

Assessing Gene Expression Measurements

Lídia Rejtő^{1,2} and Gábor Tusnády¹

July 1, 2002

Abstract

Gene array studies enable assessment of expression patterns of thousands of genes over time and under multiple conditions. The analysis of these patterns requires detecting whether observed differences in expression levels are significant or not. To perform the analysis, we must first normalize the data. Normalization is the term used to describe the process of removing differences of measurements caused by certain non-reproducibility.

We analyzed our data by non-parametric and parametric methods and used different goodness of fit criteria in order to find the normalized data. One of the method is a parametric likelihood framework for microarray data analysis where we introduced a Bayesian component; it is supposed that for each gene there is a time-dependent probability of different expression levels under different conditions. In this model, the probability of different expression levels for individual genes can be estimated.

We included missing data in the model, using appropriate techniques. In order to validate the parametric likelihood model and the parameters of the model in the actual data set we used different estimation techniques. Furthermore, we used bootstrapping to estimate the variances of the estimated parameters. Based on the parametric likelihood model, we developed clusters of gene expression levels using principal component analysis, k-mean clustering and variance clustering techniques.

1. Introduction

Gene expression data is generated by microarray techniques (i.e. DNA chips) and the data are often presented as arrays of expression levels of genes under different conditions (developmental stages, disease states, phenotypes, etc...). The usual goal of expression data analysis is to detect the role of different genes in manifestation of characters of the conditions under investigation.

We shall label the different conditions and genes by positive integers. K stands for the number of investigated conditions and L for the number of genes. The expression level of a gene may change with time under fixed condition, thus expression levels are considered as function of time. For each genes we have gene expression measurements at M different times under each condition. For the sake of simplicity, we will omit the time index in most cases.

Gene expression forms a 3-dimensional array $X(k, \ell, m)$ where $k = 1, \dots, K$, $\ell = 1, \dots, L$, $m = 1, \dots, M$. This quantity is virtual: it may be the protein concentration coded by the gene in the cells of the investigated tissue of an organism or it may be the mRNA in one cell responsible for the activity of the specific gene. Unfortunately gene expression can not be measured directly. In the experiment a subset of possible triplets

¹Alfréd Rényi Mathematical Institute of the Hungarian Academy of Sciences, Budapest, Hungary

²Statistics Program, University of Delaware, Newark, DE, USA

(k, ℓ, m) is chosen. The result of the experiment is $Y(k, \ell, m) = U(X(k, \ell, m))$, for which U is a stochastic function containing the effect of measurement process as well.

In this paper we introduce stochastic models appropriate to analyze gene expression data. The purpose is twofold: to find differently expressed genes by conditions at a fixed time and to find clusters of similarly expressed genes over time.

In the analyzed data we had 696 genes; which were printed as duplicate spots on the microarray, with the number of different genes at 410. Gene expression was measured under two different conditions at four different times. At least one censored measurement occurred in 283 cases. The expression level measurements were censored; over- or under-expressed genes were not measured. Genes are not expressed at certain times or conditions, but expression levels measured at least once are used in the analysis. We used a model-based approach for dealing with missing data.

We analyzed the data by non-parametric and parametric methods and used different goodness of fit criteria in order to normalize the data. We used principal component analysis (PCA) and k -mean clustering to find clusters of genes. For gene expression data PCA or SVD clustering is a known method, and it was used by Alter et al. (2000), Troyanskaya, O. et al. (2001). Other methods including SOM, hierarchical clustering, were used in papers Brown M. et al (2000), Herrero, J. et al. (2001), Manduchi, E. et al. (2000), Tamamyo et al. (1999), Hastie, T et al. (2001).

The paper contains four sections. Section 2. describes the stochastic model and basic assumptions, Section 3. gives the technical details of the statistical methods used to estimate parameters and normalize the data, finally Section 4. describes the clusters and the results.

2. The Model

We describe the result of the experiment Y by some stochastically monotone stochastic function, that is

$$Y(k, \ell, m) = U(X(k, \ell, m)) = H(S(X(k, \ell, m), \theta) + Z), \quad (1)$$

where θ is the parameter vector depending on the specific microarray, Z denotes a Gaussian variable with 0 expectation and standard deviation σ , which may depend on time. The function H is a truncation, there are two cutoffs B_L and B_U ,

$$H(v) = \begin{cases} C^L & \text{if } v < B^L \\ v & \text{if } B^L \leq v < B^U \\ C^U & \text{if } B^U \leq v, \end{cases} \quad (2)$$

where C^L and C^U are the codes of under or over expressed genes. In certain cases C^L and C^U may coincide.

Let us denote the number of microarrays by N and by A_n the subset of triplets (k, ℓ, m) corresponding to the n th microarray plate ($n = 1, \dots, N$). The sets A_n are disjoint and $\bigcup_{n=1}^N A_n = A$, where A is the set of all possible triplets. The number of elements of A is equal to KLM . The experiments are highly parallel, with thousands of gene expression measurements made simultaneously for each specific condition and time, accordingly $N = KM$. On each microarray, expression level is measured under the same condition at the same fixed time. Repetitions for one triplet (k, ℓ, m) may also be included in the experimental design; i.e., genes may be printed more than once on the microarrays. It is worth noting that a more sophisticated use of repetitions over different microarray plates would have some advantages in fixing the non-parametric components of the model.

We shall differentiate two types of genes:

- identical genes, for which the original gene expression level does not depend on the condition

$$X(k, \ell, m) = X_0(\ell, m) \quad k = 1, \dots, K;$$

– independent genes, for which the original gene expression levels under different conditions at some time m are independent.

The model is based on the simple fact that for identically expressed genes the expression levels follow a monotone order. Since original gene expression level is a virtual quantity we will fix its distribution, using standard Gaussian distribution. The random variables describing the virtual processes of gene expression levels are independent of the measurement error variable Z and the measurement errors are independent each other.

Given the parameters $(\theta_1, \sigma_1), \dots, (\theta_N, \sigma_N)$ of measurements of gene expressions, the profile function S , cut levels B_L, B_U and memory parameter ρ the likelihood of the entire set of data depends on the logical variables $\delta(\ell, m)$, $\ell = 1, \dots, L$, $m = 1, \dots, M$ expressing the type of genes. Set

$$\delta(\ell, m) = \begin{cases} 0 & \text{for identical expression level by condition,} \\ 1 & \text{for independent expression level by condition.} \end{cases}$$

To find the double array

$$\Delta = (\delta(\ell, m), \ell = 1, \dots, L, m = 1, \dots, M) \quad (3)$$

is the ultimate goal of the investigation. We suppose that there are time independent 0–1 variables Δ_ℓ , $\ell = 1, \dots, L$ with identical probability distribution

$$P(\Delta_\ell = 0) = 1 - p, \quad P(\Delta_\ell = 1) = p, \quad \ell = 1, \dots, L, \quad (4)$$

where p is the probability that gene ℓ has the ultimate liability for independent expression under different conditions. Furthermore once Δ_ℓ is given the random variables $\delta(\ell, m)$, $m = 1, 2, \dots, M$ are conditionally independent with distributions

$$\begin{aligned} P(\delta(\ell, m) = 0 \mid \Delta_\ell = 0) &= 1 \\ P(\delta(\ell, m) = 0 \mid \Delta_\ell = 1) &= 1 - q_m & P(\delta(\ell, m) = 1 \mid \Delta_\ell = 1) &= q_m \end{aligned} \quad (5)$$

where q_m is the conditional probability that at time m gene ℓ has independent expression levels.

