VISUALIZED ATLAS OF A GENE EXPRESSION DATABANK

Jarkko Venna[1] *and Samuel Kaski*[2,1]

[1] Neural Networks Research Centre, Helsinki University of Technology,
P.O. Box 5400, FI-02015 HUT, FINLAND
[2]Department of Computer Science, P.O. Box 68,
FI-00014 University of Helsinki, Finland

{jarkko.venna, samuel.kaski}@hut.fi

ABSTRACT

We construct an atlas of a gene expression databank, to visualize similarity relationships between expression data sets. Such an atlas could be used as an interface to the databank, for users searching for relevant background data or data for their own in-silico analyses. The two main research problems in constructing an atlas are (1) to preprocess the data to make different sets commensurable, and (2) to visualize the data. In this work we use only very simple preprocessing to study its feasibility, and focus on the visualization. We compare several recently introduced methods in the task, and show that a method called curvilinear components analysis outperforms the newer ones in terms of trustworthiness of the projections. The visualizations reveal the main sources of variation in the data, namely the differences between data sets, different labs, and different measurement methods, which supports feasibility of the visualization method in the task. The other conclusion is that better methods are needed for making the data sets commensurable.

1. INTRODUCTION

A large community-resource or private gene expression databank consists of numerous data sets submitted by several parties. They may have been measured for different purposes, with different treatments and methods in different laboratories. Several such databanks have been established and they continue to grow. A key challenge is how to best use the databanks to support further research. In this paper we discuss a subproblem: How to construct a visualization of the actual contents to help in finding interesting or relevant sets. Such a visualized atlas would complement potential meta-data about the data sets, by revealing more about the mutual similarities of the actual data than the imperfect meta-data, that is, textual annotations and descriptions.

The first big problem in visualizing gene expression data sets stems from their dimensionality, which may be thousands or even tens of thousands, equaling the number of genes on a microarray. We will compare several dimensionality reduction techniques, with the traditional linear principal components analysis (PCA) [1] serving as the baseline.

We have earlier [2] compared PCA, multidimensional scaling methods (MDS) [3], and the Self-Organizing Map (SOM) [4] in a related task: visualization of similarity relationships between genes, based on their expression profiles in a set of treatments. The result of the comparison was that the SOM visualizations were more trustworthy, in the sense that a set of genes found close-by on a SOM display was more likely to be similar in terms of the original data as well. In other words, the proximities visible on the displays were more trustworthy. The other side of the coin is whether the visualization is able to show all of the proximities present in the original data. It turned out that the SOM was among the best methods here as well. There is later evidence [5] that a related method, curvilinear components analysis (CCA) [6], may outperform even the SOM in this task.

There has recently been a surge of interest in methods for finding latent lower-dimensional manifolds of data, or nonlinear embeddings of smaller-dimensional data manifolds in a higher-dimensional data space. We will include these methods in the comparison with CCA.

Another main problem in visualizing gene expression data sets, and in fact in all comparisons of the sets, is how to make them commensurable. The measurement results depend at least on experimental and measurement procedures, the specifics of the organism and its biological state, biological sampling, measurement devices, and normalization and postprocessing procedures. Nevertheless, even very simple normalization procedures have resulted in promising data analysis results in a recent study which combined data from a variety of human cancer studies [7]. This prompted us to study the feasibility of a gene expression data atlas, where only very simple procedures have been applied to make the sets commensurable.

2. METHODS

In this section we describe briefly the main classical methods for visualizing similarity relationships in the data, and the recent ones that focus on finding data manifolds or embeddings.

2.1. Multidimensional scaling

We did not include traditional multidimensional scaling to the comparisons, but a short description helps to understand the more complex methods below.

There are several different variants of MDS [3], but they all have a common goal: to find a configuration of points that preserves the pairwise distance matrix. The simplest version is the linear MDS [8, 9], also called classical scaling. The solution to Linear MDS can be found by solving an eigenvalue problem.

A slightly more complex version is metric MDS. Its cost function is

$$E = \sum_{ij} (d_{i,j} - d(y_i, y_j))^2, \tag{1}$$

where $d_{i,j}$ is the distance in the input space and $d(y_i, y_j)$ the distance in the output space, between the representations y_i and y_j of the points.

