'DiPPP' Online Self-Improving Linear Map for Distance-Preserving Data Analysis

M. Strickert[1a], **S. Teichmann**[2], **N. Sreenivasulu**[1b], and **U. Seiffert**[1a]
[1] IPK-Gatersleben: [a] Pattern Recognition Group, [b] Gene Expression Group
[2] University of Osnabrück
[1] **{stricker,srinivas,seiffert}@ipk-gatersleben.de**, [2] **steichma@uos.de**

Abstract - *Projection Pursuit (PP) is a powerful yet undervalued dimension reduction technique that provides linear projections of high-dimensional data into a low-dimensional target space. Thereby, the projection axes are chosen by optimization of an index function which aims, in unsupervised scenarios, for example at a central mass view or at maximum data spreading. Here, an efficient index function is developed for unsupervised optimum distance reconstruction between the source data and their projections. The distance relationships resulting from the proposed distance-preserving projection pursuit (DiPPP) are much more faithfully maintained than by principal component analysis (PCA), and, additionally, DiPPP can be applied to huge data sets or even to data streams. The power of DiPPP is demonstrated for dimension reduction of a large set of gene expressions describing developmental series of cDNA experiments with 11,786 genes.*

Key words - projection pursuit, clustering, gene expression analysis.

1 Introduction

First stages of raw data analysis are often accompanied by only poor prior knowledge about the given data. In these cases, dimension reduction techniques help simplifying data complexity and they support visualization for revealing structural dependencies within the data. If only a few continuous-valued attributes from a few experiments are under consideration, scatter plots provide the most direct data displays. For high-dimensional input, principal component analysis (PCA) is one of the most prominent linear projection techniques, which defines interesting projection dimensions mathematically as the directions of largest variance [4]. Principal curves and surfaces are one- and two-dimensional non-linear counterparts of PCA that pass through the curved 'middle' of the original data manifold, onto which data are projected [3]. The self-organizing map (SOM) is a well-established neural alternative for getting non-linear clusterings of data on a low-dimensional rectangular or a hexagonal lattice of specialized neurons, thereby preserving the original data topology. This lattice can be nicely visualized as U-matrix or as attribute planes [5]. The visualization induced SOM (ViSOM) is an extension of SOM aiming at integrating distance preservation in addition to topology preservation [10]. Alternative to SOM, locally linear embeddings (LLE) and Isomap have been proposed for taking into account the properties of data neighborhoods or the underlying data manifolds [7, 9]. These two methods combine different computational

modules ranging from local PCA of k-means clusters to finding shortest paths in the data. Multidimensional scaling (MDS) [6] is another powerful visualization tool for fixed data sets, which operates on the distance matrices of the original data and of their low-dimensional reconstructions, usually imposing high memory and computing load, although an efficient high-throughput MDS implementation (HiT-MDS) has been proposed recently [8]. However, there is still much potential in linear data models, because once a mapping is computed, it yields very fast projections of unknown data and, additionally, it can be used for factor analysis. Projection pursuit PP refers to a bunch of linear projections which result from optimizing a user-defined projection index [1, 2]. For labeled data, the optimization goal might be the one to project data into the visualization plane in such a way that the entropy is minimum, i.e. classes are best separated. For unlabeled data, the criterion might be maximum spatial spreading or, on the contrary, the projection onto the central data mass. Obviously, the choice of the projection index allows to focus on different properties of the data. However, in order to obtain the most faithful linear projection, the notion of 'faithfulness' must be defined carefully. The unsupervised approach presented in the following is very intuitive: a projection is said to be faithful, if the distances between the projected points correlate best with the original distances. Like the original SOM, the proposed distance-preserving projection pursuit (DiPPP) works in an iterative training manner for obtaining faithful data mappings. But in contrast to the prototype-based SOM, DiPPP maintains the individuality of each input point and still yields strikingly compact models.

