Journal of Computational Intelligence in Bioinformatics
ISSN: 0973-385X Volume 4 Number 1 (2011) pp. 55-71
© Research India Publications
http://www.ripublication.com/jcib.htm

Understanding Principal Component Analysis to Dispel Microarray Data

Shiv Narayan Sharma[1*], Zenu Jha[1], Ravi R. Saxena[2] and D.K. Sharma[1]

[1]Department of Plant Molecular Biology & Biotechnology,
[2]Department of Agriculture Statistics,
IGKV, Raipur, India - 492006
**Corresponding Author E-mail: shivsharma.bt@ovi.com*

Abstract

The microarray experiments are producing huge amount of data sets in almost all fields of biological research.It is needs to analyze and properly exploit all the available information inherently presents in the data sets. The Principal Component Analysis (PCA) has stronghold on microarray data analysis - A black box that is widely used but poorly understood. The goal of this article is to dispel the magic behind this black box and to focus on building a solid intuition for how and why principal component analysis works in huge numbers of high throughput microarray data set.It is a versatile and easy-to-use multivariate mathematical statistical method developed to extract maximal information from large data matrices containing numerous columns and rows. It also makes possible the elucidation of the relationship between the columns and rows of any data matrix without being one of the dependent variable. So the PCA is a projection method representing the original data in smaller dimensions. We suppose that this article helpful for readers of all levels, researchers and students. They will be able to gain better understanding of the power of PCA as well as when, how and why of applying this technique in microarray data sets.

Keywords: Principal Component Analysis, Microarray, Data Mining, Gene expression.

Introduction

DNA microarray technology allows scientists to study the expression of thousands of genes - potentially entire genomes. It has been widely hailed as a powerful tool to study the global gene expression in organisms or tissues [1, 2]. Microarray can be applied to a wide range of studies including gene regulation, disease diagnosis and prognosis, cancer classification, bio-marker discovery and drug development. The microarray's capacity to compare gene expression patterns in different tissues or conditions threatens to change the way biology is practiced and understood. There are vast amount of gene expression data have generated across many conditions such as treatments or time points. But the problemis that the resultant mass of data is possibly irrelevant, insignificant or redundant. Therefore it is necessary to select methods for analyzing and representing these biological data, so it becomes possible to make a visual inspection of the relationship between conditions in such a multi-dimensional matrix. The PCA is the most important and broadly used statistical methods employed to dispel the complexity of biologic systems in analysis of microarray results. It is one of the tool for data analysis, visualization or compression and has a wide range of applications. It is finds linear combinations of the variables called principal components, corresponding to orthogonal directions maximizing variance in the data. Numerically, a full PCA involves a singular value decomposition of the data matrix. In other words, it is a mathematical algorithm that reduces the dimensionality of the data while retaining most of the variation in the data set [3]. It accomplishes this reduction by performing a covariance analysis between factors and identifying directions, called principal components, along which the variation in the data is maximal. By using a few components, each sample can be represented by relatively few numbers instead of by values for thousands of variables. Samples can then be plotted, making it possible to visually assess similarities and differences between samples and determine whether samples can be grouped.

Let's take an example that illustrates how PCA works with a microarray experiment: suppose you measure 8000 differentially expressed genes in 4 different rice genotypes. These values could form a matrix of 4 x 8000 measurements. Now imagine that each of these 8000 genes is plotted in a multi-dimensional on a scatter plot consisting of 4 axes, 1 for each genotype. The result is a cloud of values in multi-dimensional space.

Functional concepts of Principal Component analysis

To obtain more precise definition of goal, we need a more precise definition of data. A simple example to understand about the principal component analysis showed in figure -1.

A man moving along the x-axis, at that time three cameras measure the man's position (system of interest) in a three-dimensional space (since we live in a three dimensional world). Each movie camera records an image indicating a two dimensional position of the man. Unfortunately, because of our ignorance, we do not even know what are the real "x", "y" and "z" axes. Suppose the number of measurement in this case, 10000 data sets has been generated in 6-dimensional vectors, where each camera contributes a 2-dimensional projection of the man position. In general, each data sample is a vector in 'n' dimensional space, where 'n' is the number of measurement types. All measurement vectors in this space are a linear combination of this set of unit length basis vectors. A native and simple choice of a basis A is the identity matrix I.

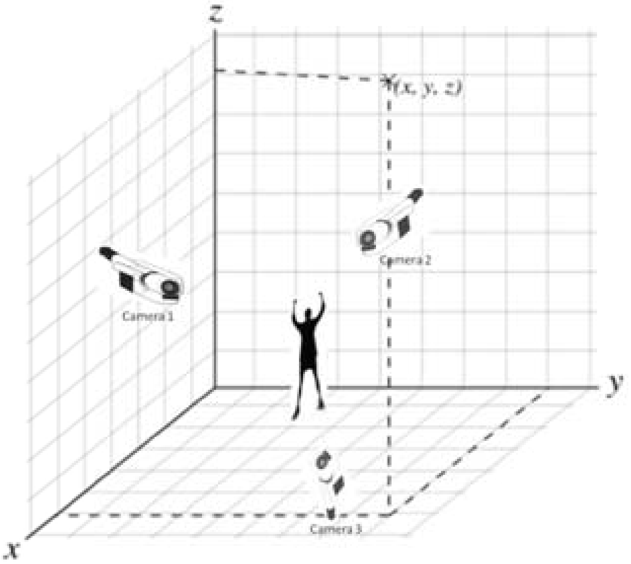


Figure 1: A model for understanding principal component analysis.

