

Principal Components Analysis (PCA) and Singular Value Decomposition (SVD) with applications to Microarrays

Prof. Tesler

Math 283
Fall 2015

Covariance

- Let X and Y be random variables, possibly dependent.
- Recall that the *covariance* of X and Y is defined as

$$\text{Cov}(X, Y) = E((X - \mu_X)(Y - \mu_Y))$$

and that an alternate formula is

$$\text{Cov}(X, Y) = E(XY) - E(X)E(Y)$$

- Previously we used

$$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y) + 2 \text{Cov}(X, Y)$$

and

$$\text{Var}(X_1 + X_2 + \cdots + X_n) = \text{Var}(X_1) + \cdots + \text{Var}(X_n)$$

Covariance properties

Covariance properties

- $\text{Cov}(X, X) = \text{Var}(X)$
- $\text{Cov}(X, Y) = \text{Cov}(Y, X)$
- $\text{Cov}(aX + b, cY + d) = ac \text{Cov}(X, Y)$

Sign of covariance $\text{Cov}(X, Y) = E((X - \mu_X)(Y - \mu_Y))$

- **When $\text{Cov}(X, Y)$ is positive:**
there is a tendency to have $X > \mu_X$ when $Y > \mu_Y$ and vice-versa, and $X < \mu_X$ when $Y < \mu_Y$ and vice-versa.
- **When $\text{Cov}(X, Y)$ is negative:**
there is a tendency to have $X > \mu_X$ when $Y < \mu_Y$ and vice-versa, and $X < \mu_X$ when $Y > \mu_Y$ and vice-versa.
- **When $\text{Cov}(X, Y) = 0$:**
 - a) X and Y **might** be independent, but it's not guaranteed.
 - b) $\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y)$

Sample variance

Variance of a random variable:

$$\sigma^2 = \text{Var}(X) = E((X - \mu_X)^2) = E(X^2) - (E(X))^2$$

Sample variance from data x_1, \dots, x_n :

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

Vector formula:

Centered data: $M = [x_1 - \bar{x} \quad x_2 - \bar{x} \quad \cdots \quad x_n - \bar{x}]$

$$s^2 = \frac{M \cdot M}{n-1} = \frac{M M'}{n-1}$$

Sample covariance

Covariance between random variables X, Y :

$$\sigma_{XY} = \text{Cov}(X, Y) = E((X - \mu_X)(Y - \mu_Y)) = E(XY) - E(X)E(Y)$$

Sample covariance from data $(x_1, y_1), \dots, (x_n, y_n)$:

$$s_{XY} = \text{cov}(x, y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) = \frac{1}{n-1} \left(\sum_{i=1}^n x_i y_i \right) - \frac{n}{n-1} \bar{x} \bar{y}$$

Vector formula:

$$M_X = \begin{bmatrix} x_1 - \bar{x} & x_2 - \bar{x} & \cdots & x_n - \bar{x} \end{bmatrix}$$

$$M_Y = \begin{bmatrix} y_1 - \bar{y} & y_2 - \bar{y} & \cdots & y_n - \bar{y} \end{bmatrix}$$

$$s_{XY} = \frac{M_X \cdot M_Y}{n-1} = \frac{M_X M_Y'}{n-1}$$

Covariance matrix

For problems with many simultaneous random variables, put them into vectors:

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

and then form a *covariance matrix*:

$$\text{Cov}(\vec{X}, \vec{Y}) = \begin{bmatrix} \text{Cov}(R, T) & \text{Cov}(R, U) & \text{Cov}(R, V) \\ \text{Cov}(S, T) & \text{Cov}(S, U) & \text{Cov}(S, V) \end{bmatrix}$$

In matrix/vector notation,

$$\text{Cov}(\vec{X}, \vec{Y}) = E [(\vec{X} - E(\vec{X})) (\vec{Y} - E(\vec{Y}))']$$

Covariance matrix (a.k.a. Variance-Covariance matrix)

Often there's one vector with all the variables:

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

$$\begin{aligned} \text{Cov}(\vec{X}) &= \text{Cov}(\vec{X}, \vec{X}) \\ &= E [(\vec{X} - E(\vec{X})) (\vec{X} - E(\vec{X}))'] \\ &= \begin{bmatrix} \text{Cov}(R, R) & \text{Cov}(R, S) & \text{Cov}(R, T) \\ \text{Cov}(S, R) & \text{Cov}(S, S) & \text{Cov}(S, T) \\ \text{Cov}(T, R) & \text{Cov}(T, S) & \text{Cov}(T, T) \end{bmatrix} \\ &= \begin{bmatrix} \text{Var}(R) & \text{Cov}(R, S) & \text{Cov}(R, T) \\ \text{Cov}(R, S) & \text{Var}(S) & \text{Cov}(S, T) \\ \text{Cov}(R, T) & \text{Cov}(S, T) & \text{Var}(T) \end{bmatrix} \end{aligned}$$

The matrix is symmetric. The diagonal entries are ordinary variances.