2 Distance-Preserving Projection Pursuit (DiPPP)

The linear projection of q-dimensional patterns $\mathbf{x}^i \in \mathbb{R}^{n\times q}$ to points $\hat{\mathbf{x}}^i = (\hat{x}^i_1, \ldots, \hat{x}^i_d)$ in the d-dimensional target space $\mathbb{R}^{n\times d}$ is given by:

$$\hat{\mathbf{X}} = \mathbf{X} \cdot \mathbf{P}, \quad \mathbf{P} \in \mathbb{R}^{q\times d}. \tag{1}$$

The complete projection model is thus determined by the parameters $\boldsymbol{p}_{lk}$ of the projection matrix $\mathbf{P}$, and the challenge of projection pursuit lies in finding these free parameters. For distance preservation between input and output space, the most canonic approach to forcing the mutual distances $\hat{\mathrm{d}}_{ij} = \mathrm{d}(\hat{\mathbf{x}}^i_\mathbf{P}, \hat{\mathbf{x}}^j_\mathbf{P})$ of all data pairs projected by $\hat{\mathbf{x}}^i_\mathbf{P} = \mathbf{x}^i \cdot \mathbf{P}$ to best match the original distances $\mathrm{d}_{ij} = \mathrm{d}(\mathbf{x}^i, \mathbf{x}^j)$ is defined by minimizing the raw 'stress' function

$$\mathrm{s} = \sum_{i=1}^{n}\sum_{j=1}^{n}(\mathrm{d}_{ij} - \hat{\mathrm{d}}_{ij})^2 \stackrel{!}{=} \min \quad \text{with distances} \quad \mathrm{d}_{ij}(\mathbf{x}^i, \mathbf{x}^j) = \sum_{k=1}^{d}(x^i_k - x^j_k)^2.$$

For computational convenience, the squared Euclidean distance can be taken and symmetry $\mathrm{d}_{ij} = \mathrm{d}_{ji}$ can be assumed for the distance matrix. Formally, the above stress function looks like Sammon's nonlinear mapping, the straightforward approach to multidimensional scaling (MDS). Here, however, the points $\hat{\mathbf{x}}^i$ cannot be moved freely in the d-dimensional target space, but they are constrained by the projection mapping $\mathbf{P}$, which implies to distances $\hat{\mathrm{d}}_{ij}$, and thus to the MDS scheme, a further level of indirection. Since the derivative $\partial\mathrm{d}/\partial\hat{\mathbf{X}}$, being optimized to 0 during the minimization of Sammon's stress function, is a factor of the projection pursuit optimization

$$\frac{\partial \mathrm{s}}{\partial \mathbf{P}} = \frac{\partial \mathrm{d}}{\partial \hat{\mathbf{X}}} \cdot \frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{P}} \stackrel{!}{=} 0 \quad \text{with} \quad \frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{P}} = \mathbf{X} \text{ (cf. Eqn. 1)},$$

finding optimum $\mathbf{P}$ is at least as difficult as optimum $\hat{\mathbf{X}}$ in Sammon's mapping. Therefore, in the following, the original goal of minimum square distance reconstruction is relaxed to a stress function for finding maximum Pearson correlation $\mathsf{r} \in [-1;1]$ between the pattern distances $\mathbf{D} = (\mathrm{d}_{ij})_{i,j=1\ldots n}$ and those of the reconstructions $\hat{\mathbf{D}} = (\hat{\mathrm{d}}_{ij})_{i,j=1\ldots n}$:

$$\mathsf{r}(\mathbf{D},\hat{\mathbf{D}}) = \frac{\sum_{i\neq j}^{n}(\mathrm{d}_{ij}-\mu_{\mathbf{D}})\cdot(\hat{\mathrm{d}}_{ij}-\mu_{\hat{\mathbf{D}}})}{\sqrt{\sum_{i\neq j}^{n}(\mathrm{d}_{ij}-\mu_{\mathbf{D}})^2}\cdot\sqrt{\sum_{i\neq j}^{n}(\hat{\mathrm{d}}_{ij}-\mu_{\hat{\mathbf{D}}})^2}} =: \frac{\mathscr{B}(\hat{\mathrm{d}})}{\sqrt{\mathscr{C}}\cdot\sqrt{\mathscr{D}(\hat{\mathrm{d}})}} \stackrel{!}{=} \max\,.$$