Gene expressions $X(k, \ell, m)$ and corresponding logical variables $\delta(\ell, m)$, Δ_ℓ are hidden variables with parameters p, q_m . The relation of the measured expressions $Y(k, \ell, m)$ to the virtual expressions $X(k, \ell, m)$ is described with parameters $\theta(k, m)$ of profile function S , $\sigma(k, m)$ standard deviation of measurement errors and cut-points $B_{k,m}^L, B_{k,m}^U$ and r_m the probability of independent expression at time m . Decomposition of mixtures is a statistical method for estimating hidden variables.

3. Methods

3.1. DECOMPOSITION OF MIXTURES

In stochastic investigations the simplest situation consists of observing a complete system of events or partitions (A_1, A_2, \dots, A_I) . Suppose that the probability of the event A_i is $p_i = P(A_i)$, $i = 1, \dots, I$. The simplest statistical task is to estimate the vector (p_1, \dots, p_I) of unknown probabilities based on experiments. There is only one way to do that: counting relative frequencies, which is possible as long as the events A_i -s are directly observable. The situation changes completely if the events A_i -s are hidden and we are offered only the possibility of observing the elements of another complete system of events B_1, B_2, \dots, B_J . Suppose that luckily we know the conditional probabilities $q_{i,j} = P(B_j \mid A_i)$ and the problem is how can we use this information in order to estimate $p_i = P(A_i)$. Observed frequencies $\nu_1, \nu_2, \dots, \nu_J$ of events B_1, B_2, \dots, B_J provide s_j an estimate of $P(B_j)$:

$$s_j = \frac{\nu_j}{\nu_1 + \nu_2 + \dots + \nu_J}, \quad j = 1, \dots, J.$$

For unknown p_1, \dots, p_I prior probabilities the joint probability of the event $A_i B_j$ is

$$P(A_i B_j) = r_{i,j} = p_i q_{i,j} = P(A_i)P(B_j|A_i).$$

The double array $r_{i,j}$, $i = 1, \dots, I$, $j = 1, \dots, J$ has two marginal distributions

$$r_{i,\bullet} = \sum_{j=1}^J r_{i,j}, \quad r_{\bullet,j} = \sum_{i=1}^I r_{i,j}.$$

Unfortunately in most cases the marginal distribution $r_{\bullet,1}, r_{\bullet,2}, \dots, r_{\bullet,J}$ is different of the observed relative frequencies s_1, s_2, \dots, s_J for any choice of the prior probabilities p_1, p_2, \dots, p_I . The problem of estimating the probabilities p_i 's from the s_j 's is called *decomposition of mixtures*, because the set of probabilities of the events $P(B_j)$ is the mixture of the conditional probabilities $P(B_j|A_i)$

$$P(B_j) = \sum_{i=1}^I P(B_j|A_i)P(A_i) = \sum_{i=1}^I q_{i,j}p_i \quad (6)$$

(see McLachlan,G.J., Krishnan,T. (1997) and Csiszár,I., Tusnády,G. (1984)). In other words the event A_i is transmitted on a noisy channel with possible outputs B_1, B_2, \dots, B_J and knowing the channel response probabilities $q_{i,j} = P(B_j|A_i)$ (i.e. the probabilities that B_j is the output given that A_i is the input) the task is to find the input distribution p_1, p_2, \dots, p_I from the output distribution s_1, s_2, \dots, s_J . This is an inversion problem. Any solution is based on the measure of the distance of $\{s_j\}_{j=1}^J$ from the marginal $\{r_{\bullet,j}\}_{j=1}^J$. A possible measure of distance between probability distributions is the Kullback–Leibler information divergence. DEFINITION: The *Kullback–Leibler information divergence* of two finite probability distributions $P = (p_1, p_2, \dots, p_L)$ and $Q = (q_1, q_2, \dots, q_L)$ is defined as

$$D(P \parallel Q) = \sum_{\ell=1}^L p_\ell \log \frac{p_\ell}{q_\ell},$$

where by definition $0 \log \frac{0}{0} = 0$ and $\varepsilon \log \frac{\varepsilon}{0} = \infty$ for all $\varepsilon > 0$.

There is an iteration resulting in the minimum of $D(\{s_j\}_{j=1}^J \parallel \{r_{\bullet,j}\}_{j=1}^J)$ based on the following two steps.

STEP 1. Projection onto given marginals.

Given the double array $\{r_{i,j}\}$, its best approximation of double arrays $\{\tilde{r}_{i,j}\}$ with fixed marginal $\tilde{r}_{\bullet,j} = s_j$, $j = 1, \dots, J$ is the so called *proportional fitting*:

$$\tilde{r}_{i,j} = r_{i,j} \frac{s_j}{r_{\bullet,j}}.$$

It means that

$$\sum_{i=1}^I \sum_{j=1}^J r_{i,j} \log \frac{r_{i,j}}{\tilde{r}_{i,j}} \geq \sum_{i=1}^I \sum_{j=1}^J r_{i,j} \log \frac{r_{\bullet,j}}{s_j},$$

for any $\{r_{i,j}\}$, $\{s_j\}$, $\{\tilde{r}_{i,j}\}$ such that $\tilde{r}_{\bullet,j} = s_j$.

STEP 2. Projection onto channel model.

Given the double array $\{r_{i,j}\}$, its best approximation with double array in form of $\{\hat{p}_i q_{i,j}\}$ is given by $\hat{p}_i = r_{i,\bullet}$.

It means that

$$\sum_{i=1}^I \sum_{j=1}^J r_{i,j} \log \frac{r_{i,j}}{\hat{p}_i q_{i,j}} \geq \sum_{i=1}^I \sum_{j=1}^J r_{i,j} \log \frac{r_{i,j}}{r_{i,\bullet} q_{i,j}}$$

for any $\{\hat{p}_i\}$, $\{q_{i,j}\}$, $\{r_{i,j}\}$ with $r_{i,\bullet} = \sum_{j=1}^J r_{i,j}$.