Most versions of MDS use a variant of this cost function. Sammon's mapping [10] gives small distances a larger weight. In non-metric MDS [11] the distances are modified by a monotonic function. There is a huge number of different variants, but all have basically the same form.

2.2. Principal component analysis (PCA)

The goal of PCA [1] is to find components having maximal variance. Linear components correspond to directions in or subspaces of the data space, and when the data are projected to a PCA component the variance in the data is preserved maximally. The components can be found by solving the eigenvalue problem

$$\mathbf{C}_x \mathbf{a} = \lambda \mathbf{a} \ , \tag{2}$$

where $\mathbf{C}_x$ is the covariance matrix of the vectorial data $\mathbf{x}$. For visualization the data points need to be projected onto a two-dimensional plane defined by the two main components. This is done by

$$\mathbf{y}_i = \mathbf{A}\mathbf{x}_i, \tag{3}$$

where $\mathbf{A}$ is the matrix containing the eigenvectors corresponding to the two largest eigenvalues, and $\mathbf{y}_i$ is the two-dimensional representation of $\mathbf{x}_i$.

PCA is very closely related to linear MDS. It can be shown [9] that when the dimensionality of the solutions is the same, the projection of the original data to the PCA subspace equals the configuration of points found by linear MDS that is calculated from the Euclidean distance matrix of the data. Thus the cost function of PCA tries to preserve the squared distances between data points.

2.3. Locally linear embedding (LLE)

The LLE algorithm [12] is based on the assumption that the data lies on or close to a low-dimensional manifold in the high-dimensional space. If this is the case then we can make a locally linear approximation of the manifold, and assume that a point and its neighbors lie in or close to a locally linear subspace on the manifold. The geometry of this subspace can be captured by calculating the linear coefficients that reconstruct each data point from its neighbors. Here the neighbors are the k nearest neighbors of the data point. The reconstruction error is defined as

$$E(\mathbf{W}) = \sum_i |\mathbf{x}_i - \sum_j W_{ij}\mathbf{x}_j|^2. \tag{4}$$

To find the optimal weight matrix $\mathbf{W}$ the reconstruction error is minimized subject to the constraints that $W_{ij} = 0$ if i and j are not neighbors, and $\sum_j W_{ij} = 1$.

For visualization we want to reduce the dimensionality of the data. To achieve this we have to solve another optimization problem,

$$E(\mathbf{Y}) = \sum_i |\mathbf{y}_i - \sum_j \mathbf{W}_{ij}\mathbf{y}_j|^2 \ , \tag{5}$$

where $\mathbf{y}_i$ is the low-dimensional representation of the data point i. This time the weight vectors are kept constant during optimization and the positions of the data points are changed. The problem can be solved by finding the $p+1$ smallest eigenvalues of the matrix $(\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W})$ (details in [12]), where p is the dimensionality of the output. The smallest eigenvalue corresponds to a constant eigenvector and the next p give the coordinates of the data points within the manifold space.

The LLE implementation at http://www.cs.toronto.edu/∼roweis/lle/ was used in the experiments.

2.4. Laplacian Eigenmap

The Laplacian Eigenmap [13] algorithm is similar to the LLE algorithm. The first step is to form the k-nearest-neighbor graph. Each data point is a vertex in the graph. There is an edge from point i to point j if j is among the k nearest neighbors of i. The graph differs from the one used in LLE in that the neighbor relation is symmetric. If the data point i is a neighbor of j then j is also always a neighbor of i. After the graph has been formed the edges have to be given weights. The simple method of assigning $W_{ij} = 1$ if the points i and j are neighbors and zero otherwise has been found to work well in practice [14].

The configuration of points in the low-dimensional space can be found by solving the generalized eigenvalue problem

$$\mathbf{L}\mathbf{y} = \lambda \mathbf{D}\mathbf{y}, \tag{6}$$

where $\mathbf{D}$ is a diagonal matrix with elements $D_{ii} = \sum_j W_{ij}$, and $\mathbf{L} = \mathbf{D} - \mathbf{W}$. The embedding of the data points is given by the eigenvectors having the p smallest eigenvalues, after discarding the smallest (always zero) eigenvalue.