This expression allows corresponding Shepard plots with points $(\mathrm{d}_{ij},\hat{\mathrm{d}}_{ij})$ to target via $\hat{\mathrm{d}}_{ij}$ at lines of arbitrary gradient, not only to the unit diagonal imposed by Sammon's mapping. The right fraction is a convenient one-to-one correspondence to the sums in the left hand term, which will be used in the following: $\mathscr{B}(\hat{\mathrm{d}})$ is related to the mixed summation of both original and reconstructed distances, $\mathscr{D}(\hat{\mathrm{d}})$ refers to the dissimilarities dependent on the choices of the reconstructions $\hat{\mathbf{X}}$, and $\mathscr{C}$ denotes the connection to the initially calculated, thus constant, input pattern distances. The dependence on $\hat{\mathrm{d}}$ is linked via $\hat{\mathbf{X}}$ to the parameters $\mathbf{P}$ of interest. Using, for better convergence, the transformation of r into a power function to be minimized as discussed in [8], we consider the stress function $\mathrm{s} = \big(\mathsf{r}(\mathbf{D},\hat{\mathbf{D}})\big)^{-2K}$ with integer exponents K. Euclidean distances are taken for getting faithful data visualizations:

$$\hat{\mathbf{D}}\big(\hat{\mathbf{X}}(\mathbf{P})\big) = \Big(\sqrt{\textstyle\sum_{k=1}^{d}(\hat{x}^i_{\mathbf{P}k}-\hat{x}^j_{\mathbf{P}k})^2}\Big)_{\substack{j\neq i,\\ i,j=1\ldots n}} \quad\text{with}\quad \hat{x}^u_{\mathbf{P}k} = \sum_{l=1}^{q} x^u_l\cdot p_{lk}\,.$$

Thus, distance relationships coded by $\mathbf{D}$ are reconstructed by minimizing inverse powers r^{-2K} in order to obtain optimum parameters for projecting the data into the Euclidean target space. The minimization is achieved by a gradient descent on the stress function s, which requires finding zeros of the derivatives of s w.r.t. to the free parameters $p_{lk}, l = 1\ldots q, k = 1\ldots d$:

$$\mathrm{s} = \mathsf{r}^{-2K}\Big(\hat{\mathbf{D}}\big(\hat{\mathbf{X}}(\mathbf{P})\big)\Big) \stackrel{!}{=} \min \;\Rightarrow\; \frac{\partial \mathrm{s}}{\partial p_{lk}} = \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\partial \mathsf{r}^{-2K}}{\partial \hat{\mathrm{d}}_{ij}}\cdot\frac{\partial \hat{\mathrm{d}}_{ij}}{\partial \hat{x}^i_{\mathbf{P}k}}\cdot\frac{\partial \hat{x}^i_{\mathbf{P}k}}{\partial p_{lk}} \stackrel{!}{=} 0 \tag{2}$$

Solutions are found by iterative updates $\Delta p_{lk} = -\gamma\cdot\frac{\partial \mathrm{s}}{\partial p_{lk}}$ of step size γ into the direction of the steepest gradient of s. The missing derivatives in Eqn. 2 are

$$\begin{aligned}\frac{\partial \mathsf{r}^{-2K}}{\partial \hat{\mathrm{d}}_{ij}} &= \frac{\partial\frac{(\mathscr{C}\cdot\mathscr{D}(\hat{\mathrm{d}}))^K}{\mathscr{B}(\hat{\mathrm{d}})^{2K}}}{\partial \hat{\mathrm{d}}_{ij}} = -2\cdot K\cdot\mathsf{r}^{-2K-1}\cdot\frac{(\hat{\mathrm{d}}_{ij}-\mu_{\hat{\mathbf{D}}})\cdot\mathscr{B}(\hat{\mathrm{d}})-(\mathrm{d}_{ij}-\mu_{\mathbf{D}})\cdot\mathscr{D}(\hat{\mathrm{d}})}{\mathscr{B}(\hat{\mathrm{d}})^3}\,,\\ \frac{\partial \hat{\mathrm{d}}_{ij}}{\partial \hat{x}^i_{\mathbf{P}k}} &= 2\cdot(\hat{x}^i_{\mathbf{P}k}-\hat{x}^j_{\mathbf{P}k})\Big/\sqrt{\textstyle\sum_{l=1}^{d}(\hat{x}^i_{\mathbf{P}l}-\hat{x}^j_l)^2}\;, \qquad \frac{\partial \hat{x}^i_{\mathbf{P}k}}{\partial p_{lk}} = x^i_l\,.\end{aligned}$$