We have two sets of double arrays: In the world of theoretical models there are the set \mathcal{P} of double arrays in form of $r_{i,j} = p_i q_{i,j}$. Considering the world of the experiment the set \mathcal{Q} formed by double arrays with given marginal: $r_{\bullet,j} = s_j$, $j = 1, \dots, J$. Minimization of $D(S \parallel Z)$ for probability distributions on the partition B_1, \dots, B_J is given by $S(B_j) = s_j$ and $Z(B_j) = z_j = \sum_{i=1}^I p_i q_{i,j}$ in p_1, \dots, p_I is equivalent to looking for the closest elements of \mathcal{P} and \mathcal{Q} . Unfortunately it can not be done in one simple step. Instead for any element

of \mathcal{P} , Step 1. results in the closest element of \mathcal{Q} , and for any element of \mathcal{Q} Step 2. results in the closest element of \mathcal{P} . Iterating these two steps we get the closest pair.

We start the iteration with an arbitrary input distribution $P = (p_1, p_2, \dots, p_I)$. Together with the given transition probabilities $q_{i,j}$ it results in the double array $r_{i,j} = p_i q_{i,j}$. Applying Step 1. the projection of this double array onto \mathcal{Q} is

$$\tilde{r}_{i,j} = p_i q_{i,j} \frac{s_j}{r_{\bullet,j}}$$

Step 2. gives

$$\tilde{p}_i = \sum_{j=1}^J \tilde{r}_{i,j} = \sum_{j=1}^J p_i q_{i,j} \frac{s_j}{r_{\bullet,j}}$$

This is the posterior distribution given by the Bayes theorem from the prior distribution $P = (p_1, p_2, \dots, p_I)$. Thus the two steps together have a shortcut interpretation: any prior distribution leads to a better input distribution, which is incidentally the posterior one. This interpretation might have a natural flavor, nevertheless the claim that the Bayes theorem gives a better approximation is not immediate, it needs some explanation. A possible explanation is the above decomposition of the Bayes iteration in two steps, another is the reference to EM algorithm (see later).

One advantage of measuring distance between probability measures by divergence is some resemblance to elementary geometry. However an important difference is that divergence is a directed quantity (i.e. $D(P \parallel Q)$ differs from $D(Q \parallel P)$) thus we have to be careful.

Let us return to our iteration procedure. It can be proved that the iteration gives the distance of the sets \mathcal{Q} and \mathcal{P} :

$$\lim_{n \rightarrow \infty} D(Q_n \parallel P_n) = \inf_{P \in \mathcal{P}} \inf_{Q \in \mathcal{Q}} D(Q \parallel P).$$

Let us observe that this statement is weaker than convergence in P_n, Q_n . In practice the iteration is convergent. The limit point P^* is a fixed point of the Bayes iteration. The probability distribution P^* is self-consistent in the sense that starting prior distribution P^* it returns as posterior one in course of the iteration.

The output distribution of the noisy channel may be continuous. This case it is given by density functions $g_i(y)$, $i = 1, \dots, I$ and the mixture distribution is

$$h(y) = \sum_{i=1}^I p_i g_i(y).$$

Having a sample of y_1, \dots, y_n the Bayes theorem implies that

$$\tilde{p}_i(y_j) = \frac{p_i g_i(y_j)}{h(y_j)}.$$

It means that for a given prior distribution (p_1, p_2, \dots, p_I) the sample elements y_j -s have quasi counts $\tilde{p}_1(y_j), \dots, \tilde{p}_I(y_j)$. Summing up the quasi counts we get the continuous version of the Bayes iteration:

$$\tilde{p}_i = \frac{1}{J} \sum_{j=1}^J \tilde{p}_i(y_j).$$

Usually the output distributions have some parameters: $g_i = g_i(y, \theta_i)$. If we have samples $(Z_1, \dots, Z_k) = (Z_{1,i}, \dots, Z_{k,i})$, $i = 1, \dots, I$ for each of the $g_i(y, \theta_i)$ -s, then the parameter θ_i can be estimated by some estimate $\hat{\theta}_i(Z_1, \dots, Z_k)$. In case of ties, different sample elements Z_1, \dots, Z_K are listed with frequencies W_1, \dots, W_k and the estimation may be extended to $\hat{\theta}_i(Z_1, W_1, \dots, Z_k, W_k)$. In case of mixtures quasi-counts $\tilde{p}_i(y_j)$ play the role of frequencies. The sample (Y_1, \dots, Y_J) is virtually cut into I different samples with appropriate quasi-counts and in one step of the iteration the parameters θ_i may be iterated together with the prior distribution.

This is the EM algorithm, a statistical device for estimating parameters of data with missing information. In our case the missing information is the input sign, i.e. the result of the first phase experiment on the original

system of events A_1, A_2, \dots, A_I . In certain cases the original events A_i -s may be endowed with some numerical elements, we may think of parameters θ_i partly as characteristics of noise and partly as characteristic of first phase experiment. In the simplest case θ_i is a location parameter: $g(y, \theta_i) = g(y - \theta_i)$. In this case there is a random variable X taking values $\theta_1, \dots, \theta_I$ with probabilities p_1, \dots, p_I . Let Z be a random variable with density g . The result of the experiment is the convolution $Y = X + Z$. If g is the Gaussian density then the estimation of the location parameter is the average. The Bayes iteration has two parts. We may iterate the probabilities p_1, \dots, p_I and the locations $\theta_1, \dots, \theta_I$ simultaneously or we may prefer to fix the locations and iterate p_1, \dots, p_I or vice versa. If the number of components I is relatively small the locations are well separated and the procedure has a clustering character. In this case it is obvious that the components have different covariance structure.

There is a certain element of danger in the situation: clusters may become irregularly small and the variances are collapsing. In multi-dimensional case the effect may result in dimension reduction: the elements of some clusters may form smaller dimensional set as galaxies form two dimensional subsets in cosmos. In practice we have to use appropriately chosen lower bounds for the eigenvalues of the covariance matrices of clusters.

In other cases we are interested in the distribution of the random variable X which may have continuous distribution. In one dimensional case with additive Gaussian noise the speed of convergence of the estimation is rather slow. The best possible error term is $(1/\sqrt{\log n})$ in case of n sample elements. Interestingly Gaussian noise gives the worst speed. One possible explanation of this phenomena is the excessive smoothing effect of Gaussian noise which makes impossible to discriminate small fluctuations of the density of X . A possible resolution of the situation is using discrete estimation of the unknown, theoretically continuous distribution. Although there is an extensive literature of the deconvolution problem there is no general agreement about the solution. Originally Fourier method was used (Zheng (1990), (1995)) and a kernel-type estimator was proposed. Barbe (1998) advocates minimal distance estimator. We believe that maximum likelihood solution of the problem has the advantage that it is without any artificial elements. In the multi-dimensional case the unknown distribution might have some specific character. It may be concentrated in smaller dimension or it may be monotone.

DEFINITION: A subset \mathcal{S} of the multi-dimensional Euclidean space \mathbf{R}^d is called *monotone* if any pair of elements $x, y \in \mathcal{S}$ can be ordered: either all coordinates of x are smaller than that of y or vice versa.