2.5. Isomap

The Isomap [15] is a variant of MDS. It finds a configuration of points that matches the given distance matrix.

The difference from traditional MDS is in how the distances are defined. Isomap uses geodesic distances instead of direct pairwise distances. The geodesic distances are approximated with the shortest path distances calculated along the k-nearest-neighbor graph. The graph is defined in the same way as in the Laplacian Eigenmap, except that the weights of the edges are set to the Euclidean distances between the connected points.

The actual embedding of points is found by standard linear MDS, applied to the shortest-path distance matrix. It has been shown [16] that this algorithm is asymptotically able to recover certain types of manifolds.

The Isomap implementation available at http://isomap.stanford.edu/ was used in the experiments.

2.6. Curvilinear component analysis (CCA)

Like Isomap, CCA [6] has similarities with MDS. Where Isomap changes the definition of distances, CCA chooses to preserve only a subset of the distances. The starting point is a random initialization of points (y_i) in the reduced-dimensional output space, and a pairwise distance matrix between the original data points (x_i). The cost function measures preservation of the original pairwise distances, but now weighted by a coefficient F that depends on the distance between the points in the *output space*. Here CCA differs from traditional MDS methods. The idea is to concentrate on preserving distances between close-by points in the output space. The cost function is

$$E = \frac{1}{2} \sum_i \sum_{j \neq i} (d(x_i, x_j) - d(y_i, y_j))^2 F(d(y_i, y_j), \lambda_y). \tag{7}$$

The term $F(d(y_i, y_j), \lambda_y))$ determines how strongly errors in reproducing the distance between the points i and j contributes to the cost function. It is usually defined as an area of influence around a data point in the output space:

$$F(d(y_i, y_j), \lambda_y)) = \begin{cases} 1 & \text{if } d(y_i, y_j) \leq \lambda_y \\ 0 & \text{if } d(y_i, y_j) > \lambda_y. \end{cases} \tag{8}$$

The cost function is optimized using a form of stochastic gradient descent algorithm. In the beginning of optimization the radius of the area of influence, λ_y, is kept large enough to cover all or at least most of the data points. During the optimization it is slowly reduced to zero. Thus, at initial stages CCA performs exactly as standard nonlinear MDS where all distances are treated equally. The dynamic reduction of the area of influence around data points results in an unfolding effect in the mapping. Contrary to the other methods described here, the cost function of CCA can have several local optima. Although this can potentially cause problems, the solutions found by CCA have been quite good in practice, even starting from only one initialization.

3. MEASURING TRUSTWORTHINESS OF A VISUALIZATION

When visualizing similarities of data points, the local ones are the most salient: when looking at a point the first perceptions are which other points are proximate, and which proximate points form groups. We have developed a way to measure how trustworthy the proximities presented by the visualization are [2, 17].

We consider a projection onto a display trustworthy if the set of k closest neighbors of a point on the display are also close-by in the original space. This is measured for all data points. Our *measure of trustworthiness* quantifies errors in terms of rank distances, sums the errors over all data points, and normalizes the result to lie between 0 and 1.

While trustworthiness measures show how well the points in a neighborhood on the display match the neighborhood in the original space, it is also of interest to know what happens to those points that are pushed out of the neighborhood in the visualization process. The original neighborhood might not be preserved because of *discontinuities* in the projection. As a result of the latter kinds of errors, not all proximities existing in the original data are visible in the visualization.

The errors caused by discontinuities may be quantified analogously to the errors in trustworthiness. A neighborhood of k closest data samples in the original space is defined for each sample, and whenever some of the samples is projected outside of the neighborhood after the projection, the errors are computed in terms of rank distances in the output space. The errors are again summed over all data samples, and the result normalized to lie in between 0 and 1.

4. ATLAS OF A DATABANK

We constructed an atlas of a gene expression databank, aimed at revealing proximity relationships between and within the data sets of the databank. The atlas is computed of a collection of cancer expression sets which have been preprocessed only lightly, to make them somewhat more commensurable.

4.1. Data and preprocessing

We used the large collection of human gene expression arrays collected by Segal et al. [7]. (The normalized expression compendium is available from http://dags.stanford.edu/cancer.) The compendium consists of 26 different sets, each from a different publication, from altogether 1973 arrays. Three different types of microarrays were included, and the studies were carried out in 6 different institutions.

