

# Principal component analysis of gene expression profiles

Leif E. Peterson, Ph.D.

Dept. of Medicine; Dept. of Molec. & Human Genetics

Baylor College of Medicine

Faculty Center, Office 10.57, ext. 8-5386

## Introduction

- Principal components analysis (PCA) is useful for reproducing the total variance among a large number of variables using a much smaller number of artificial variables called “latent factors,” or principal components
- This lecture provides the multivariate statistical theory behind PCA
- This includes the principal factor solution to the factor model of the correlation matrix  $\mathbf{R}$ , extraction of factors (components) from  $\mathbf{R}$

- Note, we will be focusing on PCA of rows of an  $n \times p$  data matrix, which represent genes. The columns are represented by arrays (patients or samples)

## Multivariate statistical theory of principal components analysis

### Matrix definitions

Let  $\mathbf{M}_{(n,p)} \in \mathfrak{R}$  be the set of all data variables over the field of real numbers, and  $\mathbf{M}_{(p)} \in \mathfrak{R}$  the set of symmetric  $p$ -square matrices over the field of real numbers. Define the *data* matrix  $\mathbf{Y} \in \mathbf{M}_{n,p}(\mathfrak{R})$  with  $n$  rows and  $p$  columns as a function on the pairs of integers  $(i, j)$ ,  $1 \leq i \leq n$ ,  $1 \leq j \leq p$ , with values in  $\mathbf{Y}$  in which  $y_{ij}$  designates the value of  $\mathbf{Y}$  at the pair  $(i, j)$  shown

as

$$\mathbf{Y}_{n \times p} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1p} \\ y_{21} & y_{22} & \cdots & y_{2p} \\ \vdots & \vdots & \cdots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{np} \end{bmatrix} \quad (1)$$

Because of random fluctuations in results across the experiments (arrays), the data matrix  $\mathbf{Y}$  is standardized to remove noise. Standardization of  $\mathbf{Y}$  is performed using the sample mean vector,  $\bar{y}$ , consisting of array(column)-specific means and standard deviations

$$s_j = \left( \frac{\sum_{i=1}^n (y_{ij}^* - \bar{y}_j^*)^2}{\sum_{i=1}^n (y_{ij}^* - \bar{y}_j^*)^2 n - 1} \right)^{1/2} \quad (2)$$

Elements of the  $n \times p$  standardized data matrix  $\mathbf{Z}$  are

$$z_{ij} = (y_{ij} - \bar{y}_j) / s_j \quad (3)$$

The pairwise correlation between genes  $k$  and  $l$  given is then calculated

from the row means and row standard deviations of  $\mathbf{Z}$  in the form

$$r_{kl} = \frac{\sum_{m=1}^p ((z_{km} - \bar{z}_k)/s_k) ((z_{lm} - \bar{z}_l)/s_l)}{p} \quad (4)$$

with summation over arrays 1 to  $p$ . This results in an  $n \times n$  (“gene by gene”) correlation matrix

$$\mathbf{R}_{n \times n} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \dots & r_{nn} \end{bmatrix}. \quad (5)$$

## Principal component solution to the factor model of $\mathbf{R}$

By the Principal Axis Theorem, there exists a rotation matrix  $\mathbf{E}$  and diagonal matrix  $\mathbf{\Lambda}$  such that  $\mathbf{ERE}' = \mathbf{\Lambda}$ . Pre-multiplying both sides by  $\mathbf{E}$ , and post-multiplying by  $\mathbf{E}'$ , yields the principal form (or spectral

decomposition) of  $\mathbf{R}$  given as

$$\mathbf{R}_{n \times n} = \mathbf{E} \mathbf{\Lambda} \mathbf{E}' = \begin{bmatrix} e_{11} & e_{12} & \dots & e_{1n} \\ e_{21} & e_{22} & \dots & e_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ e_{n1} & e_{n2} & \dots & e_{nn} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \begin{bmatrix} e_{11} & e_{21} & \dots & e_{n1} \\ e_{12} & e_{22} & \dots & e_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ e_{1n} & e_{2n} & \dots & e_{nn} \end{bmatrix} \quad (6)$$

where columns of  $\mathbf{E}$  and  $\mathbf{E}'$  are the *eigenvectors* and diagonal entries of  $\mathbf{\Lambda}$  are the *eigenvalues*. Eigenvectors and associated eigenvalues are extracted from  $\mathbf{E} \mathbf{\Lambda} \mathbf{E}'$  using the iterative Jacobi method.