A distribution is monotone if it is concentrated on a monotone set, i.e. its domain is a monotone set. A naive estimator of the unknown monotone distribution is the monotone regression.

3.2. MONOTONE REGRESSION

Let $I > 1$ be a positive integer and $a(1), a(2), \dots, a(I)$ be arbitrary real numbers. The scalar monotone regression problem is to determine the real numbers $x(1) \leq x(2) \leq \dots \leq x(I)$ such that $\sum_{i=1}^I (x(i) - a(i))^2$ is minimal.

DEFINITION: We say that a block $1 \leq \alpha \leq \beta \leq I$ of the sequence $a(1), \dots, a(I)$ is *rigid* if

$$\frac{1}{\gamma + 1 - \alpha} \sum_{i=\alpha}^{\gamma} a(i) \geq \frac{1}{\beta + 1 - \alpha} \sum_{i=\alpha}^{\beta} a(i)$$

holds true for any positive integer γ with $\alpha \leq \gamma \leq \beta$.

The solution of the scalar monotone regression problem based on disjoint decomposition of the index set $\{1, 2, \dots, I\}$ into rigid blocks such that the sequence of block averages is non-decreasing. The elements $x(1), \dots, x(I)$ of the required sequence equal to the block averages in each block. Rigidity is the sufficient and necessary condition for a constant solution. There are several algorithms for finding a solution. The required number of steps is about $3I$. A natural solution is dynamic evolution of rigid blocks. Another possibility is step by step substitute all of the monotone decreasing runs with their average, as long as remain any.

Let $I > 1$, $d > 1$ be integers and $a(1), a(2), \dots, a(I)$ be arbitrary vectors in \mathbf{R}^d the d -dimensional Euclidean space.

DEFINITION: *The multi-dimensional monotone regression problem* is to determine the permutation $\pi(1), \pi(2), \dots, \pi(I)$ of $1, 2, \dots, I$ and a monotone sequence of d -dimensional vectors $x(1) \leq x(2) \leq \dots \leq x(I)$ such that

$$\sum_{i=1}^I \|a(\pi(i)) - x(i)\|^2$$

is minimal.

DEFINITION: We say that $x < y$ for $x, y \in \mathbf{R}^d$ if $x_j \leq y_j$ for $1 \leq j \leq d$, where x_j resp. y_j stand for the j th coordinate of x resp. y .

For any permutation $\pi = (\pi(1), \pi(2), \dots, \pi(I))$ minimization of $C(\pi, x) = \sum_{i=1}^I \|a(\pi(i)) - x(i)\|^2$ requires to solve d independent scalar monotone regression, because one can determine the coordinates of the $x(i)$ -s independently. We call this step *permutation projection* and denote by $C(\pi)$ the minimum of $C(\pi, x)$. Accordingly the multi-dimensional monotone regression problem is a discrete optimizing problem, one possibility is to consider all permutations π and take the minimum.

There is a slightly different version of the problem which we state in a more general setting. Let μ be an arbitrary probability distribution in the d -dimensional space. Let $S(t) = (S_1(t), \dots, S_d(t))$ be a monotone mapping of the real line i.e. S_1, \dots, S_d are monotone increasing functions,

DEFINITION: The *S-variance of μ* is defined as

$$\Gamma(S) = \int_{\mathbf{R}^d} \inf_t \|x - S(t)\|^2 \mu(dx).$$

The monotone regression of μ is the function S minimizing $\Gamma(S)$ and the minimum is the *monotone variance of μ* . We say that the function giving the minimum is the *backbone of μ* .

Let the probability distribution μ be concentrated uniformly on $a(1), \dots, a(I)$ that is $\mu(a(i)) = \frac{1}{I}$, $1 \leq i \leq I$ and let S be an arbitrary monotone function. Suppose that S is continuous. Then $\inf_t \|a(i) - S(t)\|^2$ is achieved and let $t(i)$ be a possible choice. Let π be a permutation such that $t(\pi(i))$ is monotone non-decreasing. Set $x(i) = S(t(\pi(i)))$ then $x(1) \leq \dots \leq x(I)$ and

$$\Gamma(S) = \frac{1}{I} \sum_{i=1}^I \|a(i) - S(t(i))\|^2 = \frac{1}{I} \sum_{i=1}^I \|a(\pi(i)) - x(i)\|^2 = C(\pi, x).$$

That way any monotone curve in \mathbf{R}^d determines a permutation. We call this step *curve projection* since from a monotone curve we get a permutation.

Let π_1 be an arbitrary permutation and let $x(1) \leq \dots \leq x(I)$ be the monotone sequence minimizing $C(\pi_1, x)$, i.e. $C(\pi_1, x) = C(\pi_1)$. Let us define a function $S(t)$ as

$$S(t) = \begin{cases} x(1) & \text{for } t \leq 1 \\ (t-i)x(i+1) + (i+1-t)x(i) & \text{for } i < t \leq i+1, 1 \leq i < I \\ x(I) & \text{for } t > I, \end{cases} \quad (7)$$

If $\Gamma(S) < C(\pi_1)$ then $C(\pi)$ is not minimized in π_1 , because $C(\pi_2) \leq \Gamma(S)$, if π_2 is the curve projection of S .

Starting with a permutation π_1 the permutation projection provides some $x(1) \leq \dots \leq x(I)$. Next $x(1) \leq \dots \leq x(I)$ defines a monotone function S by formula (7) which by curve projection results in a new permutation π_2 , as long as $\Gamma(S) < C(\pi_1)$.

This is a monotone algorithm for minimizing $C(\pi)$. Evidently it terminates in finite number of steps. Experience shows that the algorithm needs only a reasonably small number of steps. As we found there are rather large number of fixed points of the iteration. Searching for a global optimum we used genetic optimization method. The possible missing values of the data may be substituted by an appropriately chosen number, say B^L or B^U accordingly.

The distribution of the residuals $\|a - x\|^2$ may shed light on the number of outliers; the independently expressed genes. Let us denote by κ the number of outliers. To each permutation π a new risk function can be defined by

$$C_\kappa(\pi) = \sum_{i=1}^{I-\kappa} \delta(i)^2, \quad (8)$$

where $\delta(1) \leq \dots \leq \delta(I)$ is the ordered sample of the residuals $\|a(\pi(i)) - x(i)\|$.

The new risk function can be minimized by a similar algorithm as the one described above. Starting with a monotone curve S we order the distances of the sample points from the curve and drop κ points with largest distances. The permutation projection step is performed on the remaining $I - \kappa$ points. It results in a new curve and the iteration continues. In course of the iteration the set of outliers may change.

A similar alternating minimization procedure may be used in general setting. Given an arbitrary probability measure μ and a monotone curve S , the curve projection is a measurable mapping T ordering to any point of \mathbf{R}^d the parameter t of the closest point of the curve. The S -variance of μ is defined by

$$\Gamma(S) = \int_{\mathbf{R}^d} \|x - S(T(x))\|^2 \mu(dx).$$

The generalization of the permutation projection means minimization of the above risk function in S . The problem reduces to the following d independent one-dimensional problems.