Covariance matrix properties

$$\text{Cov}(\vec{X}, \vec{Y}) = \text{Cov}(\vec{Y}, \vec{X})'$$

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

$$\text{Cov}(\vec{X}, C\vec{Y} + \vec{D}) = \text{Cov}(\vec{X}, \vec{Y})C'$$

$$\text{Cov}(A\vec{X} + \vec{B}) = A \text{Cov}(\vec{X})A'$$

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

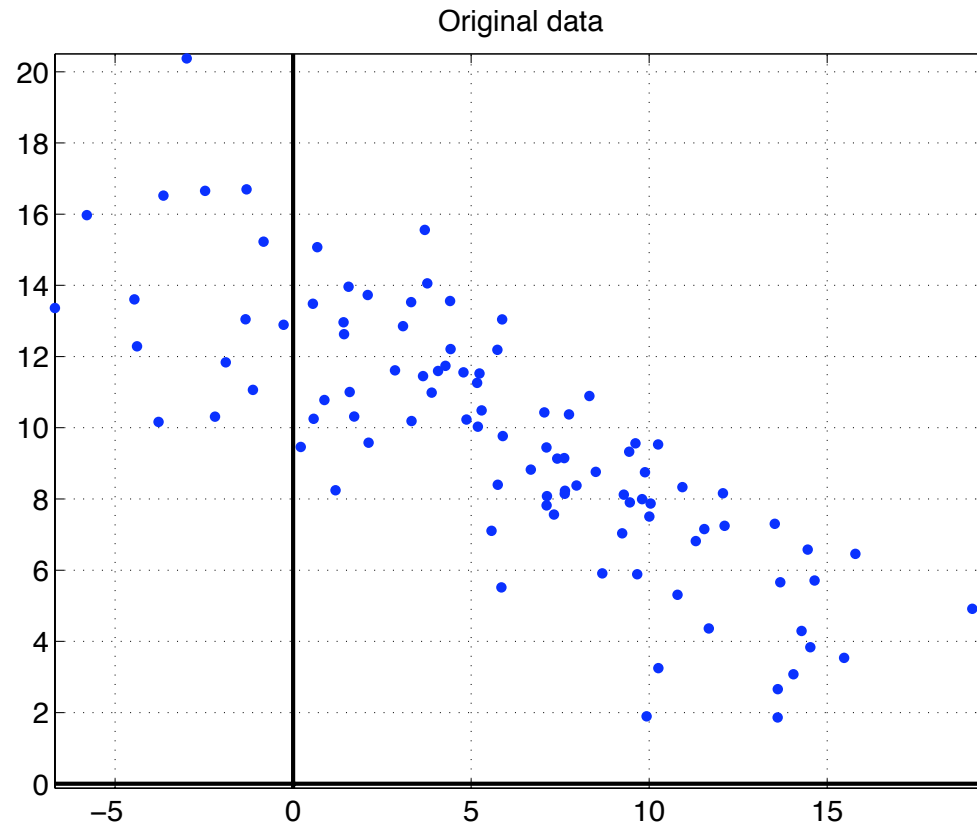
$$\text{Cov}(\vec{X}, \vec{Y}_1 + \vec{Y}_2) = \text{Cov}(\vec{X}, \vec{Y}_1) + \text{Cov}(\vec{X}, \vec{Y}_2)$$

A, C are constant matrices, \vec{B}, \vec{D} are constant vectors, and all dimensions must be correct for matrix arithmetic.

