardless of the starting configuration and initialization conditions. As illustrated in Figure 1b–d, SPE recovers the general structure of the data within a few iterations and spends the remaining cycles perfecting the coordinates to eliminate any residual stress. More importantly, even though there are over 18 million pairwise distances in the original sample, 10 million pairwise refinements (100 cycles  $\times$  100,000 steps) are sufficient to reproduce the original structure to perfection. Similar convergence is observed even when the data is embedded in fewer dimensions, where the pairwise distances cannot all be simultaneously satisfied and some information is inevitably lost upon projection (the mean stress of a two-dimensional embedding over 30 independent runs was  $0.064 \pm 0.00009$ ). The number of refinement steps required to reach a good solution appears to depend solely on the number of data points and be insensitive to the intrinsic structure of the data or the number of embedding dimensions (Figs. 5 and 6).

To ensure a smooth and efficient refinement, the learning rate must be prudently chosen (see Tables 1–4). If  $\lambda$  is too small, self-organization will be smooth but will converge slowly. If  $\lambda$  is too large, convergence will be fast but the map may become unstable and oscillatory. The latter is particularly true when the system is embedded in fewer than the intrinsic dimensions and the configuration will inevitably have some residual stress. Figure 3

**Table 1.** Stress of 2D SPE Maps for the Diamine Data Sets, Derived Using a Linearly Decreasing Learning Rate from  $\lambda_{\max}$  to  $\lambda_{\min}$ .

Size <sup>a</sup>	Dims <sup>b</sup>	Cycles	Steps	$\lambda_{\max}$	$\lambda_{\min}$	$\mu(S)^c$	$\sigma(S)^d$
10 <sup>4</sup>	2	100	10 <sup>3</sup>	1	0.001	0.4080	0.0015
10 <sup>4</sup>	2	100	10 <sup>4</sup>	1	0.001	0.3052	0.0264
10 <sup>4</sup>	2	100	10 <sup>5</sup>	1	0.001	0.2137	0.0014
10 <sup>4</sup>	2	100	10 <sup>6</sup>	1	0.001	0.2069	0.0004
10 <sup>4</sup>	2	100	10 <sup>3</sup>	1	0.5	0.4060	0.0012
10 <sup>4</sup>	2	100	10 <sup>4</sup>	1	0.5	0.2790	0.0063
10 <sup>4</sup>	2	100	10 <sup>5</sup>	1	0.5	0.2371	0.0012
10 <sup>4</sup>	2	100	10 <sup>6</sup>	1	0.5	0.2352	0.0005
10 <sup>4</sup>	2	100	10 <sup>3</sup>	1	1	0.4220	0.0016
10 <sup>4</sup>	2	100	10 <sup>4</sup>	1	1	0.3096	0.0024
10 <sup>4</sup>	2	100	10 <sup>5</sup>	1	1	0.2855	0.0014
10 <sup>4</sup>	2	100	10 <sup>6</sup>	1	1	0.2846	0.0008
10 <sup>5</sup>	2	100	10 <sup>3</sup>	1	0.001	0.9622	0.0029
10 <sup>5</sup>	2	100	10 <sup>4</sup>	1	0.001	0.4059	0.0005
10 <sup>5</sup>	2	100	10 <sup>5</sup>	1	0.001	0.3516	0.0192
10 <sup>5</sup>	2	100	10 <sup>6</sup>	1	0.001	0.2165	0.0028
10 <sup>5</sup>	2	100	10 <sup>3</sup>	1	0.5	0.8051	0.0019
10 <sup>5</sup>	2	100	10 <sup>4</sup>	1	0.5	0.4040	0.0005
10 <sup>5</sup>	2	100	10 <sup>5</sup>	1	0.5	0.3061	0.0232
10 <sup>5</sup>	2	100	10 <sup>6</sup>	1	0.5	0.2394	0.0015
10 <sup>5</sup>	2	100	10 <sup>3</sup>	1	1	0.6960	0.0021
10 <sup>5</sup>	2	100	10 <sup>4</sup>	1	1	0.4204	0.0007
10 <sup>5</sup>	2	100	10 <sup>5</sup>	1	1	0.3183	0.0077
10 <sup>5</sup>	2	100	10 <sup>6</sup>	1	1	0.2875	0.0019
10 <sup>6</sup>	2	100	10 <sup>3</sup>	1	0.001	1.4821	0.0016
10 <sup>6</sup>	2	100	10 <sup>4</sup>	1	0.001	0.9601	0.0009
10 <sup>6</sup>	2	100	10 <sup>5</sup>	1	0.001	0.4070	0.0004
10 <sup>6</sup>	2	100	10 <sup>6</sup>	1	0.001	0.3740	0.0040
10 <sup>6</sup>	2	100	10 <sup>7</sup>	1	0.001	0.2175	0.0014