The direct implementation of Eqn. 2 with the update formulas from Eqn. 3 for all $q\times d$ entries p_{lk} and for all pattern pairs $\mathbf{X}^i, \mathbf{X}^j$ would be computationally very costly. Therefore an approximating online version is proposed in the following.

3 Online DiPPP

A straightforward simplification of Eqn. 2 is obtained by replacing the total integration over all pattern pairs by stochastic pattern presentation: the projection matrix $(p_{lk})_{l=1\ldots q,k=1\ldots d}$ is

iteratively updated for randomly drawn pairs $(\mathbf{x}^i, \mathbf{x}^j)$. A crucial step towards high-throughput processing is the internal data storage in DiPPP: the projection pursuit of n data items requires a storage capacity of order $\mathcal{O}(n^2)$ for the mutual distances; therefore, computations are reduced to a small online working memory with $L \leq n$ data points. In the following, symmetric $\mathbb{R}^{L\times L}$ distance matrices are assumed, referring to a subset of already presented data points and their projections. The general processing consists of $\langle 1\rangle$ presenting a new data item, $\langle 2\rangle$ projecting that item via $\mathbf{P}$, and $\langle 3\rangle$ minimizing the discrepancy of its distances to a few N other projections in the working memory with the corresponding source data distances. This way, the current pattern is related to what is stored in memory. It has turned out that, using many training iterations, relating this pattern to a single pattern randomly drawn from memory suffices for obtaining good results.

The general iterative procedure for finding distance-preserving linear projections is outlined in Algorithm 1. Optimum projection values are obtained in a neural network training manner by repetitive presentation of input data. Instead of the random projection initialization, PCA eigenvectors could be used, or p_{lk} values from previous runs. In batch mode, one would present all available data for several cycles and print the projected data and the projection parameters after training; in online mode, one would, after a reasonable pre-training, print the currently calculated projections immediately at pattern presentation time.

Algorithm 1 DiPPP

1: Read first L input data points $\mathbf{X} \in \mathbb{R}^{L\times q}$ to working memory.
2: Initialize $\mathbf{P} \in \mathbb{R}^{q\times d}$ randomly with $p_{lk} \in (-\frac{1}{L}; \frac{1}{L})$, or read initial p_{lk} from file.
3: **repeat**
4: Draw next pattern $\mathbf{x}$, overwrite random position $\mathbf{x}^i \leftarrow \mathbf{x}$ in working memory.
5: Project data by $\hat{\mathbf{X}}_\mathbf{P} \leftarrow \mathbf{X} \cdot \mathbf{P}$. If online mode, print $\hat{\mathbf{x}}^i_\mathbf{P}$.
6: Calculate distance matrices $\mathbf{D} \in \mathbb{R}^{L\times L} \Rightarrow \mathscr{C}$ and $\hat{\mathbf{D}} \in \mathbb{R}^{L\times L} \Rightarrow \mathscr{B}(\hat{\mathrm{d}}), \mathscr{D}(\hat{\mathrm{d}})$.
7: **for** N times **do**
8: Draw a random index $1 \leq j \neq i \leq L$ pointing to $\mathbf{x}^j, \hat{\mathbf{x}}^j_\mathbf{P}$ in working memory.
9: **for all** $(l,k) \in [1;q] \times [1;d]$ **do**
10: $p_{lk} \leftarrow p_{lk} - \frac{\partial \mathsf{r}^{-2K}}{\partial \hat{\mathrm{d}}_{ij}} \cdot \frac{\partial \hat{\mathrm{d}}_{ij}}{\partial \hat{x}^i_{\mathbf{P}k}} \cdot \frac{\partial \hat{x}^i_{\mathbf{P}k}}{\partial p_{lk}}$ { adapt p_{lk} according to Eqn. 2 }
11: **end for**
12: **end for**
13: **until** convergence criterion is met or no more data available.
14: If batch mode, project all data; center and normalize by largest dimension variance.