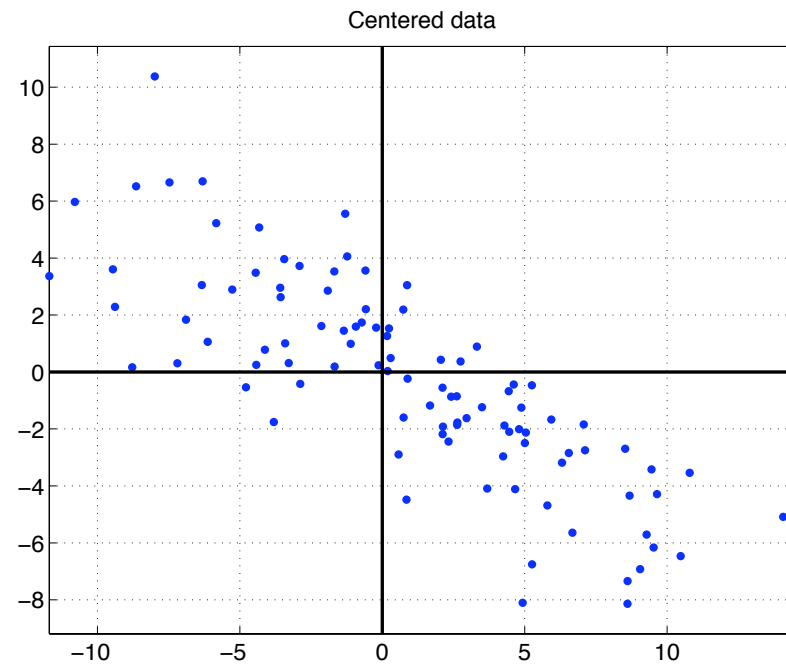
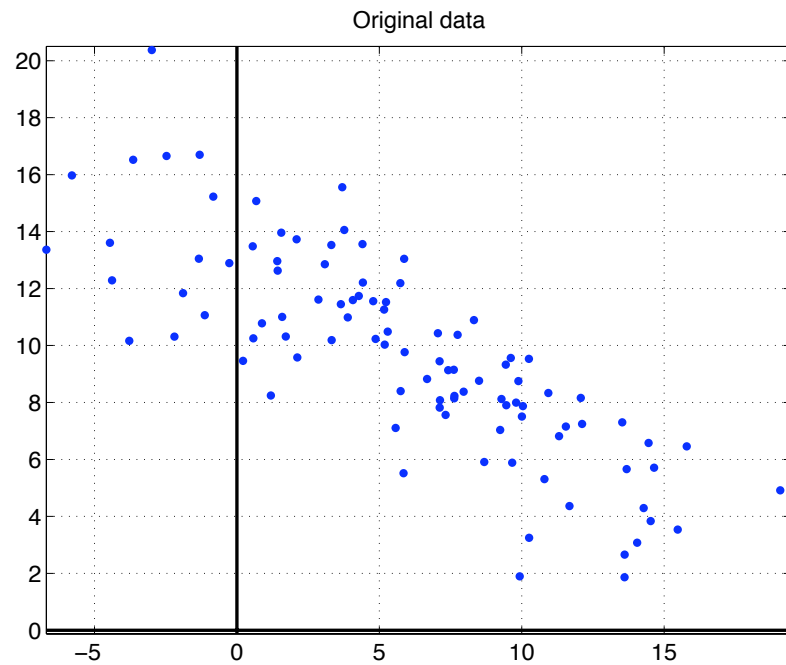
Example (2D, but works for higher dimensions too)

Data $(x_1, y_1), \dots, (x_{100}, y_{100})$:

$$M_0 = \begin{bmatrix} x_1 & \cdots & x_{100} \\ y_1 & \cdots & y_{100} \end{bmatrix} = \begin{bmatrix} 3.0858 & 0.8806 & 9.8850 & \cdots & 4.4106 \\ 12.8562 & 10.7804 & 8.7504 & \cdots & 13.5627 \end{bmatrix}$$



Centered data



Computing sample covariance matrix

- **Original data:** 100 (x, y) points in a 2×100 matrix M_0 :

$$M_0 = \begin{bmatrix} x_1 & \cdots & x_{100} \\ y_1 & \cdots & y_{100} \end{bmatrix} = \begin{bmatrix} 3.0858 & 0.8806 & 9.8850 & \cdots & 4.4106 \\ 12.8562 & 10.7804 & 8.7504 & \cdots & 13.5627 \end{bmatrix}$$

- **Centered data:** subtract \bar{x} from x 's and \bar{y} from y 's to get M ; here $\bar{x} = 5$, $\bar{y} = 10$:

$$M = \begin{bmatrix} -1.9142 & -4.1194 & 4.8850 & \cdots & -0.5894 \\ 2.8562 & 0.7804 & -1.2496 & \cdots & 3.5627 \end{bmatrix}$$

- **Sample covariance:**

$$\begin{aligned} C &= \frac{M M'}{100 - 1} = \begin{bmatrix} 31.9702 & -16.5683 \\ -16.5683 & 13.0018 \end{bmatrix} \\ &= \begin{bmatrix} s_{XX} & s_{XY} \\ s_{YX} & s_{YY} \end{bmatrix} = \begin{bmatrix} s_X^2 & s_{XY} \\ s_{XY} & s_Y^2 \end{bmatrix} \end{aligned}$$

Orthonormal matrix

- Recall that for vectors \vec{v}, \vec{w} , we have $\vec{v} \cdot \vec{w} = |\vec{v}| |\vec{w}| \cos(\theta)$, where θ is the angle between the vectors.
- *Orthogonal* means perpendicular.
 \vec{v} and \vec{w} are orthogonal when the angle between them is $\theta = 90^\circ = \frac{\pi}{2}$ radians. So $\cos(\theta) = 0$ and $\vec{v} \cdot \vec{w} = 0$.
- Vectors $\vec{v}_1, \dots, \vec{v}_n$ are *orthonormal* when
 - $\vec{v}_i \cdot \vec{v}_j = 0$ for $i \neq j$ (different vectors are orthogonal)
 - $\vec{v}_i \cdot \vec{v}_i = 1$ for all i (each vector has length 1; they are all unit vectors)
 - In short: $\vec{v}_i \cdot \vec{v}_j = \delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j. \end{cases}$

Orthonormal matrix

- Form an $n \times n$ matrix of orthonormal vectors

$$V = [\vec{v}_1 \mid \cdots \mid \vec{v}_n]$$

by loading n -dimensional column vectors into the columns of V .