$$A = \begin{bmatrix} A_1 \\ A_2 \\ \cdot \\ \cdot \\ \cdot \\ A_n \end{bmatrix} = \begin{bmatrix} 1\,0\,.\,.\,.\,1 \\ 0\,1\,.\,.\,.\,0 \\ .\quad .\qquad . \\ .\quad .\quad .\quad . \\ .\quad .\qquad . \\ 0\,0\,.\,.\,.\,1 \end{bmatrix} = I$$

Where each row is a basis vector A$_i$ with n components (n =1, 2,........,n)

At one point in time, camera A records a corresponding position (xa (t), ya (t)). Each trial can be expressed as a six dimensional column vector

$$\overset{\rightarrow}{\lambda} = \begin{bmatrix} xa \\ ya \\ xb \\ yb \\ xc \\ yc \end{bmatrix}$$

Where each camera contributes two points and the entire vector is the set of coefficients in the naive basis A. With this rigor one may now state more precisely what PCA does: PCA makes one stringent but powerful assumption: linearity. Linearity vastly simplifies the problem by restricting the set of potential bases, and formalizing the implicit assumption of continuity in a data set. A subtle point it is, but we have already assumed linearity by implicitly stating that the data set even characterizes the dynamics of the system. In other words, we are already relying on the superposition principal of linearity to believe that the data characterizes or provides an ability to interpolate between the individual data points. The above diagram has more than one dimensions and the aim of the statistical analysis of these data sets is usually to see that relationship between the dimensions.

Basic statistics measures behind PCA

The entire subject of statistics is based around the idea that you have this big set of data, and you want to analysis that set in terms of the relationships between the individual points in that data set. The basic statistical measures applied in the background of principal component analysis are important to apply PCA in microarray analysis.

Standerd deviation: It is the square root of mean of the squared deviation of individual values from their mean. It indicates a sort of group standerd spread of values from the mean.

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}}$$

Variance: Variance is another measure of the spread of data in a data set. In fact it is almost identical to the standard deviation. Variance of a random variable X are.

$$\sigma^2 = Var(\boldsymbol{X}) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}$$

Sample variance from data$x_1, \ldots, x_n$:

$$s^2 = Var(X) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i{}^2\right) - \frac{1}{n-1}\bar{x}^2$$

If considered the data of height and yield of rice plant. So the statistical analysis based on the height of plant has any effect on their yield. In this case only standard deviation and variance was operated because the data have limited dimension. So that you could only calculate the standard deviation for each dimension of the data set independently of the other dimensions. However, it is useful to have a similar measure to find out how much the dimensions vary from the mean with respect to each other.

Covariance: The standerd devation and veriance purely on dimensione data mesures. However many data sets have more than one dimension, and the aim of the statistical analysis of these data sets is usually to see if there is any relationship between the dimensions.Covariance is always worked on two dimensions.If you calculate the covariance between one dimension and itself, you get thevariance. So, if you had a 3-dimensional data set (x, y , z), then you could measure the covariance between the x and y dimensions, the x and z dimensions, and the y and z dimensions. Measuring the covariance between x and x , or y and y, or z and z would give the variance of the x , y and z dimensions respectively. Covariance between random variablesxX, Y:

$$\sigma_{xy} = Cov(x, y) = E\{(X - \bar{X}_x)(Y - \bar{Y}_y)\}$$

Sample covariance from data(x_1, y_1), . . . , (x_n, y_n):

$$s_{xy} = Cov(x, y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2(y_i - \bar{y})^2 = \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i y_i\right) - \frac{1}{n-1}\bar{x}\,\bar{y}$$

Coveriance matrix: The covariance matrix generalizes the concept of variance to random vectors, or sets of random variables. Suppose the x and y random vector than

$$M_x = [x_1 - \bar{x}x_2 - \bar{x} \ldots\ldots\ x_n - \bar{x}]$$
$$M_y = [y_1 - \bar{y}y_2 - \bar{y} \ldots\ldots\ y_n - \bar{y}]$$
$$s_{xy} = \frac{M_x.M_y}{n-1} = \frac{M_x M_{y'}}{n-1}$$

The experiment were design with many simultaneous random variables, put them into vectors. The variance-covariance matrix (or simply the covariance matrix) of a random vector $\underset{X}{\rightarrow}$ and $\underset{Y}{\rightarrow}$ is given

If the data have a 3-dimensional (x, y, z), then you can measure the covariance between the x and y dimensions, the x and z dimensions, and the y and z dimensions. Measuring the covariance between x and x and y and y and z and z would give you the variance of the x, y and z dimensions respectively. If the data sets have more than two dimensions than more than one covariance can be calculated. For three dimensional data set (dimensions x, y, z) you could calculate *cov*(x, y), *cov*(x, z) and *cov*(y, z). In fact, for *n* dimensional data calculate through $\frac{n!}{(n-2)!*2!}$ different covariance values. A useful way to get all the possible covariance values between all the different dimensions is to calculate them all and put them in a matrix.

$$C^{n\times n} = cov(Dim_i, Dim_j, Dim_k)$$

Where $C^{n\times n}$ is a matrix with *n* rows and *n* columns and $Dim_x$is the *x* dimension. If *n* dimensions of data sets, then the matrix has *n* rows and columns (so is square) and each entry in the matrix is the result of calculating the covariance between two separate dimensions. Eg. the entry on row 2, column 3, is the covariance value calculated between the 2nd dimension and the 3rd dimension. Then, the covariance matrix has 3 rows and 3 columns, and the values are this

$$c = \begin{pmatrix} cov(x,x)\ cov\ (x,y)\ cov(x,z) \\ cov(y,x)cov\ (y,y)\ cov(y,z) \\ cov(z,x)\ cov\ (z,y)\ cov(z,z) \end{pmatrix}$$

Understanding the problems with many simultaneous random variables put them into vectors.