Empiric studies on different data sets have showed that the proposed algorithm works good with $\gamma \in [0.0001; 0.1]$, exponent $K = 1, 2$. The total number of iterations depends on the complexity of the input data and on the choice of the working memory size L. If there exist reasonable projections, however, a number between 1,000 and 100,000 iterations is a first good rule of thumb, even for $N = 1$ (cf. Alg. 1, l. 7), as mentioned earlier. Furthermore, by tracking the development of distance correlations $\mathsf{r}(\mathbf{D}, \hat{\mathbf{D}})$ between input data and their projections during training, the speed and the quality of convergence can be estimated. This is useful information for choosing both the iteration number and the learning rate appropriately.

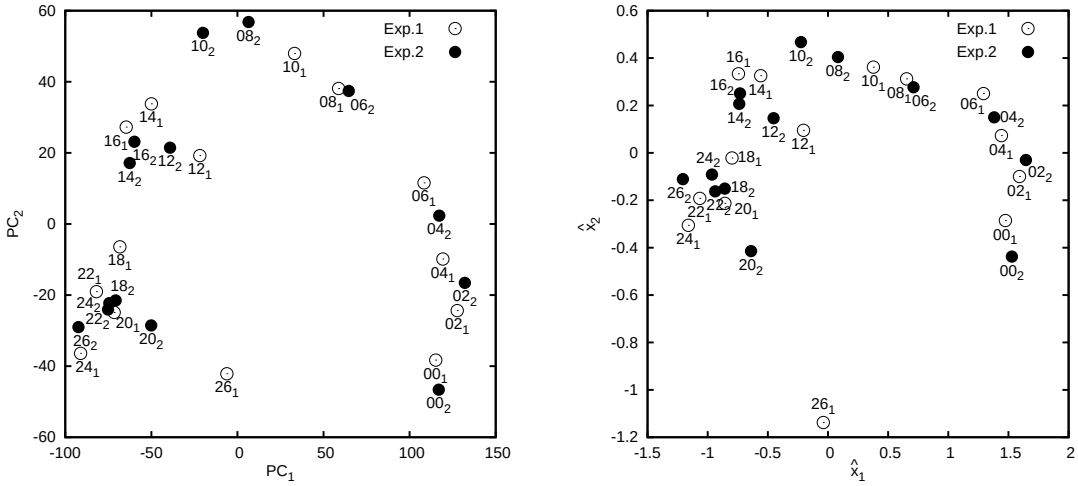

Figure 1: Inter-experiment relationships projected into 2D based on the gene expression patterns. Left: PCA, computed by SVD because of huge data dimensionality. Right: DiPPP.

4 Application to cDNA Array Expression Data

Two studies are presented that show the suitability of DiPPP for the analysis of high-throughput expression data. In the experimental design, several thousand gene expression patterns were analyzed corresponding to barley seed development, 0–26 days after flowering, in three distinct tissues: pericarp (p), endosperm (en), and embryo (e). The two major questions of interest are: 1. How are the experiments, representing the tissues at a particular developmental stage, characterized with respect to their transcriptome similarity of specifically expressed genes? 2. Can clusters of genes be identified and are they identical for repeated series of experiments? These key questions can be addressed by using DiPPP for dimension reduction by faithful projections into visualizable planes.