The data sets were normalized using the same methods as in [7]. In the expression values measured with Affymetrix chips, logs (base 2) were taken (truncating to 10 expression values that are below 10). For the data sets generated using cDNA chips the log-ratio (base 2) of the measured and control sample was taken. After this the expression values of data sets were normalized, for each gene and data set separately, by subtracting the mean of the gene's expression in the data set from each expression value. Finally, the values were rounded to the accuracy of one decimal.

For the visualization we then removed samples with missing values from the data. First we removed genes that were missing from more than 300 arrays. Then we removed the arrays that still contained missing values. This resulted in a data set containing 1278 arrays and 1339 genes.

4.2. Comparison of visualization methods

Visualization of the compendium of gene expression data sets is a very demanding task for any method. The visualization has to reduce the dimensionality from 1339 to 2 while still preserving local structure of the data. To make the task even harder the data in the compendium has been produced with different methods and comes from different types of experiments. All this means that it is very unlikely that there is a nicely-formed low-dimensional manifold in the data space.

We compared the visualization methods by computing the measures described in Section 3 for varying values of the neighborhood parameter k. Small values are the most important, for the smallish neighborhoods are more salient in visualizations, and hence if they are not trustworthy neither is the whole display. Methods having a nearest neighbor parameter were run with the parameter ranging from $k = 4$ to $k = 20$, CCA was run ten times from different random initializations, and the best ones in terms of the trustworthiness were selected.

The performance of the methods can be seen in Figure 1. None of the visualizations have a particularly high trustworthiness. This reflects the difficulty of the task. CCA was the best, followed by Laplacian Eigenmap and PCA. All of the methods were somewhat better in preserving the original neighborhoods. PCA was the best in this respect, followed by Laplacian Eigenmap. LLE performed poorly on both measures. That PCA performs so well, in conjunction with the overall low trustworthiness values, suggests that there is very little low-dimensional manifold structure that could be utilized in the data.

4.3. Are data from different platforms commensurable?

It is conceivable and even plausible that a large proportion of the variance in the data is due to secondary attributes. The measurement platform, that is, whether the data has been measured by cDNA microarrays or oligo chips, will affect the results, and likewise the measurements carried out in different laboratories may be different for many reasons.

This would naturally reduce the usefulness of the atlas. If a user is studying, say, B lymphoma and searches for relevant data, it would be nice to be able to find and use data from all platforms (and all institutions). This is of course sensible only if the variation due to the platform is relatively small.

We tested this by measuring whether data from a cancer type is more commensurable with data of the same cancer type but measured on a different platform, or with any data measured from the same platform. For instance, to see whether the B lymphoma arrays in the gene expression compendium were really organized based on the cancer type and not the platform, we selected a set of arrays that were both cDNA and measured from B lymphoma samples. We then measured whether this set was better classified to either the class of B lymphoma arrays or the class of cDNA arrays.

Table 1. Are data from the same platform more similar than data from the same cancer type? Pairwise comparison of the classification strength of the cancer type vs the platform/institution. Data points measured with the platform of the column on samples of the cancer type of the row were classified to either of the two classes. The winner (on the average) is shown in the table, together with the classification rate it achieved. Classification was done in the original data space.

Cancer type/ Platform	cDNA	Hu95	HuGeneFL
B lymphoma	cDNA		B lymphoma
	1.0000		0.7011
Leukemia		Hu95	HuGeneFL
		0.9657	0.9998
Lung Cancer	cDNA	Hu95	
	1.0000	0.8465	
NCI60	cDNA		NCI60
	1.0000		0.8704
Cancer type/ Institution	Stanford	Harvard	MIT
B lymphoma		Harvard	
		0.8397	
Leukemia		Harvard	MIT
		0.9573	0.9783
Lung Cancer	Stanford	Harvard	
	1.0000	0.7755	
NCI60			NCI60
			0.9799

4.3.1. How the classification was done

To measure the commensurability of cancer types in comparison to secondary attributes (platform or institution) we defined a data group X consisting of all data having both a common cancer type and a common value of the secondary attribute. We then defined two additional data groups, Y and Z. Group Y consisted of all data having the same cancer type as X but a different value of the secondary attribute. Group Z consisted of all data having the same secondary attribute value as group X but a different cancer type. We then classified all points as belonging to either group Y or Z.