Let ν be a one dimensional probability distribution and let f be square integrable with respect to ν . Find the monotone non-decreasing function g minimizing $\int (f - g)^2 d\nu$. The solution is given by the largest convex minorant of $F(x) = \int_0^x f d\nu$.

So far we dealt with the non-parametric regression procedure, looking for a backbone curve spanned by a monotone set of points in \mathbf{R} . The procedure has a parametric version as well.

For the sake of a certain standardization we postulate the distribution of gene expression X of a randomly chosen gene standard Gaussian:

$$P(X \leq t) = \Phi(t) = \int_{-\infty}^t \varphi(u) du,$$

where $\varphi(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}$. Consequently the random variable $\Phi(X)$ is uniformly distributed in the interval $(0, 1)$. This is the so called quantile transformation. We use the quantile transformation to parameterize the “backbone” function $S(t)$. The function

$$\Psi(u) = \lambda u + (1 - \lambda)u^\alpha \quad (9)$$

transforms monotonously the interval $(0, 1)$ into itself for any $0 \leq \lambda \leq 1$, $\alpha > 0$. The parametric form of the k th coordinate of the “backbone” function is

$$S_j(t) = A^L + \Psi_j(\Phi(t))(A^U - A^L), \quad (10)$$

where the interval (A^L, A^U) is contained by the interval (B^L, B^U) of cut-points. (The parameters $\lambda, \alpha, A^L, A^U$ depend on j , but for the sake of simplicity it is not incorporated in the notations.)

3.3. MAXIMUM LIKELIHOOD ESTIMATE

Equation (1) gives the stochastic description of the data. At any given time for a given backbone function $S(t)$ we can simulate imaginary data fields. First we should decide that a randomly chosen gene independently or identically expressed by conditions. If it is identically expressed we generate a simple standard Gaussian variable X and the corresponding gene expression levels are $H(S_k(X) + Z_k)$, $k = 1, \dots, K$, where H is given by (2), the parameters B^L, B^U may depend on the investigated condition k and Z_1, \dots, Z_K are independent Gaussian measurement errors with 0 expectations and $\sigma_1, \dots, \sigma_K$ standard deviations. If the gene is independently expressed then we generate K independent standard Gaussian variables X_1, \dots, X_K and the corresponding

measured gene expression level is $H(S_k(X_k) + Z_k)$, $k = 1, \dots, K$. If we know how to simulate data for a given backbone function $S(t)$, the actual form of $S(t)$ and the parameters $B^L, B^U, \sigma_1, \dots, \sigma_K$ can be determined in course of some “tuning” procedure. We are seeking the main characteristics of the simulation which provides the best possible fit to the actual data. To evaluate the goodness of fit of a stochastic model we can use some set of statistics, simple functions of the data. This case tuning means to find the main characteristics providing the same statistics as the original data. This procedure resembles hitting target.

The logic of the main statistical estimation procedure is somewhat different. We evaluate the probability or likelihood of the data using actual main characteristics (parameters) of the stochastic model. The tuning procedure means to chose the parameters in order to maximize the likelihood function. Without truncation the *likelihood of an identically expressed gene* is

$$f^{(\text{id})}(y) = \int_{-\infty}^{+\infty} \varphi(x) \prod_{k=1}^K \frac{1}{\sigma_k} \varphi\left(\frac{y_k - S_k(x)}{\sigma_k}\right) dx, \quad (11)$$

and *likelihood of independently expressed gene* is

$$f^{(\text{in})}(y) = \prod_{k=1}^K \int_{-\infty}^{+\infty} \varphi(x) \frac{1}{\sigma_k} \varphi\left(\frac{y_k - S_k(x)}{\sigma_k}\right) dx. \quad (12)$$

Let us denote r the probability that a gene is independently expressed. Then the density function of this model is

$$f(y) = (1 - r) f^{(\text{id})}(y) + r f^{(\text{in})}(y). \quad (13)$$

In case of missing data, when observation is C^L resp. C^U the term $\varphi\left(\frac{y_k - S_k(x)}{\sigma_k}\right)$ has to be substituted by $\Phi\left(\frac{B^L - S_k(x)}{\sigma_k}\right)$ resp. $1 - \Phi\left(\frac{B^U - S_k(x)}{\sigma_k}\right)$. If C^L and C^U coincide then the corresponding term is the sum of the above probabilities. If we have n_ℓ repeated observations for the gene ℓ under each conditions, then we include them in the above likelihoods. Evaluating the measurements at each given time m , we have to maximize the loglikelihood function:

$$\mathcal{L} = \sum_{\ell=1}^L \log f(y_\ell). \quad (14)$$

In parametric case this function depends on parameters $\lambda_k, \alpha_k, A_k^L, A_k^U, \sigma_k, k = 1, \dots, K$ and the probability of independent expression level r . The likelihood function could be optimized by standard optimization procedures. On the actual data we worked on, the result of the above procedure is given in Table 1.

In case of a linear regression the maximum likelihood estimate of the parameters are identical to the least square estimators of them. In our case, the two methods are different even for identical genes (see Table 2.). An advantage of the likelihood method is that it incorporates appropriately the censored observations. It is possible to consider the likelihood of a censored observation instead of substitute missing values for some artificial number or to leave out the corresponding data completely from the analysis.

The likelihood method gives an estimate for the probability of independent expression level r (See Table 1.). Least square method gives only the distribution of residuals. The maximum likelihood method does not give direct decision that an individual gene is independently or identically expressed. For this aim the posterior probability of independence

$$\rho(y) = \frac{r f^{(\text{in})}(y)}{f(y)}$$

can be used with some cut-points like %0.95. The estimations of probabilities p and q_m -s defined by formulas (3), (4) and (5) are $p = 0.13, q_1 = 0.45, q_2 = 0.44, q_3 = 0.27, q_4 = 0.24$.

A further advantage of the likelihood method is that it determines not only the posterior probability of independent expression but posterior distribution of the original gene expression too. In certain sense we can see through the noise, reconstructing from observed data Y the original X -s. One way to do that is

calculating the most probable values, the other is the use of conditional expectation. We prefer to use the second method. We used the reconstructed data for constructing clusters. This reconstruction is the basis of the so-called EM algorithm (McLachlan, G.J. et.al. (1997)), which is a procedure for non-parametric estimation of the “backbone” function $S(t)$. Having at hand a candidate for $S(t)$, we calculate the appropriate conditional distribution of original gene expressions for each given measurement Y . Collecting the measured expressions on virtual expression with weights given by posterior probabilities we get a new estimator $\hat{S}(t)$ for the backbone curve. The procedure comes from the logic of the EM algorithm. In its general setting we take the conditional expectation of the loglikelihood function of the full information model (E-step) and then take the maximum of this expected loglikelihood value (M-step). In this case the full information model means the knowledge of the virtual gene expressions. The conditional expectation curve is no longer monotone non-decreasing, thus we have to incorporate into the procedure a weighted monotone regression estimation resulting $\tilde{S}(t)$ backbone curve where the procedure may be reiterated. Of course in the actual implementation of the algorithm we have to use some discretisation.