1. Experiment Clustering The first DiPPP application aims at the visualization of the inter-experiment relationships: a two-dimensional projection is wanted that preserves the distances of the original space of gene expression intensities. The common way to do this is principal component analysis (PCA) which, like DiPPP, is also a linear projection technique. The analyzed data set comprises two series of 14 experiments for developmental stages 0...26 days after flowering (DAF) in barley endosperm tissue in steps of 2 days, and the final matrix of interest consists of 28×11786 logarithmic gene expression values.

Fig. 1 compares the experiment visualization of PCA (left) and DiPPP (right). Both projections show a temporal ordering of endosperm development from 00 to 26 DAF in a anticlockwise manner, starting at the lower right. DiPPP has been trained after random initialization in batch mode with $L = n = 28$, $\gamma = 0.005$, and $K = 2$ for 2,500 iterations, taking about $45s$ on a 3GHz-P4 machine, thereby adapting the 11786×2 projection parameters; due to high memory requirements, PCA is calculated by `linpack` as singular value decomposition (SVD) within $3s$. Squared distance correlations characterize the display quality; they are $\mathsf{r}^2(\mathbf{X}, \hat{\mathbf{X}}_{\mathsf{PCA}}) = 0.957$ and $\mathsf{r}^2(\mathbf{X}, \hat{\mathbf{X}}_{\mathsf{DiPPP}}) = 0.986$, indicating the superior distance preservation of the DiPPP projection. Both panels in Fig. 1 show grouping of endosperm development into distinct phases. For instance, pre-storage (0–4 DAF) is separated by a transition phase with its biologically significant genetic reprogramming (6–10 DAF), followed by mid-storage phase

(12–16 DAF) with final maturation/desiccation phase (18–26 DAF). Although the overall reproducibility between two independent experiments is high, the slight discrepancy could be observed at the beginning of the intermediate phase at 6 DAF. Additionally, 26 DAF of experiment 1 is identified as outlier by both methods, and emphasized by DiPPP. After all, the obtained DiPPP experiment groupings are very close to the best results of distance-preserving multidimensional scaling with $\mathsf{r}^2(\mathbf{X}, \hat{\mathbf{X}}_{\mathsf{MDS}}) = 0.988$ for the HiT-MDS method [8]. For the obtained DiPPP clusters, the identification of regulating key genes through factor analysis of the the projection matrix $\mathbf{P}$ will be of great interest in future investigations.

2. Experimental Reproducibility of Gene Clustering A much more demanding experiment is conducted with the extended version of the previous data set, now including gene expressions of the pericarp and the embryo tissues in addition to the endosperm, adding up to 31 developmental stages of barley. However, this data matrix is transposed in order to focus on the clustering of similar genes in contrast to the previous grouping of similar experiments. A linear projection has been calculated in online mode for the first series of experiments, and the resulting parameters have been applied to the second series. If both series were identical, no major differences should appear in the projections, which has been investigated. In order to visualize the thousands of projected 31-dimensional points, density plots are shown in Fig. 2. The left panel shows the projected genes for the first series of experiments, dark spots indicating high densities of similar gene profiles. While the PCA projection yields a distance preservation of $\mathsf{r}^2(\mathbf{X}, \hat{\mathbf{X}}_{\mathsf{PCA}}) = 0.878$, DiPPP reaches $\mathsf{r}^2(\mathbf{X}, \hat{\mathbf{X}}_{\mathsf{DiPPP}}) = 0.911$ for $\gamma = 0.025, K = 1$, a small working memory size of $L = 250$ after 58,930 iterations, i.e. 5 cylces, in about 2 minutes. The final dimension reduction model, consisting of only 31×2 projection values, is applied to the second series of experiments; this yields a slightly worse squared correlation of 0.898, which is still better than the PCA model of the same size, but which already indicates a difference between the series. More than mere indication is seen in the normalized density difference plot in the right panel of Fig. 2, zoomed to the box shown in the left. Gray means equal gene densities for both experiments, which is found for the major proportion in the image, thus pointing out a good overall correspondence between the two experiments. In addition, certain discrepancies can be observed: white patches denote a bias towards experiment 1, and black a bias towards experiment 2. A systematic difference, roughly depicted by the straight line, is found. The two high-difference clusters C1 and C2 are related to the experiment-specific gene profiles inside the circles, their average shapes are displayed in Fig. 3. Although the two profile 'prototypes' are very similar, the one for experiment 1 shows a distinct peak at day two for pericarp tissue (02p), which is a difference confirmed by other methods. In the present gene projection task, DiPPP yields a good and faithful visualization of the general distribution of the genes in their 31-dimensional expression space. Fast gene mappings are obtained with a tiny projection model.