<sup>a</sup>Number of compounds in the combinatorial library.<sup>b</sup>Dimensionality of the nonlinear map.<sup>c</sup>Mean stress over 30 independent mapping trials.<sup>d</sup>Standard deviation of stress over 30 independent mapping trials.

illustrates the problem in the simple case of a tetrahedral atom embedded in two dimensions—whereas a variable learning rate ensures that the error is distributed evenly across the entire point set, a constant learning rate of 1 puts undue emphasis on the last coordinate updates, resulting in a map with substantially higher stress (0.236 vs. 0.152 for  $\lambda_{\max} = \lambda_{\min} = 1$ ). The best results are obtained when the learning rate is gradually reduced from a relatively large ( $\lambda_{\max} \approx 1$ ) to a small ( $\lambda_{\min} \approx 0.01$ ) value during the course of the refinement—this ensures rapid initial organization followed by smooth convergence to a low-stress minimum. We have found that the performance of the algorithm is relatively insensitive to the learning schedule, and advocate the use of a linearly decreasing learning rate with a constant rate decrement  $\delta\lambda$ . Because computing the exact stress [eq. (1)] is prohibitive for data sets exceeding a few thousand items, the values reported in Tables 1–4 were calculated by random sampling of 1,000,000 pairwise distances. As shown in Figure 4, these stochastic stress values converge extremely rapidly, and 10<sup>5</sup>–10<sup>6</sup> distances suffice to approximate accurately the true stress (10<sup>6</sup> random distances pro-

**Table 2.** Stress of 2D SPE Maps for the Ugi Data Sets, Derived Using a Linearly Decreasing Learning Rate from  $\lambda_{\max}$  to  $\lambda_{\min}$ .

Size <sup>a</sup>	Dims <sup>b</sup>	Cycles	Steps	$\lambda_{\max}$	$\lambda_{\min}$	$\mu(S)^c$	$\sigma(S)^d$
10 <sup>4</sup>	2	100	10 <sup>3</sup>	1	0.001	0.4712	0.0009
10 <sup>4</sup>	2	100	10 <sup>4</sup>	1	0.001	0.4388	0.0076
10 <sup>4</sup>	2	100	10 <sup>5</sup>	1	0.001	0.3166	0.0004
10 <sup>4</sup>	2	100	10 <sup>6</sup>	1	0.001	0.3124	0.0003
10 <sup>4</sup>	2	100	10 <sup>3</sup>	1	0.5	0.4678	0.0010
10 <sup>4</sup>	2	100	10 <sup>4</sup>	1	0.5	0.4279	0.0149
10 <sup>4</sup>	2	100	10 <sup>5</sup>	1	0.5	0.3486	0.0005
10 <sup>4</sup>	2	100	10 <sup>6</sup>	1	0.5	0.3478	0.0006
10 <sup>4</sup>	2	100	10 <sup>3</sup>	1	1	0.4720	0.0010
10 <sup>4</sup>	2	100	10 <sup>4</sup>	1	1	0.4467	0.0091
10 <sup>4</sup>	2	100	10 <sup>5</sup>	1	1	0.4044	0.0013
10 <sup>4</sup>	2	100	10 <sup>6</sup>	1	1	0.4044	0.0012
10 <sup>5</sup>	2	100	10 <sup>3</sup>	1	0.001	0.4769	0.0003
10 <sup>5</sup>	2	100	10 <sup>4</sup>	1	0.001	0.4713	0.0005
10 <sup>5</sup>	2	100	10 <sup>5</sup>	1	0.001	0.4501	0.0010
10 <sup>5</sup>	2	100	10 <sup>6</sup>	1	0.001	0.3151	0.0003
10 <sup>5</sup>	2	100	10 <sup>3</sup>	1	0.5	0.4759	0.0003
10 <sup>5</sup>	2	100	10 <sup>4</sup>	1	0.5	0.4682	0.0005
10 <sup>5</sup>	2	100	10 <sup>5</sup>	1	0.5	0.4531	0.0073
10 <sup>5</sup>	2	100	10 <sup>6</sup>	1	0.5	0.3470	0.0003
10 <sup>5</sup>	2	100	10 <sup>3</sup>	1	1	0.4787	0.0004
10 <sup>5</sup>	2	100	10 <sup>4</sup>	1	1	0.4723	0.0004
10 <sup>5</sup>	2	100	10 <sup>5</sup>	1	1	0.4662	0.0048
10 <sup>5</sup>	2	100	10 <sup>6</sup>	1	1	0.4033	0.0006
10 <sup>6</sup>	2	100	10 <sup>3</sup>	1	0.001	0.4914	0.0002
10 <sup>6</sup>	2	100	10 <sup>4</sup>	1	0.001	0.4769	0.0003
10 <sup>6</sup>	2	100	10 <sup>5</sup>	1	0.001	0.4716	0.0004
10 <sup>6</sup>	2	100	10 <sup>6</sup>	1	0.001	0.4513	0.0004
10 <sup>6</sup>	2	100	10 <sup>7</sup>	1	0.001	0.3153	0.0003