Technically, we used a k-nearest-neighbor classifier ($k = 5$). If the groups Y and Z differed in size a data set of the same size as the smaller one was sampled from the larger group (same size to give both groups the same prior probability). Each data point in X was then classified by a knn classifier, by finding the k nearest neighbors from the combined set of Y and Z. The majority class within the k

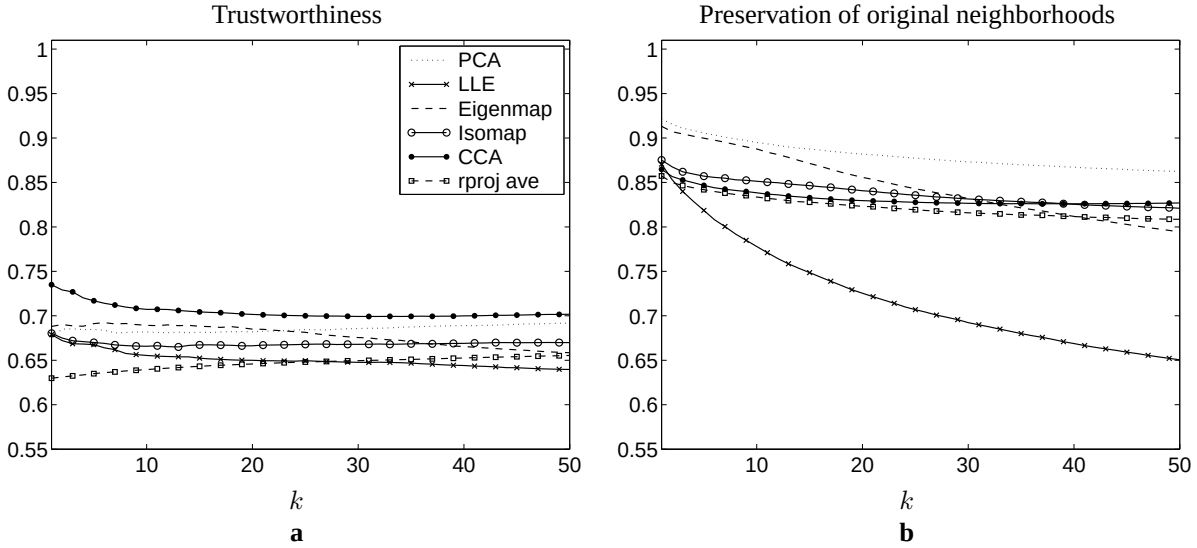

Figure 1. The change in trustworthiness (**a**) and preservation of original neighborhoods (**b**) of the visualized gene expression compendium, as the number of neighbors k in the neighbor set is varied.

closest samples wins.

The classification was repeated 1000 times with different random samples from the larger group. At the end the *mean classification rate* of the group was calculated and reported. We additionally computed P-values for rejecting the hypothesis that the group X comes from the distribution of the attribute group with the larger classification rate, but the results were almost always highly significant and we decided to report the more informative mean classification rates.

4.3.2. Results

The result was that cDNA measurements of B lymphoma samples were always closer to other cDNA measurements, and B lymphoma samples measured with other platforms were more different.

We performed the same experiment on several cancer type vs. platform or institute pairs. The complete results are shown in Table 1.

The results are quite clear concerning the cDNA platform. The cDNA measurements are always closer to other cDNA measurements than to measurements of the same cancer type but made with a different platform. A similar effect, although not as strong, can be found for the Hu95 platform. On HuGeneFL the results vary.

These results strongly suggest that the simple preprocessing used in [7] is not able to remove effects caused by the platform. Note that this does not necessarily imply anything about the validity of the results in [7]; the paper has different goals and methods.

A similar effect can be found between the cancer types and the institute where the data was measured. This may, however, to a large extent be explained by the fact that most institutes prefer a specific platform.

