Supplementary material for:

Sorting Points Into Neighborhoods (SPIN): Data Analysis and Visualization by Ordering Distance Matrices

D. Tsafrir[1], I. Tsafrir[1], L. Ein-Dor[1], O. Zuk[1], D.A. Notterman[2] and E. Domany[1]

[1] Department of Complex Systems, Weizmann Institute of Science, Rehovot 76100, Israel;

[2] Departments of Pediatrics and Molecular Genetics, UMDNJ-Robert Wood Johnson Medical School.

January 25, 2005

Running Times

Table 1 summarizes the parameters and running times for some examples given in this article. The results may depend on the starting permutation and several restarts are sometimes needed to find the global minimum. However, one of the strengths of SPIN is that from a practical point of view, convergence to the global minimum is often not necessary. In most cases the local minima reached by SPIN are almost as informative for extracting structural information.

Proofs for the STS algorithm

Complexity

We prove that the STS problem is NP-Complete by finding a reduction from the STS problem to the well known, NP-complete, problem of proving that a graph contains a *clique* of size k [Garey and Johnson, 1979].

Let $G =< V, E >$ be an (undirected) graph on n vertices. Define D as follows:

$$D_{ij} = \begin{cases} 1 & \text{if } i \neq j,\ (V_i, V_j) \in E, \\ 2 & \text{if } i \neq j,\ (V_i, V_j) \notin E, \\ 0 & \text{if } i = j. \end{cases} \tag{1}$$

Clearly, D is non-negative, symmetric, and satisfies the triangle equality. Thus, D is a distance matrix.

Data	Size	σ	Iter.	T(sec)
rod	500	100,10,1	3	1.3
Ring	500	50,20,10,1	4	1.9
Cell Cycle	500	50,20,10,5	4	1.9
Smiley	800	1.2	50	92
7rods	1400	0.3	50	960

Table 1: Details for several of the examples given in this work. Size refers to the number of points in the data. σ is the width of the neighborhood (i.e. the running times were calculated using the *Neighborhood* algorithm. The *STS* algorithm gives results of the same quality only for the rod data set, with the benefit of slightly reduced running time). When using the "annealing" procedure the widths are given in order of use. Iter. stands for the number of iterations required to obtain the image presented in the article. T is the running time on an IBM with Intel(R) Pentium(R) 4 Mobile CPU 1.6 Ghz. Since the results may depend on initial starting permutation, the number of iterations may vary, and several restarts may be needed.

Fix now some $k \in [1, n]$. Define the vector $X \in \mathbb{R}^n$ by $X_i = 1_{i \geq n-k+1}$. Clearly, X is non-decreasing.

claim: $G =< V, E >$ has a clique of size k if and only if $\min_{P \in S_n} X^T PDP^T X = (k-1)k$.

proof: Let $C \subset V$ be a $k-$*clique* on $G =< V, E >$, and let $\hat{P} \in S_n$ be a permutation such that $\hat{P}(i) \in C$ $\forall\ n-k+1 \leq i \leq n$. Thus, the bottom-right $k \times k$ sub-matrix of the permuted distance matrix, $D^{\hat{P}} = \hat{P}D\hat{P}^T$, corresponds to the k vertices composing C. Therefore, for every $n-k+1 \leq i \neq j \leq n$, $D^{\hat{P}}_{ij} = 1$. Using the X defined above to calculate $\mathcal{F}(\hat{P})$ one obtains:

$\sum_{i,j=1}^{n} X^T{}_i D^{\hat{P}}_{ij} X_j = \sum_{i,j=n-k+1}^{n} D^{\hat{P}}_{ij} = (n-(n-k+1)+1)(n-(n-k+1)+1) - (n-(n-k+1)+1) = k(k-1)$.

But from definition :

$$D^Q{}_{ij} \geq D^{\hat{P}}{}_{ij} \forall Q \in S_n,\ \forall n-k+1 \leq i,j \leq n$$

Therefore: $\sum_{i,j=1}^{n} X^T{}_i D^Q_{i,j} X_j = \sum_{i,j=n-k+1}^{n} D^Q_{i,j} \geq (n-(n-k+1)+1)(n-(n-k+1)+1) - (n-(n-k+1)+1) = k(k-1)$,

which yields: $\min_{P \in S_n} X^T PDP^T X = (k-1)k$ ■

Convergence

We now prove that convergence to a fixed point is guaranteed after a finite time. To do this we give the requirements for the input matrix D and the weight vector X in mathematical terms:

Take n **distinct** points $z^1, \ldots z^n \in R^d$, for some $d \in \mathbb{N}$, and a real $p \in (1, 2]$, such that :

$$D_{i,j} = ||z^i - z^j||_p,\ 1 \leq i,j \leq n$$

Let X be a strictly monotonically increasing vector, namely: $X_i < X_j \iff i < j$.