$$\vec{X}= \begin{bmatrix} R \\ S \end{bmatrix} \vec{Y}= \begin{bmatrix} T \\ U \\ V \end{bmatrix}$$

and then form a covariance matrix: $cov(\vec{X}\vec{Y})= \begin{pmatrix} cov(R,T)\ cov\ (R,U)\ cov(R,V) \\ cov(S,T)cov\ (S,U)\ cov(S,V) \end{pmatrix}$

In matrix/vector notation,$cov(\vec{X}\vec{Y})= E\left[\{\vec{X} - E(\vec{X})\}\{\vec{Y} - E(\vec{Y})\}\right]$

If there's one vector with all the variables:

$$\vec{X} = \begin{bmatrix} R \\ S \\ T \end{bmatrix}$$

$$Cov\ \bar{X} = Cov\ \bar{X},\bar{X} = E\ \left[(\vec{X} - E\overrightarrow{(X)})\ (\vec{X} - E\overrightarrow{(X)})'\right]$$

$$= \begin{pmatrix} cov(R,R)\ cov\ (R,S)\ cov(R,T) \\ cov(S,R)cov\ (S,S)\ cov(S,T) \\ cov(T,R)cov\ (T,S)\ cov(T,T) \end{pmatrix}$$

$$= \begin{pmatrix} var(R)\ cov\ (R,S)\ cov(R,T) \\ cov(S,R)\ var(S)\ cov(S,T) \\ cov(T,R)cov\ (T,S)\ var(T) \end{pmatrix}$$

The matrix is symmetric and the diagonal entries are ordinary variances.
Covariance matrix properties

$$cov(\vec{X}\vec{Y})= cov(\vec{X}\vec{Y})\ '$$

$$cov(A\ \vec{X} + \vec{B},\vec{Y}) = A\ cov(\vec{X}\vec{Y})$$

$$cov(\overrightarrow{X,}\ C\vec{Y} + \vec{D}) = cov(\vec{X}\vec{Y})C'$$

$$cov(A\vec{X} + \vec{B})= Acov(\vec{X}\ )A'$$

$$cov(\vec{X}_1+ \vec{X}_2,\vec{Y}) = cov(\vec{X}_1,\vec{Y})\ +cov(\vec{X}_2,\vec{Y})$$

$$cov(\vec{X}\vec{Y}_1 +\ \vec{Y}_2) = cov(\vec{X}_1\vec{Y}_1) + cov(\vec{X}_2\vec{Y}_2)$$

Eigenvalues and Eigenvectors: Theyare the properties of a matrix. It gives important information about the matrix, and can be used in matrix factorization. In general, a matrix acts on a vector by changing both its magnitude and its direction. Let A be a square matrix of size n.

Let A, be an *n* ×*n* matrix. The number λ is an eigenvalue of A. If there exists a non-zero vector v such that

λA=λv

In this case, vector **v** is called an eigenvector of *A* corresponding to λ.

λA=λv as (A−λ*I*)v = 0

Where I is the n x n identity matrix. Now, in order for a non-zero vector v to satisfy this equation, $A-\lambda I$ must not be invertible. Scalar λ is called an eigenvalue of A, vector $\underset{v}{\rightarrow} \neq 0$ is called an eigenvector of A associated with eigenvalue and the null space of $A-\lambda I$ is called the eigenspace of A associated with eigenvalue. The eigenvectors can only be found for squarematrices but not every square matrix has eigenvectors. In the given n x n matrix that does have eigenvectors, there are n of them. The description of calculation of eigenvalue and eigenvector beyond the scope of this article. The reader may consult some specific topic.

Principal component analysis for Evaluation of microarray data

The study of gene expression has been greatly facilitated by the application of DNA microarray technology [4]. DNA microarrays consist of single-stranded DNA fragments affixed to a solid support [5, 6, 7]. It measures the expression of thousands of genes simultaneously. The each spot on the microarray consists of a population of identical DNA fragments that represent one particular gene. To measure expression, the total RNA of a cell is harvested and labelled with fluorescent nucleotide tags during reverse transcription to make fluorescent probes. Commonly, two cell populations are used—cells under control and experimental conditions. The probes are then placed on the chip and permitted to hybridize with the target fragments on the corresponding spot. The intensity of the spot is approximately proportional to the probe and hence mRNA concentration. In a typical experiment, two colors (red and green) are used to measure expression of the experimental population relative to the control. Equal total mRNA probe concentrations are used to query the microarray and intensity ratios between the colors are calculated and reported as data [4] (Fig. - 2).

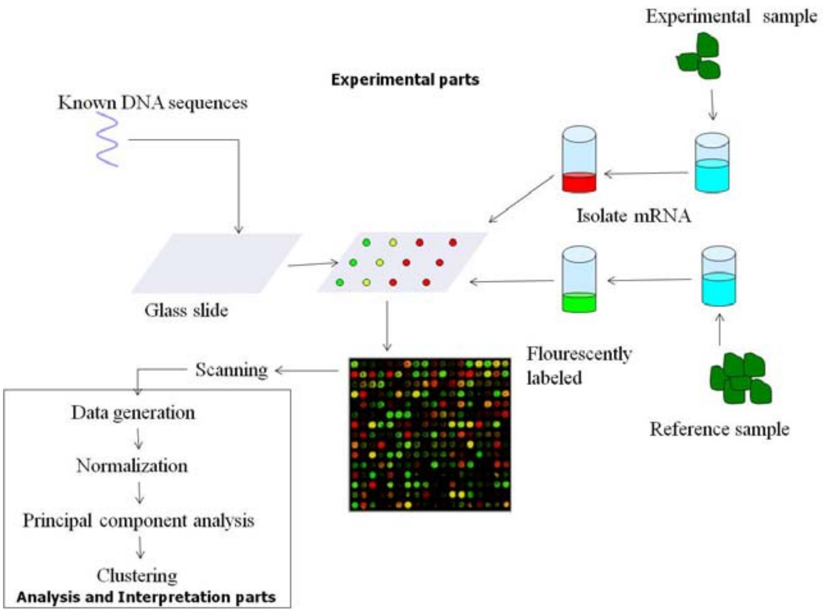


Figure 2: overview of the microarray experiment, data generation and analysis.