4.4. Visualizing the gene expression compendium

Displays of the gene expression compendium, that is, gene expression atlases computed with the different methods, are presented in Figure 2. In the displays the symbols denote the measurement platform but they could of course alternatively show the institution or cancer type.

First of all, the display shows that the PCA mixes up the measurement platforms badly, whereas CCA differentiates them nicely into different areas of the display (note that none of the methods has had access to the labels; all are completely unsupervised). LLE and Laplacian Eigenmap have artifacts in their display, resulting in badly varying display resolution and hence difficulties in interpretation without using a magnifying glass. The trustworthiness comparisons in Section 4.2 suggest that the displays may not be the most useful ones even with a magnifying glass. The Isomap display is more informative but apparently not as clear in separating the classes as CCA.

Secondly, the CCA display shows that CCA is capable of revealing the separation of the measurement platforms in the data space, which was shown in the previous section to be a major source of variation in the data.

5. CONCLUSIONS

We benchmarked a set of methods for the extremely difficult task of visualizing proximity relationships within the high-dimensional space of microarray measurements. It turned out that an older method called curvilinear components analysis (CCA) outperformed newer methods in terms of trustworthiness of the visualizations.

The methods were compared as a feasibility study for constructing a visualizable gene expression atlas, that is, an atlas of gene expression data sets. It turned out that the simple preprocessing methods could not make the different data sets particularly commensurable. The visualizations did show, however, relationships between the dif-