<sup>a</sup>Number of compounds in the combinatorial library.<sup>b</sup>Dimensionality of the nonlinear map.<sup>c</sup>Mean stress over 30 independent mapping trials.<sup>d</sup>Standard deviation of stress over 30 independent mapping trials.**Table 3.** Stress of 5D SPE Maps for the Diamine Data Sets, Derived Using a Linearly Decreasing Learning Rate from  $\lambda_{\max}$  to  $\lambda_{\min}$ .

Size <sup>a</sup>	Dims <sup>b</sup>	Cycles	Steps	$\lambda_{\max}$	$\lambda_{\min}$	$\mu(S)^c$	$\sigma(S)^d$
10 <sup>4</sup>	5	100	10 <sup>3</sup>	1	0.001	0.2899	0.0005
10 <sup>4</sup>	5	100	10 <sup>4</sup>	1	0.001	0.1693	0.0034
10 <sup>4</sup>	5	100	10 <sup>5</sup>	1	0.001	0.0811	0.0008
10 <sup>4</sup>	5	100	10 <sup>6</sup>	1	0.001	0.0773	0.0003
10 <sup>5</sup>	5	100	10 <sup>3</sup>	1	0.001	0.5469	0.0015
10 <sup>5</sup>	5	100	10 <sup>4</sup>	1	0.001	0.2884	0.0003
10 <sup>5</sup>	5	100	10 <sup>4</sup>	1	0.001	0.1825	0.0042
10 <sup>5</sup>	5	100	10 <sup>6</sup>	1	0.001	0.0838	0.0012

<sup>a</sup>Number of compounds in the combinatorial library.<sup>b</sup>Dimensionality of the nonlinear map.<sup>c</sup>Mean stress over 30 independent mapping trials.<sup>d</sup>Standard deviation of stress over 30 independent mapping trials.

**Table 4.** Stress of 5D SPE Maps for the Ugi Data Sets, Derived Using a Linearly Decreasing Learning Rate from  $\lambda_{\max}$  to  $\lambda_{\min}$ .

Size <sup>a</sup>	Dims <sup>b</sup>	Cycles	Steps	$\lambda_{\max}$	$\lambda_{\min}$	$\mu(S)^c$	$\sigma(S)^d$
10 <sup>4</sup>	5	100	10 <sup>3</sup>	1	0.001	0.3124	0.0003
10 <sup>4</sup>	5	100	10 <sup>4</sup>	1	0.001	0.2540	0.0092
10 <sup>4</sup>	5	100	10 <sup>5</sup>	1	0.001	0.1358	0.0005
10 <sup>4</sup>	5	100	10 <sup>6</sup>	1	0.001	0.1332	0.0001
10 <sup>5</sup>	5	100	10 <sup>3</sup>	1	0.001	0.5069	0.0004
10 <sup>5</sup>	5	100	10 <sup>4</sup>	1	0.001	0.3128	0.0003
10 <sup>5</sup>	5	100	10 <sup>5</sup>	1	0.001	0.2834	0.0055
10 <sup>5</sup>	5	100	10 <sup>6</sup>	1	0.001	0.1356	0.0008

<sup>a</sup>Number of compounds in the combinatorial library.