The anticipated flood of biological information produced by the microarray experiments will open new doors into genetic analysis [8]. Microarray data has been used to identify gene clusters based on co-expression [9, 10], define metrics that measure a gene's involvement in a particular cellular event or process [11], predict regulatory elements [12], and reverse engineer transcription networks [13, 14]. It is high-throughput genomic tools have sparked a remarkable increase in data production, leading to new evolutionary insights but the resultant masses of data are possibly irrelevant, insignificant or redundant. Therefore, we should carefully select methods for analyzing and representing microarray data.

Microarray data have high levels of noise and scattered distribution, so that it is desirable to carry out the analysis of microarray data within a statistical framework. Pre-processing and normalisation is an important aspect of the analysis is the strategy required to reduce the impact of noise and correct for systematic variation introduced by experimental procedures. The common normalisation strategies on the distribution of measured expression levels are these. Normalization is a critical step for obtaining data that are reliable and usable for subsequent analysis such as identification of differentially expressed genes and clustering.

Geometric mean (GM) normalisation

Geometric mean normalisation involves a simple linear transformation of data fromeach chip experiment such that the logged expression data lies on a standard scalein which it has mean zero and unit variance. The mean and standard deviation arecalculated by ignoring data which is ± 3 standard deviations from the mean of theun-normalised data, in order to avoid bias introduced by outliers. Correcting forthe mean of the logged data corresponds to correcting by the geometric mean inthe un-logged data, hence the name. We also correct for the variance of the loggeddata, since it appears that there are significant differences in this quantity betweenexperiments.

Least squares (LS) normalisation

One common normalisation approach is to do a linear regression of expressiondata from one chip to expression data on a reference chip followed by a lineartransformation to ensure all experiments have the same slope and intersect. This is unprincipled since standard linear regressionassumes that only one experiment is noisy (the measurement) while theother experiment can be considered a reference variable. In practice all experimentsconsidered will have comparable noise levels. The use of standard linearregression therefore leads to an asymmetrical method in which the result of normalisationis not equivalent for different choices of reference chip. A nice exampleof this asymmetry effect in regression is given by Hastie and Stuetzle [15] whoconsider a simple one dimensional example where interchanging the measurementand reference variables results in a significantly different regression line.

A major problem in microarray analysis is the large number of dimensions. In gene expression experiments each gene and each experiment may represent one dimension. For example, a set of 10 experiments involving 20,000 genes may be conceptualized as 20,000 data points (genes) in a space with 10 dimensions (experiments) or 10 points (experiments) in a space with 20,000 dimensions (genes).

Both situations are beyond the capabilities of current visualization tools and beyond the visualization capabilities of our brains. A natural solution would be to try to reduce the number of dimensions by eliminating those dimensions that are not "important". The PCA does exactly that by ignoring the dimensions in which data do not vary much. PCA calculates a new system of coordinates. The directions of the coordinate system calculated by PCA are the eigenvectors of the covariance matrix of the patterns. An eigenvector of a matrix A is defined as a vector z such as:

$$A_z = \lambda_z$$

where λ is a scalar called eigenvalue. For instance, the matrix:

$$A = \begin{bmatrix} -1 & 1 \\ 0 & -2 \end{bmatrix}$$

The eigenvalues λ_1= -1 and λ_2= - 2 and the eigenvectors $z_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $z_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

In intuitive terms, the covariance matrix captures the shape of the set of data points. PCA captures, by the eigenvectors, the main axes of the shape formed by the data diagram in an n-dimensional space. The eigenvalues describe how the data are distributed along the eigenvectors and those with the largest absolute values will indicate that the data have the largest variance along the corresponding eigenvectors. For instance, the figure below shows a data set with data points in a 2-dimensional space. However, most of the variability in the data lies along a one-dimensional space that is described by the first principal component (P_1). In this example the second principle component (P_2) can be discarded because the first principle component captures most of the variance present in the data (fig. - 3).

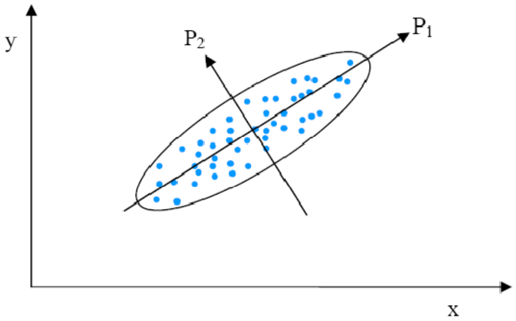


Figure 3: Each data point in this diagram has two coordinates. However, this data set is essentially one dimensional because most of the variance is along the first eigenvector p_1. The variance along the second eigenvector $p_2$is marginal, thus, p_2 may be discarded.

It is important to notice that in some circumstances, the direction of the highest variance may not be the most useful. For example, in gene expression diagram which describes gene expression levels from two samples, the PCA would capture two axes. One axis would represent the within-experiment variation, while the other would represent the inter-experiment variation. Although the within-experiment axis could show much more variance than the inter-experiment axis, the within-experiment axis is of no use for us. This is because we know a priorithat genes will be expressed at all levels [16]. The dimensionality reduction is achieved through PCA by selecting a small number of directions and look at the projection of the data in the coordinate system formed with only those directions. In spite of its usefulness, PCA has also limitations. Those limitations are mainly related to the fact that PCA only takes into consideration the variance of the data which is a first order statistical characteristic of the data. Another major limitation is that PCA takes into account only the variance of the data and completely discards the class of each data point.