The proof is based on showing that every STS iteration reduces the cost function, $\mathcal{F} = {X^t}^T D X^t$, utilizing the following lemma:

Lemma

1. ${X^{t+1}}^T D X^t \leq (QX^0)^T D X^t, \ \forall Q \in S_n, \ t \geq 0$
2. $X^{t+1} \neq X^t \Rightarrow {X^{t+1}}^T D X^t < {X^t}^T D X^t$.

Proof

1. Note that :

$$
\begin{aligned}
{X^{t+1}}^T D X^t = (P^t{}^T X^0)^T D X^t = \\
= {X^0}^T P^t D X^t \equiv {X^0}^T U^t \qquad (2) \\
(QX^0)^T D X^t = {X^0}^T Q^T D X^t = \\
= {X^0}^T Q^T (P^t)^{-1} P^t D X^t \equiv {X^Q}^T U^t \qquad (3)
\end{aligned}
$$

   And X^Q is some permutation of X^0, for any $Q \in S_n$. But, U^t is a non-increasing vector, while X^0 is a strictly increasing vector. Thus, according to a theorem by *Hardy*, *LittleWood* and *Polya* [Hardy et al., 1959] $X^0 U^t \leq Q(X^0) U^t$, $\forall Q \in S_n$, so $X^0 U^t \leq X^Q U^t$, as desired.

2. From 1 we get ${X^{t+1}}^T D X^t \leq {X^t}^T D X^t$. Assume negatively that equality holds. Then, we get : ${X^0}^T P^{t-1} D X^t = {X^t}^T D X^t = {X^{t+1}}^T D X^t = {X^0}^T P^t D X^t$. But X^0 is increasing, $P^t D X^t$ is decreasing and $P^{t-1} D X^t$ is a permutation of it. Therefore $P^{t-1} D X^t = P^t D X^t$, and thus $P^{t-1} D X^t$ is non-increasing, and since we have started from it, we get $P^t = P^{t-1}$ and $X^{t+1} = X^t$, which is a contradiction. ■

claim: The STS algorithm converges to a fixed point after a finite number of iterations.

Proof :

According to Thm. 2.11 in [Baxter, 1991], D is Almost Negative Definite. That is, $\sum_{i=1}^n V_i = 0 \Rightarrow V^T D V \leq 0, \forall V \in \mathbb{R}^n$. Since X^t is a permutation of X it follows that

$$
(X^{t+1} - X^t)^T D (X^{t+1} - X^t) \leq 0 \qquad (4)
$$

Since D is symmetric it follows that

$$
{X^{t+1}}^T D X^{t+1} + {X^t}^T D X^t \leq 2 {X^{t+1}}^T D X^t \Rightarrow
$$

$$X^{t+1^T} D X^{t+1} - X^{t^T} D X^t \leq 2(X^{t+1^T} D X^t - X^{t^T} D X^t), \tag{5}$$

But the algorithm never stays at the same point for more than one iteration (step 4), namely $X^{t+1} \neq X^t$ and therefore, according to the previous lemma:

$$X^{t+1^T} D X^{t+1} - X^{t^T} D X^t < 0$$

To conclude, the energy function $\mathcal{F}(t) = X^{t^T} D X^t$ is a strictly decreasing function of t. Therefore the algorithm terminates after a finite number of steps. ■

This proves that for L_p norms with $p \in (1, 2]$, STS converges to a local minimum. For other norms, STS might converge to a cycle, however the cycle can be viewed as *local minima*, since it still minimizes $\mathcal{F}(X^t, X^{t+1})$ (All the cycle has the same $\mathcal{F}$.) Convergence to a global minimum of $\mathcal{F}_{\mathcal{X}}$ is not guaranteed.