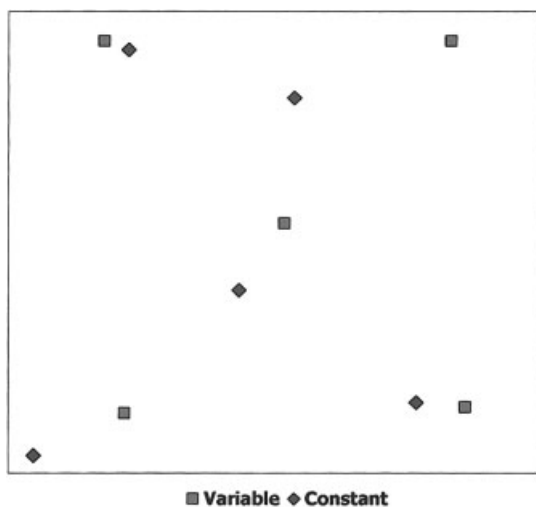
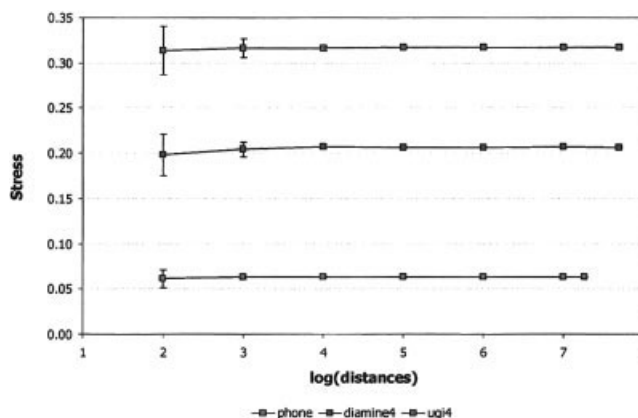
<sup>b</sup>Dimensionality of the nonlinear map.

<sup>c</sup>Mean stress over 30 independent mapping trials.

<sup>d</sup>Standard deviation of stress over 30 independent mapping trials.

duced a stress with a standard deviation of 0.0001, 0.0002, 0.0003, 0.0003, and 0.0002 for the *phone*, *diamine4*, *ugi4*, *diamine6*, and *ugi6* 2-D maps, respectively; for the first three, the mean stochastic stress differed from the true stress in the fifth decimal place—note that these standard deviations refer to the stress of a single representative embedding, and are not to be confused to the values reported in Tables 1 and 2).

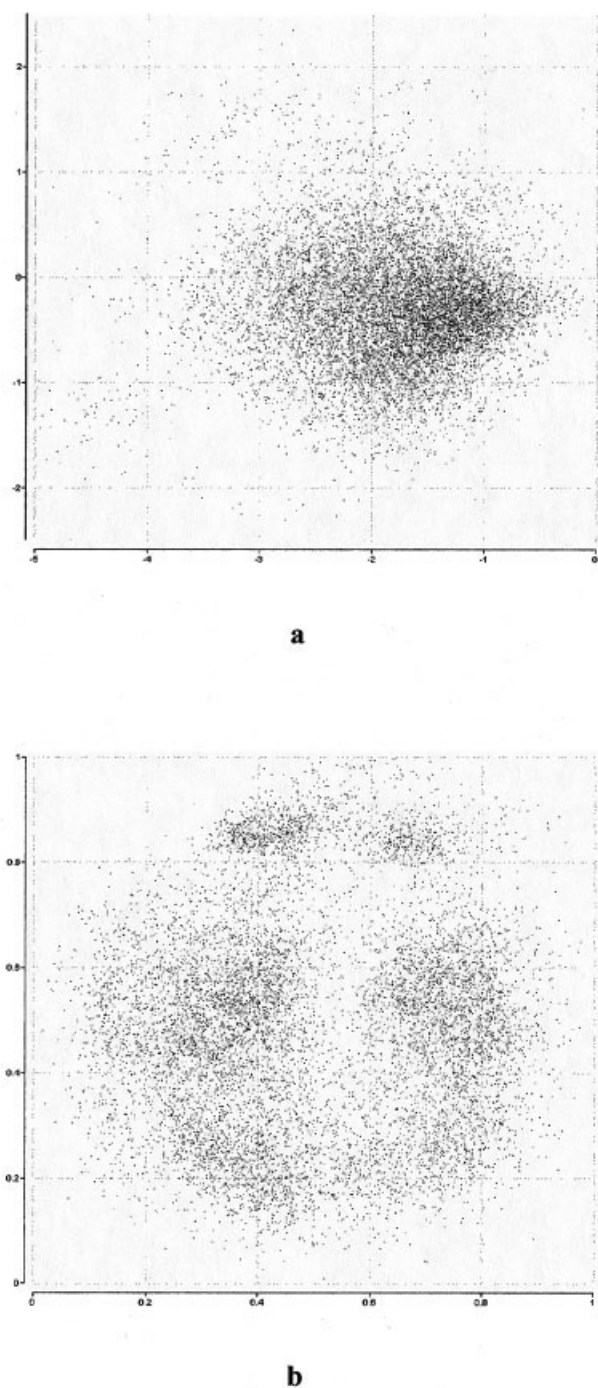
We postulate that the power of SPE stems from the stochastic nature of the refinement scheme and the vast redundancy of the distance matrix. Indeed, the algorithm is reminiscent of the stochastic approximation approach introduced by Robbins and Monroe<sup>16</sup> and popularized by Rumelhart's back-propagation algorithm for training multilayer perceptrons.<sup>17</sup> The direction of each pairwise refinement can be thought of as an instantaneous gradient—a stochastic approximation of the true gradient of the