Hypotehetical example of typical microarrya analysis

To understand the principal component analysis in a microarray hybridization experiement a simple experimentgiven with ‘g’ gene on ‘n’ number of hybridization than the data matrix $g \times n$ gene by array. The intensity indicate that the upregulation and down regulation of the gene.The extent of florecent intensity with negative sine showed the extent of down-regulation wheras the positive sign shows the extent of up-regulation of the gene. The valus of intensity used for constration of matrix for PCA analysis.

Genes		Arrays				
	Array1	Array2	Array3	Array4	Array5	……
	0.46	0.30	0.80	1.51	0.90	……
	-0.10	0.49	0.24	0.06	0.46	……
	0.15	0.74	0.04	0.10	0.20	……
	-0.45	-1.03	-0.79	-0.56	-0.32	……
	-0.06	1.06	1.35	1.09	-1.09	…..
	…..	……	……	……	……	……

Figure 8: the measurement of florescent intensity of microarray data for G gene and n hybridization.

Here we consider only 5 array and 25 spots in above example for the calculating PCA. M is 5 x 25. Where M (Original value) M` transform value of original value ($Y=X_i - \bar{X}$)

X	Array1	Array2	Array3	Array4	Array5	Average	SD	Variance
Gene A	0.46	0.3	0.8	1.51	0.9	0.794	0.468914	0.21988
Gene B	-0.1	0.49	0.24	0.06	0.46	0.23	0.254165	0.0646
Gene C	0.15	0.74	0.04	0.1	0.2	0.246	0.282454	0.07978
Gene D	-0.45	-1.03	-0.79	-0.56	-0.32	-0.63	0.2824	0.07975

Gene E	-0.06	1.06	1.35	1.09	-1.09	0.47	1.027302	1.05535
$Y=X_i-\bar{X}$	-0.334	-0.494	0.006	0.716	0.106			
	-0.33	0.26	0.01	-0.17	0.23			
	-0.096	0.494	-0.206	-0.146	-0.046			
	0.18	-0.4	-0.16	0.07	0.31			
	-0.53	0.59	0.88	0.62	-1.56			

Where $c = \frac{MM`}{25-1}$ is 25 x 25!

Covariances of the gene and array matrix are

	Gene A	*Gene B*	*Gene C*	*Gene D*	*Gene E*
Gene A	0.175904				
Gene B	-0.0231	0.05168			
Gene C	-0.06452	0.03446	0.063824		
Gene D	0.0439	-0.02112	-0.04128	0.0638	
Gene E	0.03388	-0.02542	0.02846	-0.18248	0.84428

The Singular Value Decomposition of M is M = USV^T, where

U is orthonormal, p × p.

V is orthonormal, q × q.

S is a diagonal p × q matrix, s1 ≥ s2 ≥ · · · ≥ 0.

$$M = \begin{bmatrix} & & \mathrm{U} & & \\ 0.01 & -0.85 & -0.51 & -0.09 & 0.07 \\ 0.00 & 0.15 & -0.23 & -0.63 & -0.73 \\ 0.00 & 0.40 & -0.51 & -0.44 & 0.62 \\ -0.01 & -0.03 & 0.65 & -0.64 & 0.38 \\ 1.00 & 0.00 & 0.01 & -0.01 & 0.00 \end{bmatrix} =$$

$$\begin{bmatrix} & & \mathrm{S} & & \\ 0.87 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.20 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.09 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.05 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.04 \end{bmatrix} = \begin{bmatrix} & & \mathrm{V}^{\mathrm{T}} & & \\ 0.04 & -0.03 & 0.03 & -0.21 & 0.98 \\ -0.97 & 0.14 & 0.19 & 0.10 & 0.02 \\ -0.25 & -0.50 & -0.68 & 0.45 & 0.12 \\ -0.03 & -0.65 & -0.04 & -0.74 & -0.18 \\ 0.01 & 0.56 & 0.70 & 0.44 & 0.05 \end{bmatrix}$$

Mis now decomposed into three matrices with SVD (singular value decomposition); i.e., A = USV^T. These terms are defined as follows. V^{T}is the transpose of V and S is a diagonal matrix that stores singular values (i.e., λ_i… λ_i + 1… λ_k). U and V are orthogonal matrices.

Their column vectors are the so-called *left* and *right* eigenvectors of A.

When these eigenvectors multiply Y, coordinates are shifted and rotated until they end up aligned with vectors, termed now basis vectors. This is an affine transformation since it involves translation. Note that PCA now re-expresses the data as a linear combination of its basis vectors, YV. Vcolumns (V^Trows) are found to produce the desired linear combinations. The first column of V corresponds to the largest PC, the second column corresponds to the second largest PC, and so on. These define the direction in which the variability of the original data set is maximized. Compute V from $V^T$and YV and plot the first two columns of YV.