Relationship between STS and PCA

We now give a heuristic argument showing that the permutation outputted by the STS algorithm, is likely to be similar to ordering the points according to their projection on the first PCA. Nevertheless, as will be demonstrated later, in some cases the two permutations are different, and the STS ordering may be more useful. Let D be the genes' distance matrix, we first prove that for normalized genes, the genes' projections on the first PCA are given by the components of the eigenvector of D that corresponds to the most negative eigenvalue. Consider the SVD representation of the expression matrix E as $E = USV^T$. Here $U_{n\times n}$ and $V_{m\times m}$ are unitary matrices, and $S_{n\times m}$ is a diagonal matrix whose diagonal elements are the singular values of E. We denote by $V_1(U_1)$ the eigenvector of $EE^T(E^TE)$ corresponding to the largest eigenvalue λ_1. Assuming that the expression matrix is centered and normalized, the distance matrix D is given by $D = 2(I - EE^T)$. Thus D has the same eigenvectors as EE^T, and its eigenvalues are given by $2(1 - \lambda_i)$, $i = 1, \ldots, n$. Using the unitarity of V we write the projection of E on the first PCA as :

$$EV_1 = USV^TV_1 = US_1 = \lambda_1 U_1,$$

where S_1 is the first column of S. But U_1 is also an eigenvector of D, specifically corresponding to its smallest eigenvalue $(2(1 - \lambda_1))$.

As a conclusion, we get that if P is a permutation such that $U_1^{(P)} \equiv PU_1$ is a monotonically decreasing vector, permuting D according to P gives the ordering according to the projection on the first PCA. Thus, $PDP^TU_1^{(P)} = \lambda_1 U_1^{(P)}$ which is a monotonically decreasing vector. Now represent X as a linear combination $X = \sum_{i=1}^n \alpha_i U_i^{(P)}$, to get $PDP^TX = \sum_{i=1}^n \alpha_i \lambda_i U_i^{(P)}$. Note that λ_1 is the largest absolute value eigenvector, and also, since $U_1^{(P)}$ and X are both increasing, we expect α_1 to be large. Therefore, PDP^TX gets a large contribution from the component $\alpha_1\lambda_1 U_1^{(P)}$, thus it is likely to be approximately also decreasing. This explains why, in the majority of examined

cases, P, or a permutation very close to it, was a fixed point of the STS iteration, yielding a final STS ordering which is indeed similar to ordering according to the first PCA.

Figure 1 provides a toy example where the STS permutation and the PCA progression produce different results. The three compact spheres that are embedded in the data can not be clearly distinguished by using PCA projection (fig. 1a-b). The STS-permuted distance matrix (fig. 1c-d), on the other hand, can be used for visual identification of the three distinct objects, and for determining that they are all compact spheres.

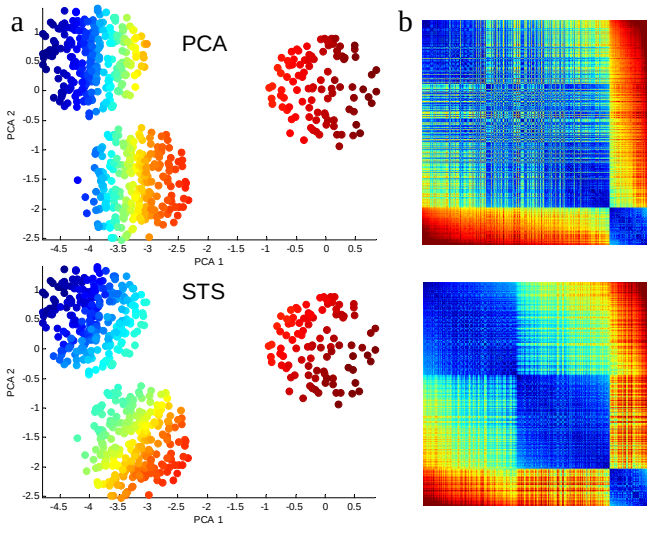

Figure 1: Toy data set of 600 points in $2D$ that comprise three distinct spheres. (a) A scatter plot of the points in the first and second PCA plane. The coloring of points indicates their relative placement in the ordering according to projection on to the first PCA, going from dark blue to dark red. (b) The correspondingly permuted distance matrix, i.e. the point that is colored the darkest blue in a. is represented by the first row, etc. Note that the direction of the first PCA is highly affected by the right-most sphere, to the degree that the two spheres on the left are not distinguished. (c)-(d) The corresponding figures, permuted according to STS, which clearly separates all three objects.