**Figure 3.** Two-dimensional SPE maps of methane derived using (1) a constant learning rate of 1 (blue diamonds), and (2) a linearly decreasing learning rate from 1 to 0.01 (red squares).**Figure 4.** Mean and standard deviation of the stochastic stress of a typical two-dimensional SPE map of the *phone*, *diamine4*, and *ugi4* data sets as a function of the number of distances sampled. The right-most point in each series represents the exact stress.

stress function. For sufficiently small values of  $\lambda$ , the average direction of these refinements approximates the direction of steepest descent. Unlike classical gradient minimization schemes, the use of stochastic gradients changes the effective error function in each step, and the method becomes less susceptible to local minima.

The second advantage stems from the fact that the distance matrix is highly overdetermined. It is well known that the relative configuration of  $N$  points in a  $D$ -dimensional space can be fully described using only  $(N - D/2 - 1)/(D + 1)$  distances. SPE exploits this redundancy through random sampling. The random pairwise refinements are highly cooperative—moving one pair of points towards their target distance simultaneously improves many other distances involving these points. This results in a compelling advantage over kindred techniques—linear scaling with respect to the number of objects in the data set. As shown in Figure 6 and 7, when SPE is applied to data sets of increasing size drawn from the same probability distribution (and therefore expected to have comparable stress), the number of sampling steps required to achieve a particular stress value increases in linear fashion. Similarly, the memory requirements of the algorithm grow linearly as well, because the distances (and the original proximities) can be computed “on the fly” and need not be explicitly stored. Linear scaling in both time and memory is critical in modern data mining where large data sets abound. These results also suggest that to obtain a good embedding, the total number of pairwise refinements should be approximately 1000 times the number of objects in the data sample. This effort can be further reduced by using a sensible box size and/or a partially organized initial configuration, such as a principal component projection.

Just like classical MDS and NLM, SPE produces maps that exhibit meaningful structure even when the data is embedded in fewer than the intrinsic dimensions. Interactive inspection of the combinatorial libraries illustrated in Figure 5 reveals that the maps preserve the notion of molecular similarity, exhibit a