The Jacobi method, which is a “foolproof” but time-consuming algorithm for finding the eigenvalues and eigenvectors of a symmetric square matrix, goes as follows. Before iterating, set  $\mathbf{\Lambda}_0 = \mathbf{R}$  and  $\mathbf{E}_0 = \mathbf{I}$ . Use Meglicki’s [33] method to speed up the Jacobi transformation by looping 50 times, calculating a threshold each time as

$$threshold = \begin{cases} \frac{1}{5} \frac{1}{n^2} \sum_{p=1}^{n-1} \sum_{q=p+1}^n \lambda_{pq} & 1 \leq iter \leq 3 \\ 0 & iter > 3 \end{cases} \quad (7)$$

where  $\lambda_{pq}$  ( $p < q$ ) are off-diagonal elements in the upper triangular of  $\Lambda_0$  and  $n$  is the number of genes. During each iteration, loop through all  $\lambda_{pq}$  off-diagonal elements of  $\Lambda_0$  (i.e.,  $p=1$  to  $n-1$ ;  $q = p+1$  to  $n$ ). Within the loops for  $p$  and  $q$ , if  $\lambda_{pq}$  exceeds *threshold* for that iteration, then rotate the off-diagonal element  $\lambda_{pq}$  and calculate new corresponding eigenvalues  $\lambda_{pp}$  and  $\lambda_{qq}$  on the diagonal of  $\Lambda_0$ . After the third iteration, off-diagonals,  $\lambda_{pq}$ , that do not strongly influence eigenvalues undergoing rotation, i.e.,  $|\lambda_{pp}| + 100|\lambda_{pq}| = |\lambda_{pp}|$  and  $|\lambda_{qq}| + 100|\lambda_{pq}| = |\lambda_{qq}|$  are set to zero. During the rotations, let  $\mu = (\lambda_{pp} - \lambda_{qq})/2$  and calculate  $\omega$  as

$$\omega \equiv \text{sgn}(\mu) \frac{\lambda_{pq}}{\sqrt{\lambda_{pq}^2 + \mu^2}} \quad . \quad (8)$$

Calculate trigonometric relationships

$$\sin(\theta) = \frac{\omega}{\sqrt{2(1 + \sqrt{1 - \omega^2})}} \quad (9)$$

$$\cos(\theta) = \sqrt{1 - \sin^2(\theta)}. \quad (10)$$

and substitute into the  $n \times n$  rotation matrix  $\mathbf{S}$  as

$$\mathbf{S} = \begin{bmatrix} \mathbf{I} & 0 & 0 \\ 0 & \mathbf{T} & 0 \\ 0 & 0 & \mathbf{I} \end{bmatrix} \quad (11)$$

where the top row partition has  $p-1$  rows, the bottom row partition has  $n - q$  rows, and the  $\mathbf{T}$  matrix with  $q - p+1$  rows is

$$\mathbf{T} = \begin{bmatrix} \cos(\theta) & 0 & 0 & \cdots & 0 & 0 & -\sin(\theta) \\ 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ & & \vdots & & & & \\ 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ \sin(\theta) & 0 & 0 & \cdots & 0 & 0 & \cos(\theta) \end{bmatrix} . \quad (12)$$

The iterative matrix operations  $\mathbf{\Lambda}_{m+1} = \mathbf{S}\mathbf{\Lambda}_m\mathbf{S}'$  and  $\mathbf{E}_m = \mathbf{S}\mathbf{E}_m$  are carried out during rotations within each iteration. When the sum of the off-diagonal elements of  $\mathbf{\Lambda}_m$  is less than, say, 0.01, then terminate the program. Most runs are completed after about 6-8 iterations. After  $m$  iterations,  $\mathbf{\Lambda}_m$  will be a diagonal matrix with near-zero off-diagonals and

diagonal elements equal to the  $n$  eigenvalues of  $\mathbf{R}$ , and  $\mathbf{E}_m$  will be the matrix of eigenvectors of  $\mathbf{R}$ . A check should be made to ensure that  $\mathbf{R}=\mathbf{E}\mathbf{\Lambda}\mathbf{E}'$ .

## Extraction of components

Now that the eigenvalues of  $\mathbf{R}$  are known, they are sorted in descending order. Because the total variance of  $\mathbf{R}$  is  $n$  and each eigenvalue contributes  $\lambda_{ii}/n$  to total variance, only components whose eigenvalues exceed unity,  $\lambda_{ii} > 1$ , are extracted from  $\mathbf{\Lambda}$  and used in the quicksort. (At this point, notation for eigenvalues changes from  $\lambda_{nn}$  to  $\lambda_m$ ). Thus,  $m$  sorted eigenvalues are selected with values greater than unity so that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$ .

Since each eigenvalue represents a *component*, there will be  $m$  components selected. The correlation between each gene expression profile and

the  $m$  components is represented by a matrix of *loadings* given as

$$\mathbf{L}_{n \times n} = \begin{bmatrix} \sqrt{\lambda_1}e_{11} & \sqrt{\lambda_2}e_{12} & \dots & \sqrt{\lambda_m}e_{1m} \\ \sqrt{\lambda_1}e_{21} & \sqrt{\lambda_2}e_{22} & \dots & \sqrt{\lambda_m}e_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{\lambda_1}e_{n1} & \sqrt{\lambda_2}e_{n2} & \dots & \sqrt{\lambda_m}e_{nm} \end{bmatrix} \quad (13)$$

where rows represent gene expression profiles and columns represent components, and, for example,  $\sqrt{\lambda_1}e_{11}$  is the loading (correlation) between gene 1 and component 1.