- Transpose it to convert the vectors to row vectors:

$$V' = \begin{bmatrix} \vec{v}'_1 \\ \vec{v}'_2 \\ \vdots \\ \vec{v}'_n \end{bmatrix}$$

- $(V'V)_{ij}$ is the i^{th} row of V' dotted with the j^{th} column of V :

$$(V'V)_{ij} = \vec{v}_i \cdot \vec{v}_j = \delta_{ij} \quad V'V = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

- Thus, $V'V = I$ ($n \times n$ identity matrix), so $V' = V^{-1}$.
- An $n \times n$ matrix V is **orthonormal** when $V'V = I$ (or equivalently, $VV' = I$), where I is the $n \times n$ identity matrix.

Diagonalizing the sample covariance matrix C

- C is a real-valued symmetric matrix. It can be shown that it can be diagonalized $C = VDV'$ where V is orthonormal, so $V^{-1} = V'$.

$$\begin{matrix} C \\ \left[\begin{array}{cc} 31.9702 & -16.5683 \\ -16.5683 & 13.0018 \end{array} \right] \end{matrix} = \begin{matrix} V \\ \left[\begin{array}{cc} -0.8651 & -0.5016 \\ 0.5016 & -0.8651 \end{array} \right] \end{matrix} \begin{matrix} D \\ \left[\begin{array}{cc} 41.5768 & 0 \\ 0 & 3.3952 \end{array} \right] \end{matrix} \begin{matrix} V' \\ \left[\begin{array}{cc} -0.8651 & 0.5016 \\ -0.5016 & -0.8651 \end{array} \right] \end{matrix}$$

- Also, all eigenvalues are ≥ 0 (" C is *positive semidefinite*"): For all vectors \vec{w} ,

$$\vec{w}'C\vec{w} = \frac{\vec{w}'MM'\vec{w}}{n-1} = \frac{|M'\vec{w}|^2}{n-1} \geq 0$$

Eigenvector equation $C\vec{w} = \lambda\vec{w}$ gives $\vec{w}'C\vec{w} = \lambda\vec{w}'\vec{w} = \lambda|\vec{w}|^2$.
So $\lambda|\vec{w}|^2 = \vec{w}'C\vec{w} \geq 0$, giving $\lambda \geq 0$.

- It is conventional to put the eigenvalues on the diagonal of D in decreasing order: $\lambda_1 \geq \lambda_2 \geq \dots \geq 0$.

Principal axes

- The columns of V are the right eigenvectors of C .
- Multiply each eigenvector by the square root of its eigenvalue to get the principal components.

Eigenvalue	Eigenvector	PC
41.5768	$\begin{bmatrix} -0.8651 \\ 0.5016 \end{bmatrix}$	$\begin{bmatrix} -5.5782 \\ 3.2343 \end{bmatrix}$
3.3952	$\begin{bmatrix} -0.5016 \\ -0.8651 \end{bmatrix}$	$\begin{bmatrix} -0.9242 \\ -1.5940 \end{bmatrix}$

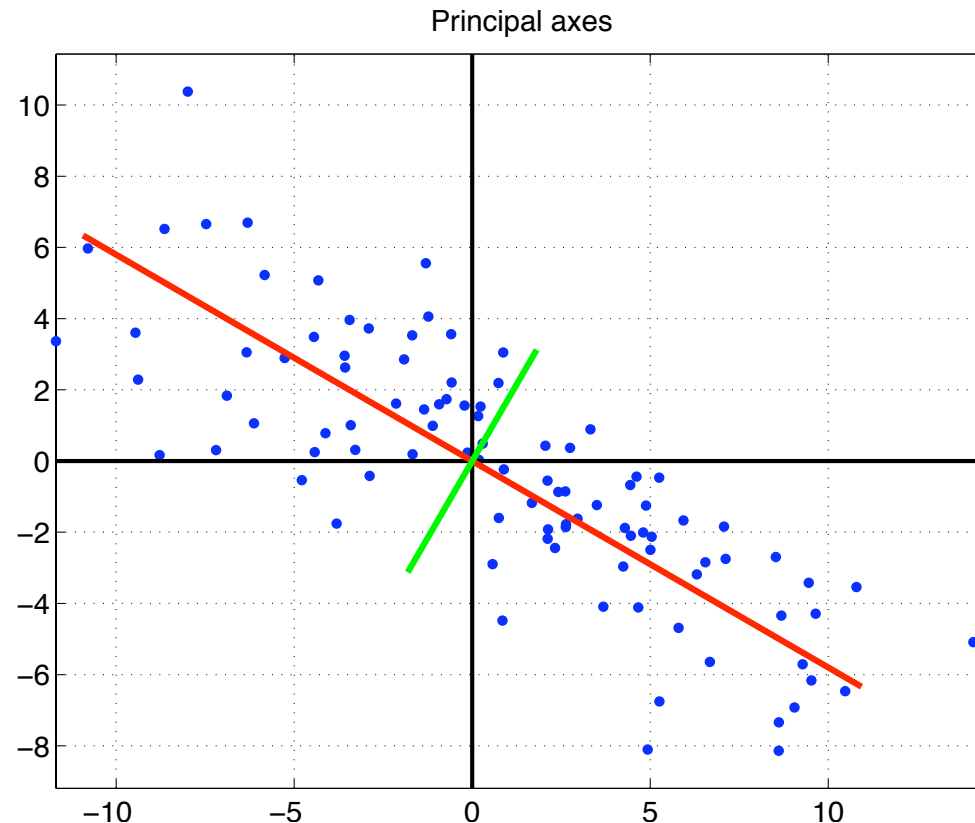
- Put them into the columns of a matrix:

$$P = V \sqrt{D} = \begin{bmatrix} -5.5782 & -0.9242 \\ 3.2343 & -1.5940 \end{bmatrix}$$

- $C = VDV' = V \sqrt{D} \sqrt{D}' V' = (V \sqrt{D})(V \sqrt{D})' = PP'$

Principal axes

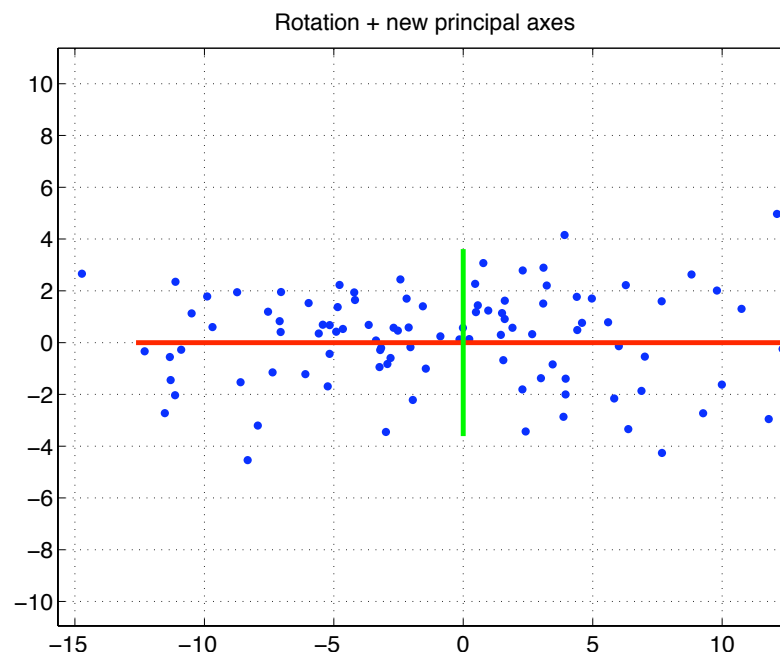
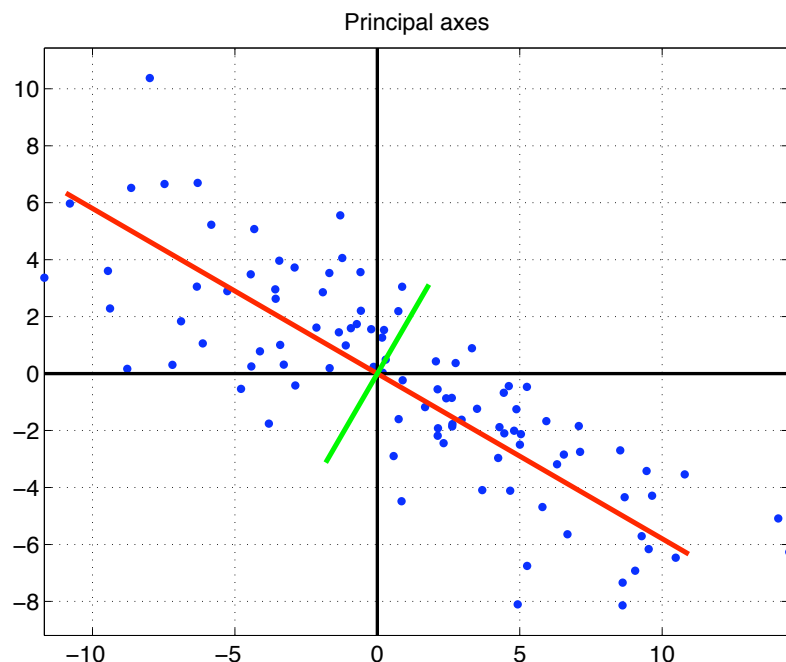
- Plot the centered data with lines along the principal axes:



- Sum of squared perpendicular distances of data points to first PC line (red) is minimum among all lines through origin.
- i th PC is perpendicular to the previous ones, and the sum of squared perpendicular distances to the span (line, plane, ...) of the first i PCs is minimum among all i -dim. spaces through origin.