Loss of structure in randomized expression data

In order to show that SPIN does not find structures that do not exist, we performed a random-permutation test with the following outcome (see fig. 2). This question is interesting since *SPIN* will always try to locate the most informative permutation with regard to the given data's structure. The same colon cancer expression data that was analyzed in the article is also used here, with the addition of one initial pre-processing step, consisting of random permuting of the values in the expression matrix. All other analysis stages remaind the same as described in section *Application to colon cancer*, and the results of *SPIN*-sorting the $1,000$ highest variance *genes* of the randomized expression data are given in fig. 2.

As expected, the randomization process destroys the structure of the expression data; instead of containing objects of various shapes the conformation is in that of a rather uniform sphere (see fig. 2a). The image of the SPIN-sorted distance matrix for randomized data is extremely different from that of the original expression data

(compare fig. 4b-c in the article to fig. 2b-c), conveying to the user that the randomized data does not contain any structured objects. The observation that *SPIN* does not generate false positives (i.e. the appearance of a signature of elongation in a *SPIN*-sorted matrix necessarily indicates an elongated conformation of data points) follows from the fact that in a *SPIN* analysis the distance matrix is only permuted and not distorted in any way.

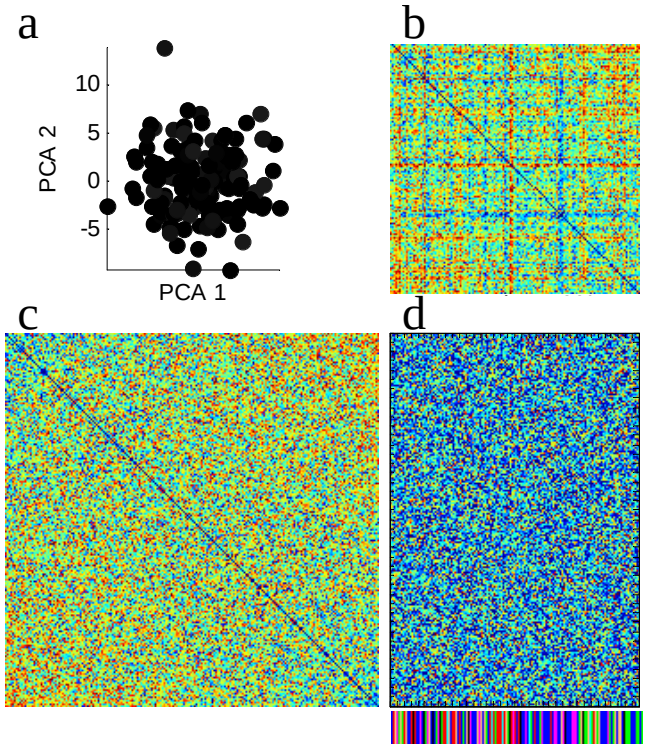

Figure 2: *SPIN* analysis of randomized expression data. (a) Projection of the samples onto the first (x-axis) and second (y-axis) principal components, calculated in gene-space. (b) *SPIN*-permuted distance matrix for the samples. Colors depict dissimilarity levels between samples, with red (blue) indicating large (small) distances. Note that this rather uniform spherical cluster, which has no significant elongation, manifests in a diffuse texture in the SPIN-sorted distance matrix. As a rule, the smoothness of the texture in the image of the distance matrix is a function of the elongation of the cluster.(c) Genes *SPIN*-permuted distance matrix. (d) Two-way sorted expression matrix. Here colors depict relative expression intensities, where red (blue) denotes relatively high (low) expression. The colored bar below the matrix provides the tissues' clinical identity, using the same color scheme as in fig. 4.

References

B.J.C. Baxter. Conditionally positive functions and p-norm distance matrices. *Constr. Approx.*, 7:427–440, 1991.

M. R. Garey and D. S. Johnson. *Computer & Intractability: A Guide to the Theory of NP-Completeness.* W H Freeman, 1979.

G. H. Hardy, J. E. Littlewood, and G. Polya. *Inequalities.* Cambridge University Press, Cambridge, England, 1959.