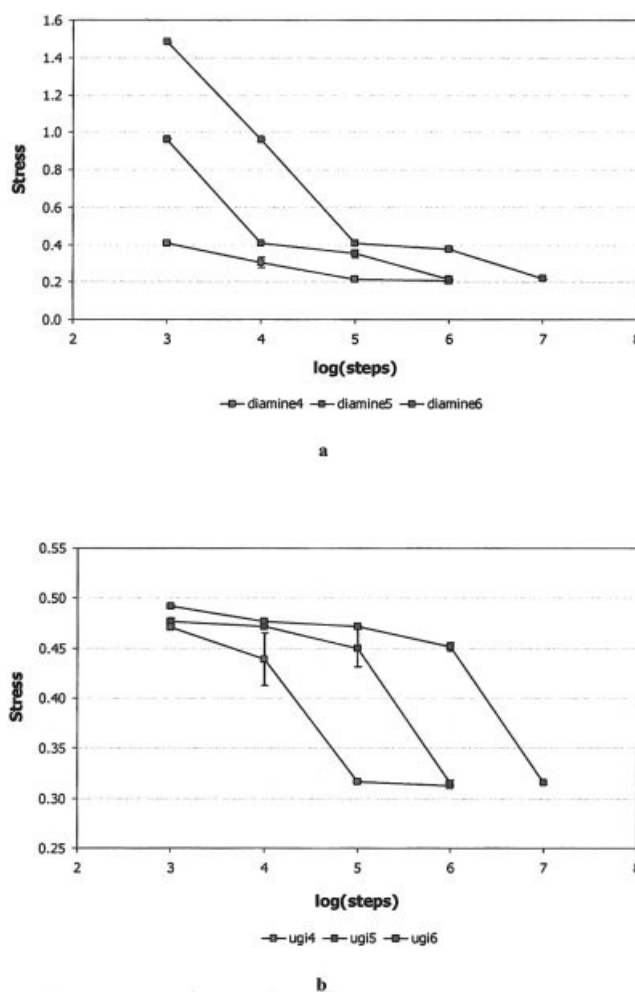
**Figure 5.** Two-dimensional SPE maps of the *diamine4* and *ugi4* data sets. The maps exhibit a very different structure, which reflect the different chemical descriptors and similarity measures employed.

chemically meaningful global structure, and manifest clear variation patterns that correspond to chemically distinguishing features such as chain length, ring structure, heteroatom content, and other relevant properties captured by the input descriptors.

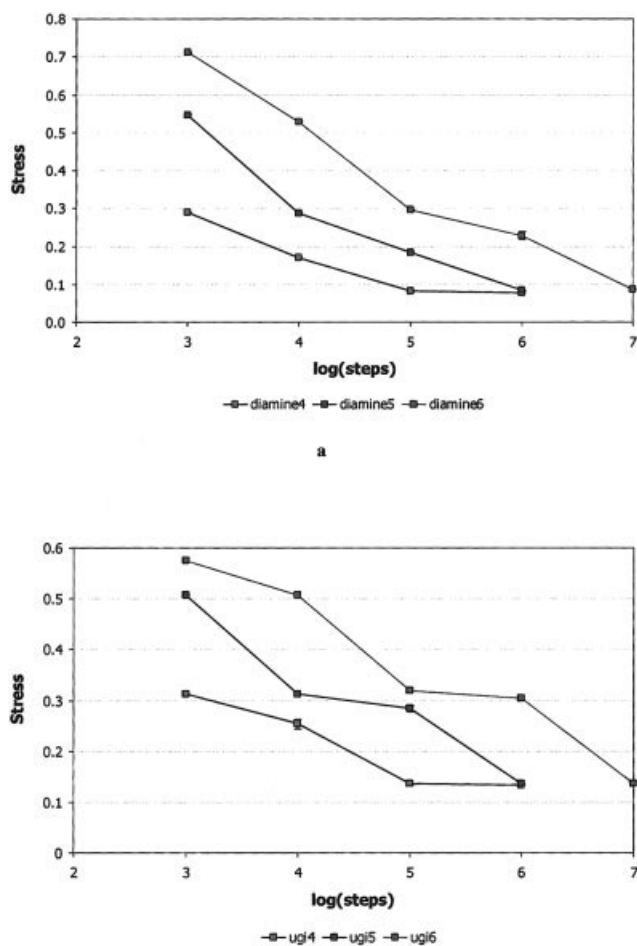
The analysis of chemical spaces, particularly those derived from combinatorial chemical libraries, is becoming an integral part of modern drug design and involves data sets of truly staggering size.<sup>18</sup> We believe that SPE will be an invaluable tool in this and many other problem domains because of its ability to convey complex relationships in an intuitive manner without loss of individual detail.