In Figures 1.a.–d. we present backbone or profile curves estimated by four different methods discussed so far. The measured gene expressions (small crosses) are colored blue for identically expressed genes and red for independently expressed genes. This coloring is based on the parametric maximum likelihood estimation. The non-parametric curve estimators, both the least square and the likelihood, have large intervals parallel to the coordinate axes. We started our data analysis with the non-parametric least square estimator (black curves on Figures 1.a.–d.) We turned to the likelihood method partly looking for smooth curves, to get rid of the steps, with no avail. Theoretical monotone regression of uniform distribution on a strip around the line of points with equal coordinates gives steps that are one third of the size of the width of the strip.

4. Results: Forming Clusters of Gene Expressions

Although it is possible to cluster the raw experimental data of gene expressions, we prefer to use the conditional expectations of virtual gene expressions given by the maximum likelihood estimator. One advantage of this method is that conditional expectations are close to each other for identical genes. For each gene the average of virtual expressions under investigated conditions were taken at each time. Then we took differences from this average and performed principal component analysis to this data. Figure 2.a. shows the genes in the plane spanned by the two main factors of the principal component analysis. The majority of the investigated 410 genes are situated in the middle. However 44 genes outside of the origins are forming four “arms”. We performed k -means clustering the data, with $k = 5$, and the colors of Figure 2.a. are showing these clusters in the plane of the two principal components. Figure 2.b. shows the time dynamics in the four clusters. We found that clusters of colors Red and Black contain mostly genes with different expression levels at time 1, while the Green and Blue clusters contain genes with independent expression levels mostly at time 2.

We call conditional expectations as Gaussian normalization of the measured gene expressions, because we postulated the distribution of gene expressions to be standard Gaussian. On the whole set of estimated gene expressions we formed 18 clusters which are shown by Figure 3. and the colors are showing the genes previously clustered by the k -means method. The figures showing the time dynamics of gene expressions at the observed four times. Curves with resp. without black dots showing time dynamics of gene expressions under condition 1 resp. condition 2.

The identity or independence of gene expressions was incorporated into the procedure of decomposition of mixtures of Gaussian distributions a similar way as it described in the previous section. As we remarked earlier decomposition of Gaussian mixtures is heavily sensitive for small eigenvalues of the covariance matrix and found that the cut-point 0.1 is the best choice for our data.

It is important that in forming 18 clusters we did not use the result of the previously mentioned clustering. However Figure 3. is colored according to the colors of the cluster formed by the k -means method.

References

- Alter,O., Brown,P.O. and Botstein,D. (2000). Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Nat. Acad. Sci. USA*, **97**, 10101–10106.
- Baldi,P., and Long,A.D. (2001). A Bayesian framework for the analysis of microarray expression data: regularized t -test and statistical inferences of gene changes. *Bioinformatics*, **17**, 509–519.
- Barbe,P. (1998). Statistical analysis of mixtures and the empirical probability measure. *Acta Applicandae Mathematicae*, **50**, 253–340.
- Brown,M.P.S., Grundy,N.W., Lin,D., Cristianini,N., Sugnet,C.W., Furey,T.S., Ares,Jr.M. and Haussler,D. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Nat. Acad. Sci. USA*, **97**, 262–267.
- Csiszár,I. and Tusnády,G. (1984). Information geometry and alternating minimization procedures. *Statistics and Decisions*, Supplement Issue **1**, 205–237.
- Groeneboom,P., Jongbloed,G. and Wellner,J. (2001). Estimation of convex function: characterization and asymptotic theory. (manuscript).
- Herrero,J., Valencia,A. and Dopazo,J. (2001). A hierarchical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics*, **17**, 126–136.
- Holter,N.S., Mitra,M., Maritan,A., Cieplak,M., Banavar,J.R. and Fedoroff,N.V. (2000). Fundamental patterns underlying gene expression profiles: Simplicity from complexity. *Proc. Nat. Acad. Sci. USA*, **97**, 8409–8414.
- Khan,J., Wei,S.J., Ringner,M., Saal,L.H., Ladanyi,M., Westerman,F., Berthold,F., Schwab,M., Antonescu,C.R., Peterson,C. and Meltzer,P.S. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, **7**, 673–679.
- Magder,L.S. and Zeger,S.L. (1996). A smooth nonparametric estimate of a mixing distribution using mixtures of Gaussians. *Journal of Amer. Statist. Assoc.*, **91**, 1141–1151.
- Manduchi,E., Grant,G.R., McKenzie,S.E., Overton,G.C., Surrey,S. and Stoeckert,C.J. (2000) Generation of patterns from gene expression data by assigning confidence to differentially expressed genes. *Bioinformatics*, **16**, 685–698.
- McLachlan,G.J. and Krishnan,T. (1997) *The EM algorithm and extensions*. John Wiley, New York.
- Tamayo,P., Slonim,D., Mesirov,J., Zhu,Q., Kitareewan,S., Dmitrovsky,E., Lander,E.S. and Golub,T.R. (1999) Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Nat. Acad. Sci. USA*, **96**, 2907–2912.
- Troyanskaya,O., Cantor,M., Sherlock,G., Brown,P., Hastie,T., Tibshirani,R., Botstein,D. and Altman,R.B. (2001) Missing value estimation methods for DNA microarrays. *Bioinformatics*, **17**, 520–525.
- Wright,F.T. (1981) The asymptotic behavior of monotone regression estimates. *Ann. Statist.*, **9**, 443–448.
- Zhang,C.-H. (1990). Fourier methods for estimating mixing densities and distributions, *Ann. of Statist.*, **18**, 806–831.
- Zhang,C.-H. (1995). On estimating mixing densities in discrete exponential family models, *Ann. of Statist.*, **23**, 929–945.