5 Conclusions and Outlook

The proposed distance-preserving projection pursuit DiPPP is useful for learning faithful linear mappings for which the mutual distances of the projections aim at matching the original distances. A large set of gene expression data has been processed to show the general suitability of DiPPP for batch learning of experiment groupings, and for online learning of gene profile mappings. DiPPP has turned out to yield better distance preservation than PCA, if distance correlations between source data and the corresponding projections are considered.

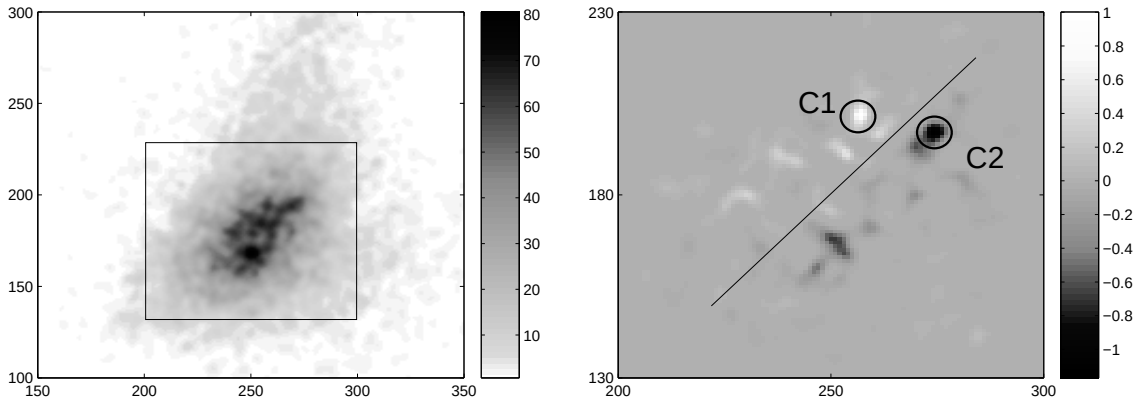

Figure 2: Genes DiPPP-projected to 2D. Left: density plot of Euclidean projections of the 11,786 genes from the first series of experiments. Right: densities of the projected second series subtracted from the first image, zoomed to the box shown left. Overall differences are close to zero (gray), indicating high correspondence of experimental reproductions. However, encircled areas C1 and C2 point out a clear bias towards experiment 1 and 2, respectively. The related expression profiles are given in Fig. 3.