## Conclusions

Stochastic proximity embedding is a simple, fast, and scalable algorithm for producing low-dimensional representations of data



**Figure 6.** Mean and standard deviation of the stress obtained after 30 independent two-dimensional SPE embeddings of the *diamine* and *ugi* data sets, as a function of the number of steps. Each embedding was carried out using 100 cycles and a linear learning schedule from  $\lambda_{\max} = 1$  to  $\lambda_{\min} = 0.01$ , and was based on a different initial configuration and random number seed. Three data sets containing  $10^4$ ,  $10^5$ , and  $10^6$  members, respectively, are shown for each library, to demonstrate the linear scaling of the algorithm.



**Figure 7.** Mean and standard deviation of the stress obtained after 30 independent five-dimensional SPE embeddings of the *diamine* and *ugi* data sets, as a function of the number of steps. Each embedding was carried out using 100 cycles and a linear learning schedule from  $\lambda_{\max} = 1$  to  $\lambda_{\min} = 0.01$ , and was based on a different initial configuration and random number seed. Three data sets containing  $10^4$ ,  $10^5$ , and  $10^6$  members, respectively, are shown for each library, to demonstrate the linear scaling of the algorithm. This plot, along with Figure 6, demonstrate that the computational complexity of the algorithm is independent of the origin, internal structure and dimensionality of the data set.

supplied in the form of proximities. The method offers two main computational advantages: (1) it does not require the complete distance ( $d_{ij}$ ) or proximity ( $r_{ij}$ ) matrix, thus avoiding quadratic

scaling in time and memory; and (2) it uses stochastic descent, which produces a good solution after a relatively modest number of iterations. SPE can be used to extract constraint surfaces of any desired dimensionality and, because it works directly with proximity data, it can be used for both dimensionality reduction and feature extraction. The coordinates produced by SPE can further be used as input to a parametric learner such as a neural network to derive an explicit mapping function between the observation and embedded spaces.

## Acknowledgments

The author thanks Drs. Victor S. Lobanov, Dmitrii N. Rassokhin, Sergei Izrailev, Huafeng Xu, and Raymond F. Salemme of 3-Dimensional Pharmaceuticals, Inc. for many useful discussions.

## References

1. Kruskal, J. B. *Psychometrika* 1964, 29, 115.
2. Sammon, J. W. *IEEE Trans Comp* 1969, 18, 401.
3. Borg, I.; Groenen, P. J. F. *Modern Multidimensional Scaling: Theory and Applications*; Springer: New York, 1997.
4. Crippen, G. M.; Havel, T. F. *Distance geometry and molecular conformation*; Research Studies Press: Somerset, UK, 1988.
5. Chang, C.-L.; Lee, R. C. T. *IEEE Trans Syst Man Cybern* 1973, 3, 197.
6. Pykett, C. E. *Electron Lett* 1978, 14, 799.
7. Lee, R. C. Y.; Slagle, J. R.; Blum, H. *IEEE Trans Comp* 1977, 27, 288.
8. Biswas, G.; Jain, A. K.; Dubes, R. C. *IEEE Transact Pattern Anal Machine Intell* 1981, 3, 701.
9. Agrafiotis, D. K.; Lobanov, V. S. *J. Chem Inf Comput Sci* 2000, 40, 1356.
10. Rassokhin, D. N.; Lobanov, V. S.; Agrafiotis, D. K. *J Comput Chem* 2001, 22, 373.
11. Agrafiotis, D. K.; Rassokhin, D. N.; Lobanov, V. S. *J Comput Chem* 2001, 22, 488.
12. Agrafiotis, D. K.; Lobanov, V. S. *J Comput Chem* 2001, 22, 1712.
13. Edelsbrunner, H.; Mucke, E. *ACS Transact Graphics* 1994, 13, 43.
14. Hall, L. H.; Kier, L. B. In *Reviews in Computational Chemistry*; Boyd, D. B., Lipkowitz, K. B., Eds.; VCH Publishers: New York, 1991; p 367.
15. Agrafiotis, D. K.; Bone, R. F.; Salemme, F. R.; Soll, R. M.; 3-Dimensional Pharmaceuticals, Inc., 1995.
16. Robbins, H.; Monroe, S. *Ann Math Stat* 1951, 22, 400.
17. Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. *Nature* 1986, 323, 533.
18. Agrafiotis, D. K.; Lobanov, V. S.; Salemme, F. R. *Nat Rev Drug Discovery* 2002, 1, 337.