Rotate axes

Transform M to $M_2 = V'M$ and plot points given by the columns of M_2 :



- From linear algebra, a linear transformation $M \mapsto AM$ does a combination of rotations, reflections, scaling, shearing, and orthogonal projections.
- V is orthonormal, so $M_2 = V'M$ rotates/reflects all the data.
- $M = VM_2$ recovers centered data M from rotated data M_2 .
- The sample covariance matrix of M_2 is

$$\frac{M_2 M_2'}{n-1} = \frac{(V'M)(V'M)'}{n-1} = \frac{V'MM'V}{n-1} = V' \left(\frac{MM'}{n-1} \right) V = V'CV = D$$

- Note $C = VDV'$ and $V' = V^{-1}$, so $D = V^{-1}C(V')^{-1} = V'CV$.

New coordinates

- The rotated data has new coordinates $(t_1, u_1), \dots, (t_{100}, u_{100})$ and covariance matrix D :

$$V'CV = D = \begin{bmatrix} \text{var}(T) & \text{cov}(T, U) \\ \text{cov}(T, U) & \text{var}(U) \end{bmatrix} = \begin{bmatrix} 41.5768 & 0 \\ 0 & 3.3952 \end{bmatrix}$$

- In D , the *total variance* is $\text{var}(T) + \text{var}(U) = 44.9720$.
Note that this is the sum of the eigenvalues, $\lambda_1 + \lambda_2 + \dots$.
- The *trace of a matrix* is the sum of its diagonal entries.
So the total variance is $\text{Tr}(D) = \lambda_1 + \lambda_2 + \dots$.
- General linear algebra fact:** $\text{Tr}(X) = \text{Tr}(AXA^{-1})$.
So $\text{Tr}(C) = \text{Tr}(VDV') = \text{Tr}(VDV^{-1}) = \text{Tr}(D)$.
Below, $\text{Tr}(C) = \text{Tr}(D) = 44.9720$.

$$C = \begin{bmatrix} 31.9702 & -16.5683 \\ -16.5683 & 13.0018 \end{bmatrix} \quad D = \begin{bmatrix} 41.5768 & 0 \\ 0 & 3.3952 \end{bmatrix}$$

Part of variance explained by each axis

- The part of the variance *explained* by each axis is λ_i /total variance:

Eigenvector	Eigenvalue	Explained
$\begin{bmatrix} -0.8651 \\ 0.5016 \end{bmatrix}$	41.5768	$41.5768/44.9720 = 92.45\%$
$\begin{bmatrix} -0.5016 \\ -0.8651 \end{bmatrix}$	3.3952	$3.3952/44.9720 = 7.55\%$
Total	44.9720	100%

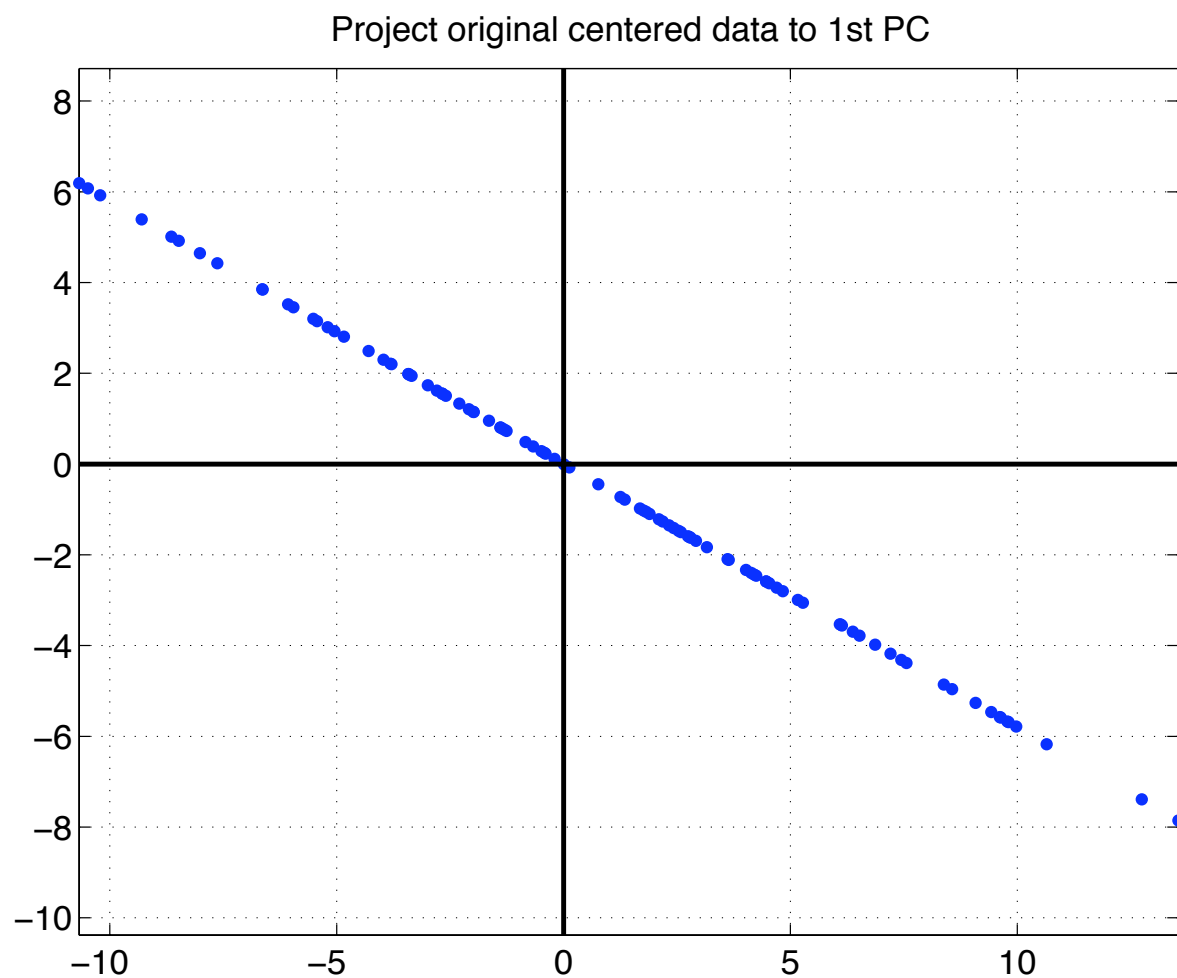
- This is an application of $\text{Cov}(A\vec{X}) = A \text{Cov}(\vec{X})A'$:

$$\text{Cov} \left(V' \begin{bmatrix} X \\ Y \end{bmatrix} \right) = V' \text{Cov} \left(\begin{bmatrix} X \\ Y \end{bmatrix} \right) V$$

Dimension reduction

To clean up “noise,” set all $u_i = 0$ and rotate back:

$$V \begin{bmatrix} t_1 & t_2 & t_3 & \cdots \\ 0 & 0 & 0 & \cdots \end{bmatrix} = \begin{bmatrix} \tilde{x}_1 & \tilde{x}_2 & \tilde{x}_3 & \cdots \\ \tilde{y}_1 & \tilde{y}_2 & \tilde{y}_3 & \cdots \end{bmatrix}$$



Dimension reduction

- Say we want to keep enough information to explain 90% of the variance.
- Take enough top PCs to explain $\geq 90\%$ of the variance.
- Let M_3 be M_2 (rotated data) with the remaining coordinates zeroed out.
- Rotate it back to the original axes with VM_3 .
- In other applications, a dominant signal can be suppressed by zeroing out the coordinates for the top PCs instead of the bottom PCs.

Variations for PCA (and SVD, upcoming)

- Some people reverse the roles of rows and columns of M .
- In some applications, M is “centered” (subtract off row means) and in others, it’s not.
- If the ranges on the variables (rows) are very different, the data might be rescaled in each row to make similar ranges. For example, replace each row by Z -scores for the row.

Sensitivity to scaling

- PCA was originally designed for measurements in ordinary space, so all axes had the same units (e.g., cm or inches) and equivalent results would be obtained no matter what units were used.
- It's problematic to mix physical quantities with different units:
 - Length of (a, b) in (seconds, mm): $\sqrt{a^2 + b^2}$
(adding sec² plus mm² is not legitimate!)
Convert to (hours, miles): $\left| \left(\frac{a}{3600}, \frac{b}{1609344} \right) \right| = \sqrt{\frac{a^2}{3600^2} + \frac{b^2}{1609344^2}}$
 - Angles are also distorted by this unit conversion:
$$\arctan(b/a) \neq \arctan\left(\frac{a}{3600} / \frac{b}{1609344}\right).$$
 - $|(0^\circ\text{C}, 0^\circ\text{C})| = 0$ vs. $|(32^\circ\text{F}, 32^\circ\text{F})| = 32\sqrt{2}$.
Both $^\circ\text{C}$ and $^\circ\text{F}$ use an arbitrary “zero” instead of absolute zero.
- PCA is sensitive to differences in the scale, offset, and ranges of the variables. Rescaling one row w/o the others changes angles and lengths nonuniformly, and changes eigenvalues and eigenvectors in an inequivalent way.
- Typically addressed by replacing each row with Z-scores.

Microarrays

- Before we considered single genes where “red” or “green” (positive or negative expression level) distinguished the classes.



- If x_i is the expression level of gene i then

$$L = a_1x_1 + a_2x_2 + \cdots$$

is a linear combination of genes.

- Next up: a method that finds linear combinations of genes where $L > C$ and $L < C$ distinguish two classes, for some constant C . So $L = C$ is a line / plane / etc. that splits the multidimensional space of expression levels.
- Different classes are not always separated in this fashion. In some situations, nonlinear relations may be required.

Microarrays

- Consider an experiment with 80 microarrays, each with 10000 spots.
- M is 10000×80 .
- $C = \frac{MM'}{80-1}$ is 10000×10000 !
- $M'M$ is 80×80 .
- We will see that MM' has the same 80 eigenvalues as $M'M$, plus an additional $10000 - 80 = 9920$ eigenvalues equal to 0.
- Some of the 80 eigenvalues of $M'M$ may also be 0.
For centered data, all row sums of M are 0 so $[1, \dots, 1]'$ is an eigenvector of $M'M$ with eigenvalue 0.
- We will see we can work with the smaller of MM' or $M'M$.