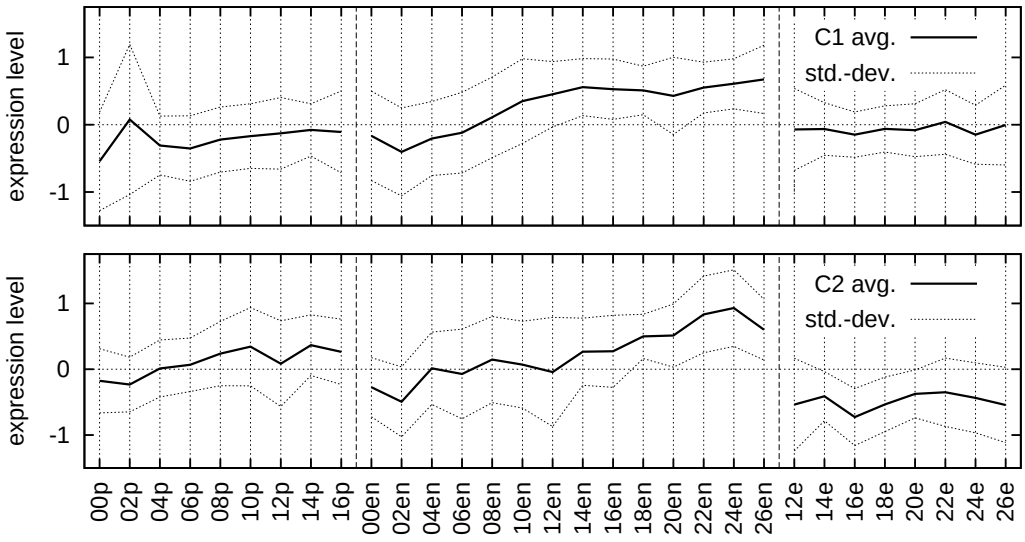

Figure 3: Experiment-specific gene profiles in clusters C1 and C2 related to Fig. 2 given by average and standard deviation. Top: cluster C1 containing 163 genes. Bottom: cluster C2 with 175 genes. Both groups of genes refer to up-regulation of endosperm gene expression levels, emphasizing that areas close in the projection Fig. 2 map back to similar genes profiles. Anyhow, C1 shows a distinct peak at 02p and C2 exhibits specifically low embryo expression levels.

Thus, DiPPP is favored over PCA in cases where the projection should not be biased towards dimensions of maximum data variance, as PCA does by its definition. DiPPP is linear and at the same time it aims at distance preservation; in these respects, DiPPP is located between PCA and MDS, also with respect to the quality of distance preservation. The problem of getting stuck in local minima does not exist for PCA; in DiPPP, however, local minima are counteracted by using a correlation-based, well-converging projection constraint. Although MDS-embeddings yield better low-dimensional models for fixed data sets, the benefit of the linear distance-preserving DiPPP-method is easy inclusion and projection of new data. Thus, as SOM, DiPPP is useful in many applications where easily interpretable data projections are desired, such as for dimension reduction and visualization; in contrast to SOM, DiPPP training is faster, models are generally smaller and only linear, but all data points maintain their individuality in the absence of neuron competition. Furthermore, DiPPP can be used for converting similarity measures into Euclidean distances; the currently used Euclidean input distance metric d_{ij} can just be replaced by any data similarity measure, such as general Minkowski metrics, the cosine distance, or the Pearson correlation. Although the corresponding similarity-preservation found by DiPPP might be poor due to its simple linear projection model, still valuable data groupings can be obtained; this is currently studied and yet tentatively confirmed for correlation-based color space conversion for enhancing microscope images. A couple of questions will be addressed in the future regarding the choice of the working memory size L and the learning rate γ. Moreover, factor analysis will be considered by selecting from $\mathbf{P}$ only those rows that contain information required to reconstruct by projection best the original data. Further applications in bioinformatics are currently surveyed.

Acknowledgments Thanks to Dr. Volodymyr Radchuk for helping probe preparation for the macroarray experiments. The work is supported by BMBF grant FKZ 0313115, GABI-SEED-II.

References

[1] D. Cook, A. Buja, and J. Cabrera. Grand tour and projection pursuit. *Journal of Computational and Graphical Statistics*, 4(3):155–172, 1995.

[2] J. Friedman and J. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, 9(23):881–889, 1974.

[3] T. Hastie. *Principal Curves and Surfaces*. PhD thesis, Stanford University, 1984.

[4] I. Jolliffe. *Principal component analysis*. Springer Verlag, New York, 1986.

[5] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, 3rd edition, 2001.

[6] J. Kruskal and M. Wish. *Multidimensional Scaling*. Sage Publications, New Park, Ca., 1978.

[7] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[8] M. Strickert, S. Teichmann, N. Sreenivasulu, and U. Seiffert. High-Throughput Multi-Dimensional Scaling (HiT-MDS) for cDNA-Array Expression Data. Accepted at ICANN 2005.

[9] J. Tenenbaum, V. deSilva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[10] H. Yin. Data visualisation and manifold mapping using ViSOM. *Neur. Net.*, 15:1005–1016, 2002.