Lidia Rejtő
Statistics Program
University of Delaware
Newark, DE 19717
USA
rejto@udel.edu

Gábor Tusnády
Alfréd Rényi Mathematical Institute
Hungarian Academy of Sciences
Reáltanoda u. 13-15
1053 Budapest, Hungary
tusnady@renyi.hu

TABLE 1. Estimators of Parameters by Different Methods

| | | Time 1. | | | |
|-----------|--|--------------|--------------------|--------------|--------------------|
| | | Condition 1. | | Condition 2. | |
| | | LEAST SQUARE | MAXIMUM LIKELIHOOD | LEAST SQUARE | MAXIMUM LIKELIHOOD |
| α | | 9.7648 | 9.5180 | 10.1034 | 9.4512 |
| λ | | 0.1025 | 0.0715 | 0.0755 | 0.0588 |
| A^U | | 0.0000 | 9.2800 | 3.0634 | 9.0400 |
| A^L | | 302.5534 | 291.5582 | 283.6295 | 260.0800 |

| | | Time 2. | | | |
|-----------|--|--------------|--------------------|--------------|--------------------|
| | | Condition 1. | | Condition 2. | |
| | | LEAST SQUARE | MAXIMUM LIKELIHOOD | LEAST SQUARE | MAXIMUM LIKELIHOOD |
| α | | 9.9138 | 9.4404 | 11.1313 | 10.0831 |
| λ | | 0.1102 | 0.0619 | 0.1803 | 0.0969 |
| A^U | | 4.8243 | 6.8400 | 0.0000 | 6.3200 |
| A^L | | 228.9280 | 202.3663 | 266.8219 | 238.9873 |

| | | Time 3. | | | |
|-----------|--|--------------|--------------------|--------------|--------------------|
| | | Condition 1. | | Condition 2. | |
| | | LEAST SQUARE | MAXIMUM LIKELIHOOD | LEAST SQUARE | MAXIMUM LIKELIHOOD |
| α | | 13.0326 | 11.8306 | 9.3522 | 9.1290 |
| λ | | 0.0800 | 0.0410 | 0.0914 | 0.0514 |
| A^U | | 1.2648 | 11.4800 | 0.9523 | 5.3200 |
| A^L | | 886.3566 | 826.5200 | 488.5640 | 483.5200 |

| | | Time 4. | | | |
|-----------|--|--------------|--------------------|--------------|--------------------|
| | | Condition 1. | | Condition 2. | |
| | | LEAST SQUARE | MAXIMUM LIKELIHOOD | LEAST SQUARE | MAXIMUM LIKELIHOOD |
| α | | 9.3450 | 8.8874 | 11.9890 | 10.9055 |
| λ | | 0.1045 | 0.0423 | 0.1398 | 0.0677 |
| A^U | | 2.4985 | 6.7600 | 0.0000 | 6.5600 |
| A^L | | 528.9654 | 504.6000 | 577.3708 | 569.3600 |

TABLE 2. Maximum Likelihood Estimators of Parameters

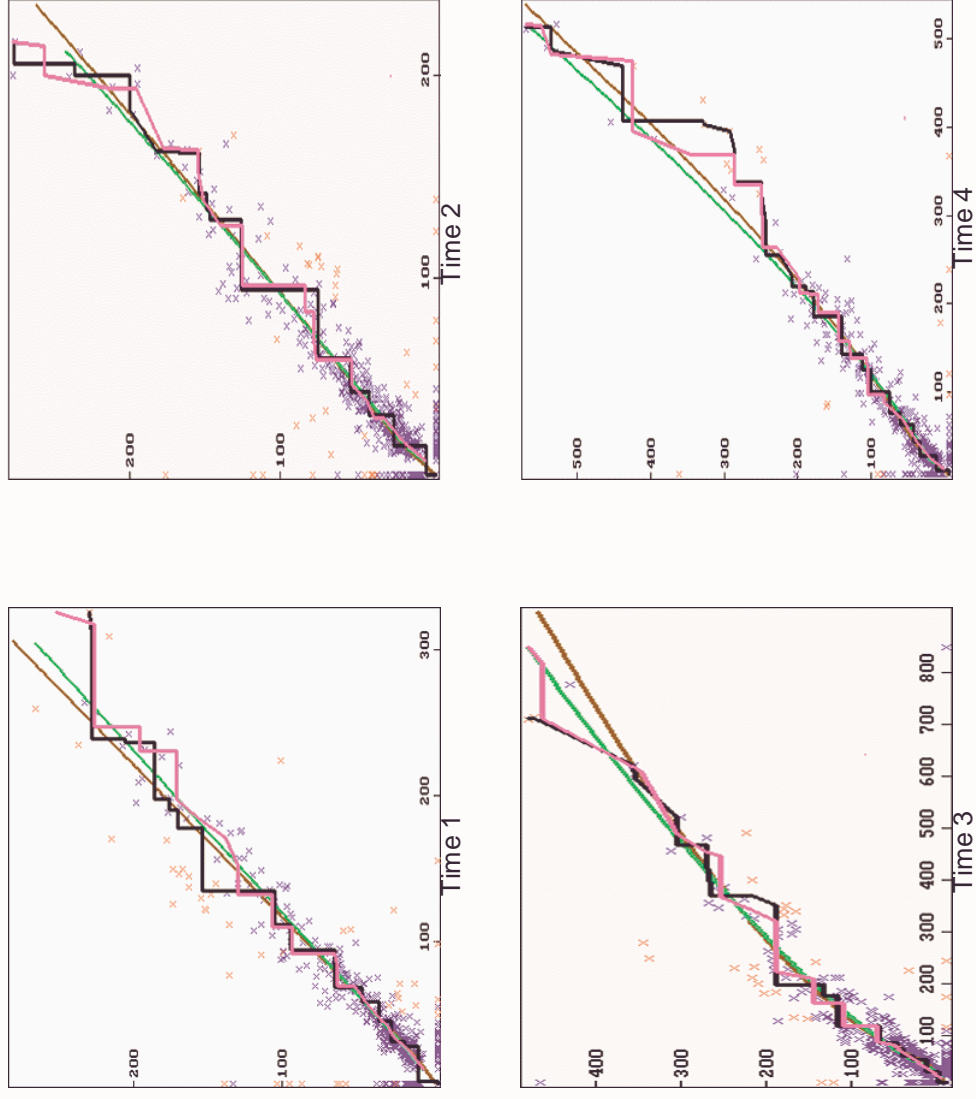
| | | Time 1. | | | | | |
|-----------|--|--------------|----------|----------|--------------|----------|----------|
| | | Condition 1. | | | Condition 2. | | |
| | | ESTIMATE | AVERAGE | ST.ERROR | ESTIMATE | AVERAGE | ST.ERROR |
| r | | 0.1717 | 0.1708 | 0.0249 | - | - | - |
| α | | 9.5180 | 10.9904 | 0.4212 | 9.4512 | 10.3656 | 0.5275 |
| σ | | 9.6589 | 9.9296 | 0.3102 | 6.1100 | 6.4510 | 0.2566 |
| λ | | 0.0715 | 0.0757 | 0.0041 | 0.0588 | 0.0550 | 0.0045 |
| A^U | | 9.2800 | 9.3703 | 0.0494 | 9.0400 | 9.3382 | 0.4052 |
| A^L | | 291.5582 | 291.7968 | 5.4295 | 260.0800 | 254.1093 | 3.9305 |