	Gene A	Gene B	Gene C	Gene D	Gene E
V	0.04	-0.03	0.03	-0.21	0.98
	-0.97	0.14	0.19	0.1	0.02
	-0.25	-0.5	0.68	0.45	0.12
	-0.03	-65	-0.04	-0.74	-0.18
	0.01	0.56	0.7	0.44	0.05
YV	PC1	PC2	PC3	PC4	PC5
	0.44	-46.54	-0.05	-0.46	-0.46
	-0.26	11.22	0.21	0.33	-0.27
	-0.43	9.64	-0.08	0.06	-0.08
	0.44	-4.36	0.03	-0.07	0.15
	-0.85	-41.52	-0.42	-0.58	-0.59

Plotting of above matrix in two dimensional plot (Figure - 4)

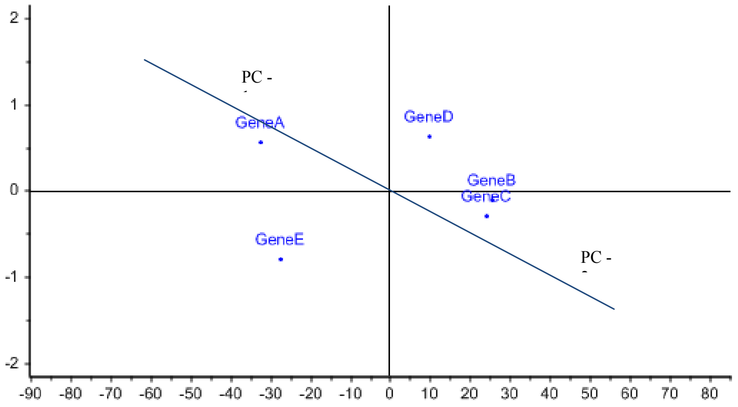
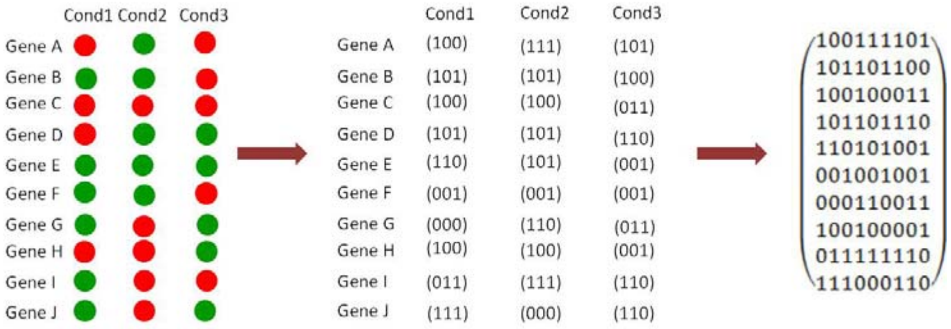


Figure 4

Another Approach for Analysis of Microarray Data

The given a matrix of expression data (fig. -5), where each row corresponds to a different gene and each column corresponds to one of several different conditions to which the cells were exposed. The a_{it}entry of the matrix contains the i^{th} gene's relative expression ratio with respect to a control population under condition t. To equalize the influence of induction and repression for microarray analysis natural log transform to all ratios need to be applying [17]. Up-regulated genes have a positive log expression ratio, while down-regulated genes have a negative log expression ratio.



	Cond1	Cond2	Cond3
Gene A	🟥	🟩	🟥
Gene B	🟩	🟩	🟥
Gene C	🟥	🟥	🟥
Gene D	🟥	🟩	🟩
Gene E	🟩	🟩	🟩
Gene F	🟩	🟩	🟥
Gene G	🟩	🟥	🟩
Gene H	🟥	🟥	🟩
Gene I	🟩	🟥	🟥
Gene J	🟩	🟥	🟩

	Cond1	Cond2	Cond3
Gene A	(100)	(111)	(101)
Gene B	(101)	(101)	(100)
Gene C	(100)	(100)	(011)
Gene D	(101)	(101)	(110)
Gene E	(110)	(101)	(001)
Gene F	(001)	(001)	(001)
Gene G	(000)	(110)	(011)
Gene H	(100)	(100)	(001)
Gene I	(011)	(111)	(110)
Gene J	(111)	(000)	(110)

$$\begin{pmatrix} 100111101 \\ 101101100 \\ 100100011 \\ 101101110 \\ 110101001 \\ 001001001 \\ 000110011 \\ 100100001 \\ 011111110 \\ 111000110 \end{pmatrix}$$

Figure5: microarray profile of gene and condition metrix.

To compute the principal components, the *n* eigenvalues and their corresponding eigenvectors are calculated from the *n*x*n* covariance matrix of conditions. Each eigenvector defines a principal component. A component can be viewed as a weighted sum of the conditions, where the coefficients of the eigenvectors are the weights. Each of the *n* components can be calculated for a given gene:

$$a'_{ij}=\sum_{t=1}^{n} a_{it}\, v_{tj}$$

Where $v_{tj}$is the t^{th} coefficient for the j^{th} principal component; a_{it} is the expression measurement for gene *i* under the t^{th} condition. *A'* is the data in terms of principal components. Since *V* is an orthonormal matrix, *A'* is a rotation of the data from theoriginal space of observations to a new space with principal component axes.The variance accounted for by each of the components is its associatedeigenvalue; it is the variance of a component over all genes. Consequently, theeigenvectors with large eigenvalues are the ones that contain most of theinformation; eigenvectors with small eigenvalues are uninformative.Determining *r*, the true dimensionality of the data, and eliminating noisycomponents is often ad hocand many heuristics exist. Eliminating low variancecomponents, while reducing noise, also discards some valuable information.

Theabove approach for analysis of microarray data by using principal component analysis given by Sager [18] in which the nucleotide base used for the coding the sequence of gene. He defined alignment matrix *F*, each row of which is a sequence vector $F^k$for the k^{th} gene sequence. Each base is encoded to a 4-bit binary number (A, C, G and T are respectively encoded to 1000, 0100, 0010 and 0001). A sequence vector consists of 1s and 0s, and corresponds to a point in 4*l* -dimensional space, where *l* is the length of the sequence. The number $C^{kk'}$of matchedmasses between genes *k* and *k'* can be expressed as the inner product of the gene vector. A comparisonmatrix *C*, each element of which is the number of matches for all pairs of genes can thus be expressedas the matrix product between alignment *F* and its transpose F^T. The principal axes $u_p$are definedas $Cu_p= \lambda_p u_p$where u_p is an eigenvector and λ_p is the corresponding eigenvalue of comparisonmatrix *C*. Each genes is plotted on the two-dimensional plane called gene space. The coordinate$x^k_p$of gene *k* in dimension p is given by $x^k_p= \lambda pu^k_p$Genes are classified into one or more groups,according to the distance between the two-dimensional gene plots. The coordinates $y_p$of condition inthe sequence are given by $y_p= F^T u_p$. The i^{th} element of $y_p$corresponds to a condition at position$i$ in the gene, and characteristic conditions of each group are detected by comparing the conditionswith the group of genes (fig. - 6).