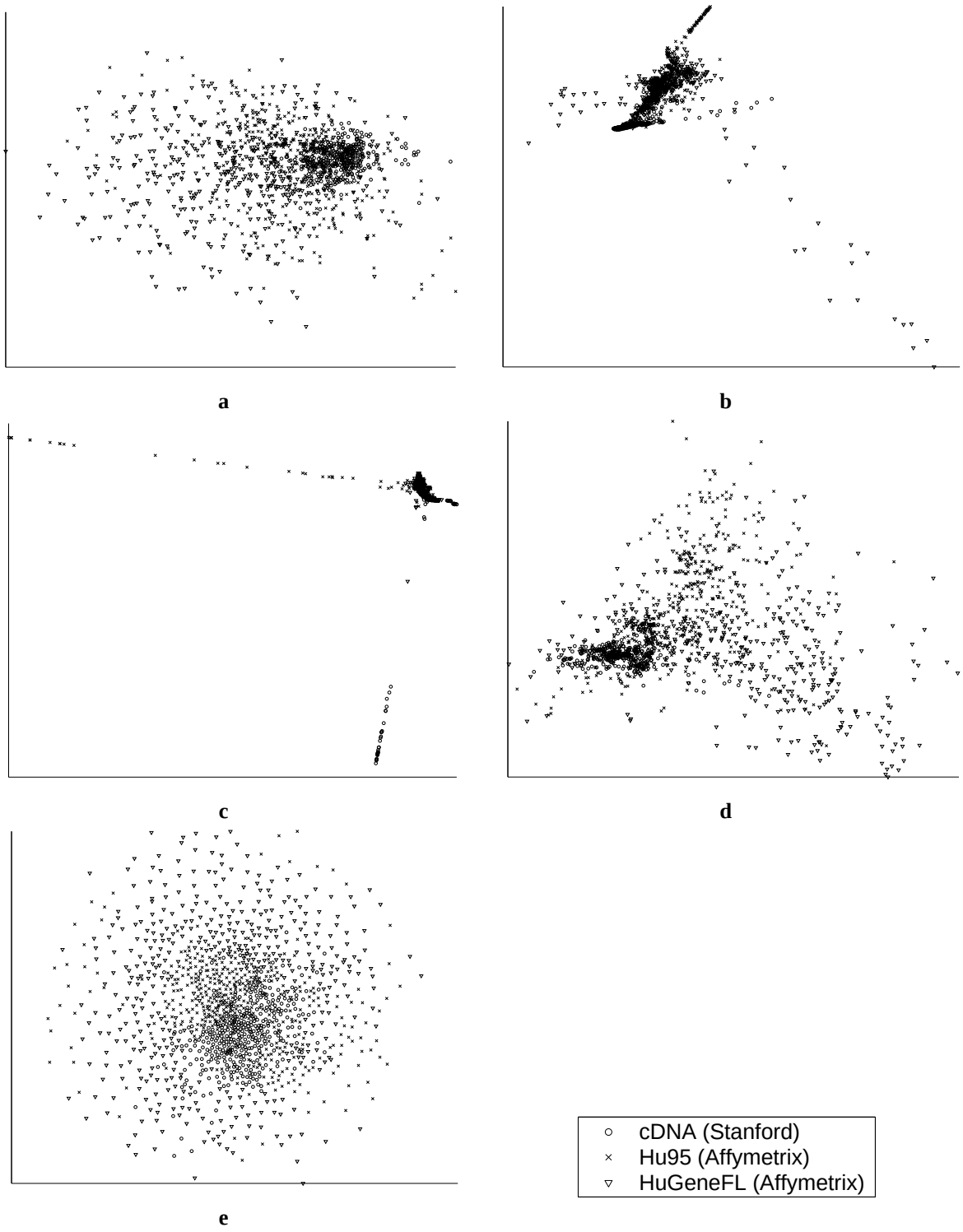

Figure 2. Sample visualizations of the gene expression atlas by (**a**) PCA, (**b**) LLE, (**c**) Laplacian Eigenmap, (**d**) Isomap, and (**e**) CCA. Each dot denotes one microarray; the symbols here show the measurement platform.

ferent labs and measurement array platforms, which are the main sources of variation in the data. Hence, if standardization and more sophisticated preprocessing methods continue to develop to bring the biologically interesting variation to the fore, the information visualization methods are likely to be able to visualize it.

6. ACKNOWLEDGMENTS

This work was supported by the Academy of Finland, decision numbers 79017 and 207467 and by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-202-506778.

7. REFERENCES

[1] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, pp. 417–441,498–520, 1933.

[2] Samuel Kaski, Janne Nikkilä, Merja Oja, Jarkko Venna, Petri Törönen, and Eero Castrén, "Trustworthiness and metrics in visualizing similarity of gene expression," *BMC Bioinformatics*, vol. 4, pp. 48, 2003.

[3] Ingwer Borg and Patrick Groenen, *Modern Multidimensional Scaling*, Springer, New York, 1997.

[4] Teuvo Kohonen, *Self-Organizing Maps*, Springer, Berlin, 3rd edition, 2001.

[5] Johan Himberg, *From insights to innovations: data mining, visualization, and user interfaces*, Ph.D. thesis, Helsinki University of Technology, Espoo, Finland, 2004.

[6] Pierre Demartines and Jeanny Hérault, "Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 148–154, January 1997.

[7] Eran Segal, Nir Friedman Amd Daphne Koller, and Aviv Regev, "A module map showing conditional activity of expression modules in cancer," *Nature genetics*, vol. 36, no. 10, pp. 1090–1098, 2004.

[8] Warren S. Torgerson, "Multidimensional scaling: I. theory and method," *Psychometrika*, vol. 17, no. 4, pp. 401–419, Dec 1952.

[9] J. C. Gower, "Some distance properties of latent root and vector methods used in multivariate analysis," *Biometrika*, vol. 53, no. 3/4, pp. 325–338, December 1966.

[10] John W. Sammon, Jr., "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. C-18, pp. 401–409, 1969.

[11] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrica*, vol. 29, no. 1, pp. 1–26, Mar 1964.

[12] Sam T. Roweis and Lawrence K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, December 2000.

[13] Mikhail Belkin and Partha Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems 14*, Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, Eds., Cambridge, MA, 2001, MIT Press.

[14] Mikhail Belkin and Partha Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," Tech. Rep. TR-2002-01, Department of Computer Science,The University of Chicago, 2002.

[15] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, December 2000.

[16] Mira Bernstein, Vin de Silva, John C. Langford, and Joshua B. Tenenbaum, "Graph approximations to geodesics on embedded manifolds," Tech. Rep., Department of Psychology, Stanford University, 2000.

[17] Jarkko Venna and Samuel Kaski, "Neighborhood preservation in nonlinear projection methods: An experimental study," in *Proceedings of ICANN 2001, International Conference on Artificial Neural Networks*, G. Dorffner, H. Bischof, and K. Hornik, Eds., Berlin, 2001, pp. 485–491, Springer.