| | | Time 2. | | | | | |
|-----------|--|--------------|----------|----------|--------------|----------|----------|
| | | Condition 1. | | | Condition 2. | | |
| | | ESTIMATE | AVERAGE | ST.ERROR | ESTIMATE | AVERAGE | ST.ERROR |
| r | | 0.1651 | 0.1657 | 0.0131 | - | - | - |
| α | | 9.4404 | 11.9181 | 0.9751 | 10.0831 | 12.9796 | 0.8758 |
| σ | | 8.8160 | 8.4438 | 0.2281 | 8.0482 | 8.3315 | 0.1674 |
| λ | | 0.0619 | 0.0680 | 0.0146 | 0.0969 | 0.0935 | 0.0156 |
| A^U | | 6.8400 | 7.2067 | 0.5370 | 6.3200 | 7.3114 | 0.8358 |
| A^L | | 202.3663 | 198.6596 | 5.1067 | 238.9873 | 236.3293 | 8.2851 |

| | | Time 3. | | | | | |
|-----------|--|--------------|----------|----------|--------------|----------|----------|
| | | Condition 1. | | | Condition 2. | | |
| | | ESTIMATE | AVERAGE | ST.ERROR | ESTIMATE | AVERAGE | ST.ERROR |
| r | | 0.1112 | 0.0903 | 0.0031 | - | - | - |
| α | | 11.8306 | 13.4448 | 0.8954 | 9.1290 | 10.4524 | 0.6454 |
| σ | | 20.3065 | 20.9693 | 1.5705 | 10.3463 | 10.8319 | 0.1193 |
| λ | | 0.0410 | 0.0390 | 0.0068 | 0.0514 | 0.0557 | 0.0042 |
| A^U | | 11.4800 | 13.3093 | 2.7964 | 5.3200 | 5.4332 | 0.0970 |
| A^L | | 826.5200 | 773.4680 | 30.6279 | 483.5200 | 460.5552 | 15.9368 |

| | | Time 4. | | | | | |
|-----------|--|--------------|----------|----------|--------------|----------|----------|
| | | Condition 1. | | | Condition 2. | | |
| | | ESTIMATE | AVERAGE | ST.ERROR | ESTIMATE | AVERAGE | ST.ERROR |
| r | | 0.0867 | 0.0792 | 0.0193 | - | - | - |
| α | | 8.8874 | 9.9130 | 0.4373 | 10.9055 | 11.4834 | 0.6577 |
| σ | | 10.9122 | 11.3682 | 0.2556 | 14.3294 | 14.7056 | 0.3634 |
| λ | | 0.0423 | 0.0507 | 0.0044 | 0.0677 | 0.0643 | 0.0043 |
| A^U | | 6.7600 | 6.8486 | 0.0556 | 6.5600 | 8.3907 | 1.7315 |
| A^L | | 504.6000 | 492.5773 | 2.0364 | 569.3600 | 540.6404 | 14.1653 |

Figure 1
Comparison of Backbone Functions



black: non-parametric least square estimate
 brown: parametric least square estimate
 purple: non-parametric maximum likelihood estimate
 green: parametric maximum likelihood estimate

Figure 2

Clusters on the Plane of Main Factors of Normalized Data

Time Dynamics in Clusters of Normalized Data

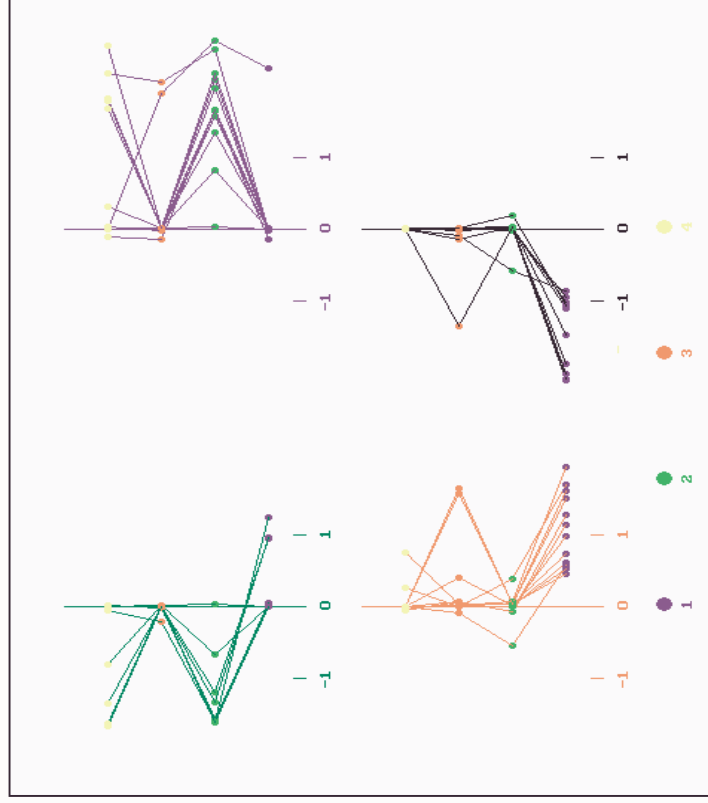
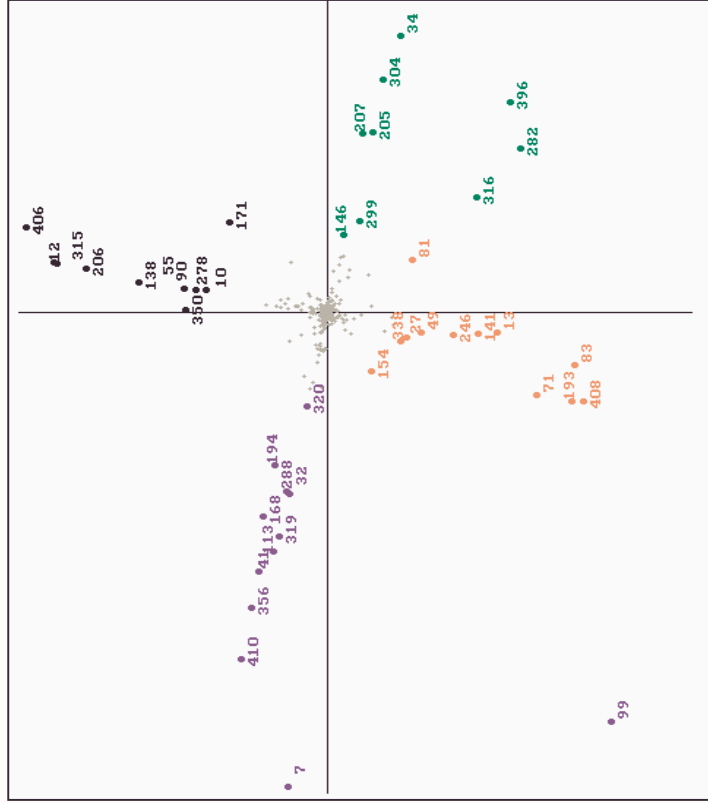


Figure 3.
Clusters of Gene Expressions