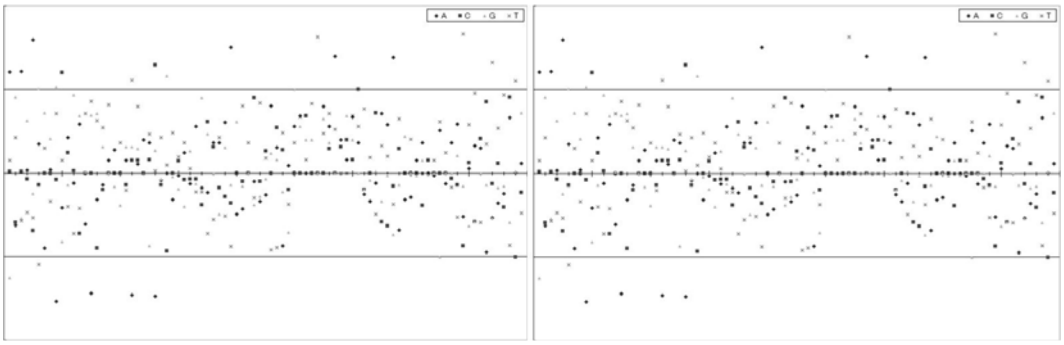


Figure 6: Characteristic bases on 2nd principal axis (left) and 3rd principal axis (right).

Principal Component Analysis based Clustering

Principal components analysis is often used as a pre-processing step to clustering [19]. It is a widely used technique for summarising expression levels obtained from microarray data and as an exploratory technique for finding functional analogues.It is sometimes applied to reduce the dimensionality of the data set prior to clustering.The hope for using PCA prior to cluster analysis is that PC's may "extract" the cluster structure in the data set. Since PC's are uncorrelated and ordered, the first few PC's, which contain most of the variations in the data, are usually used in cluster analysis, for example [20]. There are some common rules of thumb to choose how many of the first PC's to retain, but most of these rules are informal and unplanned [21]. On the other hand, there is a theoretical result showing that the first few PC's may not contain cluster information: assuming that the data is a mixture of two multivariate normal distributions with different means but with an identical within-cluster covariance matrix [22] showed that the first few PC's may contain less cluster structure information than other PC's. For example a subset of the sporulation data (477 genes) were classified into 7 temporal patterns [23]. Figure 7(a) is a visualization of this data in the space of the first 2 PC's, which contains 85.9% of the variation in the data. Each of the seven patterns is represented by a different color or different shape. The seven patterns overlap around the origin in figure 7(a). However, if we view the same subset of data points in the space of the first 3 PC's (containing 93.2% of the variation in the data) in figure 7(b), the seven patterns are much more separated. This example shows that a small variation (7.4%) in the data helps to distinguish the patterns, and different numbers and different sets of PC's have varying degree of effectiveness in capturing cluster structure. Therefore, there is a great need to investigate the effectiveness of PCA as a preprocessing step to cluster analysis on gene expression data before one can identify clusters in the space of the PC's.

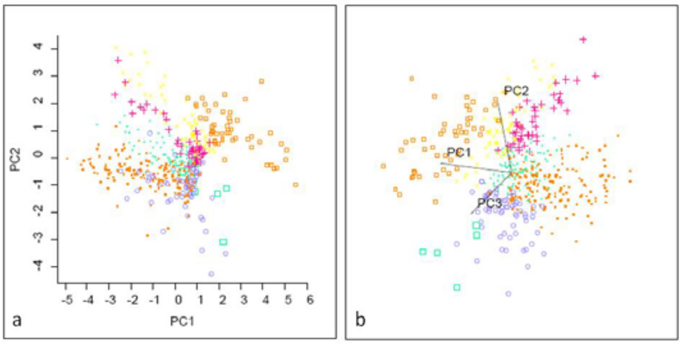


Figure 7: Visualization of a subset of the sporulation data.

K-means and Hierarchical clusteringis a commonly used data clustering for performing unsupervised learning tasks. New lower bounds for *K*-means objective function are derived by substracting total variance with the eigenvalues of the data covariance matrix. These results indicate that unsupervised dimension reduction is closely related to unsupervised learning.

The use of PCA before clustering can be justified by the fact that the larger principal components are expected to capture the structure in the data set.However, standard PCA does not always improve the clustering, since the dominant components, which contain most of the variation in the data, are highly influenced by the very noisy data points.By accounting for the variance in the log expression levels, our algorithm automatically down weights noisy values and ensures that the components we extract accurately reflect the structure of the data.The clustering is further improved when performed on the denoised reconstructed profiles, as these are the best estimates of the true profiles. This leads to much tighter and biologically plausible clusters in the data set under consideration, as shown in figure-8.

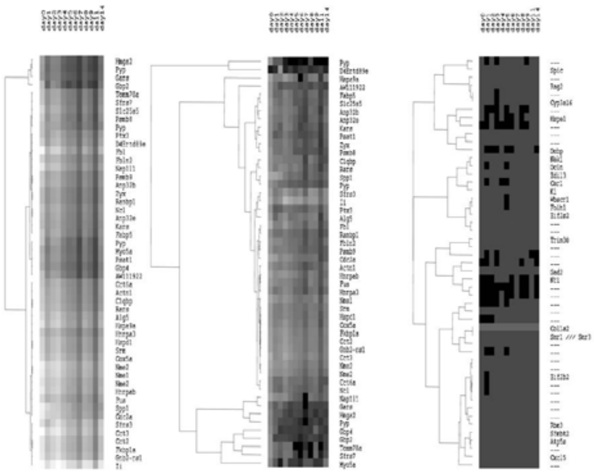


Figure 8: Hierarchical clustering of microarray data *left*: the top 50 genes in the second principal component obtained using our model (denoised profiles);*middle*: the

top 50 genes in the second principal component obtained using our model (original profiles) and *right*: the top 50 genes in the second principal component obtained by standard PCA. Clustering was performed using the GeneCluster software from the Eisen Lab.

Conclusion

The concluding remark is that, the PCA is a great tool to reduce dimensionality of biological data sets for visualization and interpretationbecause it is a simple, non-parametric method of extracting relevant information from confusing data sets. That's why PCA is used abundantly in all forms of analysis - from neuroscience to computer graphics. With a minimal additional effort PCA provides a roadmap for how to reduce a complex data set to a lower dimension to reveal the sometimes hidden, simplified dynamics that often underlie it. The data clusteringfollowed PCs enhance cluster quality only when right number of components or when the right set of PCs chosen.

References

[1] Tefferi, A., Bolander, M.E., Ansell, S.M., Wieben, E.D., and Spelsberg, T.C., 2002, "Primer on medical genomics: Microarray experiments and data analysis," Mayo Clin Proc., 77(9), pp. 927-940.

[2] Young, R.A., 2000, "Biomedical Discovery with DNA Arrays,"Cell, 2000(102), pp. 9-15.

[3] Zou, H., Hastie, T., and Tibshirani, R., 2004, "Sparse principal component analysis," Journal of Computational and Graphical Statistics, 11, pp. 545–581.

[4] Schena, M., Shalon, D., Davis, R.W., and Brown, P.O., 1995, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray," Science,270, pp. 467-470.

[5] Chee, M., Yang, R., Hubbel, E., Berno, A., Huang, X.C., Stern, D., Winkler, J., Lockhart, D.J., Morris, M.S., and Fodor, S.P.A., 1996 "Accessing genetic information with high-density DNA arrays," Science. 274, pp. 610-614.

[6] Chen, J.J.W., Wu, R., Yang, P.-C., Huang, J.Y., Sher, Y.P., Han, M.H., Kao, W.C., Lee, P.J., Chiu, T.F., Chang, F., Chu, Y.W., Wu, C.W., and Peck, K., 1998., "Profiling expression patterns and isolating differentially expressed genes by cDNA microarray system with colorimetry detection," Genomics, 51, pp.313-324.

[7] Duggan, D.J., Bittner, M., Chen, Y., Meltzer, P., and Trent, J.M., 1999, "Expression profiling using cDNA microarrays," Nature Genetics, 21, pp.10-14.

[8] Lander, E.S., 1999, "Array of hope," Nature Genetics, 21, pp.3-4

[9] Eisen, M.B., Spellman, P.T., Brown, P.O., and Botstein, D., 1998, "Cluster analysis and display of genome wide expression patterns," Proc. Natl. Acad. Sci., 95, pp. 14863-14868.

[10] Michaels, G.S., Carr, D.B., Askenazi, M., Fuhrman, S., Wen, X., and Somogyi, R., 1998, "Cluster analysis and data visualization of large-scale gene expression data," Pacific Symposium on Biocomputing, 3, pp. 42-53.

[11] Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., and Fucher, B., 1998, "Comprehensive identification of cell cylce-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization," Molecular Biology of the Cell, 9, pp. 3273-3297.

[12] Brazma, A., Jonassen, I., Vilo, J., and Ukkonen, E., 1998, "Predicting gene regulatory elements *in silico* on a genomic scale," Genome Research, 8, pp.1202-1215.

[13] D'Haeseleer, P., Wen, X., Fuhrman, S., and Somogyi, R., 1999, "Linear modelling of mRNA expression levels during CNS development and injury," Pacific Symposium on Biocomputing, 4, pp.41-52.

[14] Liang, S., Fuhrman, S., Somogyi, R., 1998, "REVEAL, a general reverse engineering algorithm for inference of genetic network architectures," Pacific Symposium on Biocomputing, 3, pp.18-29.

[15] Hastie, T. and Stuetzle, W., 1989, "Principal curves," Journal of the American Statistical Society, 84, pp. 502–516.

[16] Draghici, S., 2003, "Data analysis tools for DNA microarrays". Chapman and Hall/CRC, London.

[17] Eisen, M.B., Spellman, P.T., Brown, P.O., and Botstein, D., 1998, "Cluster analysis and display of genome-wide expression patterns," Proc. Nat. Acad. Sci., 95, pp14863-14868.

[18] Sagara, J. 2006, "Analysis of *Aspergillus oryzae* tRNA genes using multivariate analysis," Genome Informatics. pp56-59.

[19] Everitt, B.S., 1993, "Cluster Analysis". John Wiley & Sons, New York, NY.

[20] Jolliffe, I.T., Jones, B. and Morgan, B.J. T. 1980, "Cluster analysis of the elderly at home: a case study". Data analysis and Informatics, pp 745–757.

[21] Jolliffe, I. T., 1986, "Principal component analysis". New York : Springer-Verlag.Chang, 1983

[22] Chu, S., DeRisi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P.O. and Herskowitz, I.. 1998, "The transcriptional program of sporulation in budding yeast," Science, 282, 699–705.