Singular Value Decomposition (SVD)

Let M be a $p \times q$ matrix (not necessarily “centered”).

The *Singular Value Decomposition* of M is $M = USV'$, where

- U is orthonormal, $p \times p$.
- V is orthonormal, $q \times q$.
- S is a diagonal $p \times q$ matrix, $s_1 \geq s_2 \geq \dots \geq 0$.
- If M is 5×3 , this would look like

$$\begin{matrix} M \\ \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \end{matrix} = \begin{matrix} U \\ \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \end{matrix} \begin{matrix} S \\ \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{matrix} \begin{matrix} V' \\ \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \end{matrix}$$

“Compact” SVD

For $p > q$: The bottom $p - q$ rows of S are all 0. Remove them. Keep only the first q rows of S and first q columns of U .

- U is orthonormal, $p \times q$.
- V is orthonormal, $q \times q$.
- S is a diagonal $p \times q$ matrix, $s_1 \geq s_2 \geq \dots \geq 0$.
- If M is 5×3 , this would look like

$$\overset{M}{\begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}} = \overset{U}{\begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}} \overset{S}{\begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{bmatrix}} \overset{V'}{\begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}}$$

- For $q > p$: keep only the first p columns of S and first p rows of V .
- Matlab and R have options for full or compact form in `svd(M)`.

Computing the SVD

- $M'M = (VS'U')(USV') = V(S'S)V' = V \begin{bmatrix} s_1^2 & 0 & 0 \\ 0 & s_2^2 & 0 \\ 0 & 0 & s_3^2 \end{bmatrix} V'$
- $MM' = (USV')(VS'U') = U(SS')U' = U \begin{bmatrix} s_1^2 & 0 & 0 & 0 & 0 \\ 0 & s_2^2 & 0 & 0 & 0 \\ 0 & 0 & s_3^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} U'$
- This diagonalization of $M'M$ and MM' shows they have the same eigenvalues up to the dimension of the smaller matrix.
The larger matrix has all additional eigenvalues equal to 0.
- **Compute the SVD using whichever gives smaller dimensions!**

Computing the SVD

- $M'M = (VS'U')(USV') = V(S'S)V' = V \begin{bmatrix} s_1^2 & 0 & 0 \\ 0 & s_2^2 & 0 \\ 0 & 0 & s_3^2 \end{bmatrix} V'$
- $MM' = (USV')(VS'U') = U(SS')U' = U \begin{bmatrix} s_1^2 & 0 & 0 & 0 & 0 \\ 0 & s_2^2 & 0 & 0 & 0 \\ 0 & 0 & s_3^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} U'$
- First method (recommended when $p \geq q$):
 - Diagonalize $M'M = VDV'$.
 - Compute $p \times q$ matrix S with $S_{ii} = \sqrt{D_{ii}}$ and 0's elsewhere.
 - The *pseudoinverse* of S is S^{-1} : replace each nonzero diagonal entry of S by its reciprocal, and transpose to get a $q \times p$ matrix.
 - Compute $U = MVS^{-1}$.
- $q \geq p$ is analogous: diagonalize $MM' = UDU'$; compute S from D ; then compute $V = (S^{-1}U'M)'$.
- `svd(M)` in both Matlab and R.

Singular values and singular vectors

- Let M be a $p \times q$ matrix (not necessarily centered). Suppose
 - s is a scalar.
 - \vec{v} is a $q \times 1$ unit vector (column vector).
 - \vec{u} is a $p \times 1$ unit vector (column vector).

- s is a *singular value* of M with *right singular vector* \vec{v} and *left singular vector* \vec{u} if

$$M\vec{v} = s\vec{u} \quad \text{and} \quad \vec{u}'M = s\vec{v}' \quad (\text{same as } M'\vec{u} = s\vec{v}).$$

- Break U and V into columns

$$U = [\vec{u}_1 \mid \vec{u}_2 \mid \cdots \mid \vec{u}_p]$$

$$V = [\vec{v}_1 \mid \vec{v}_2 \mid \cdots \mid \vec{v}_q]$$

Then $M\vec{v}_i = s_i\vec{u}_i$ and $M'\vec{u}_i = s_i\vec{v}_i$ for i up to $\min(p, q)$.

If $p > q$: $M'\vec{u}_i = \vec{0}$ for $i > q$. If $q > p$: $M\vec{v}_i = \vec{0}$ for $i > p$.

- **To get full-sized $M = USV'$ from compact ($p \geq q$ case):** choose the remaining columns of U from the nullspace of M' in such a way that the columns of U are an orthonormal basis of \mathbb{R}^p .

Relation between PCA and SVD

Previous computation for PCA

- Start with centered data matrix M (n columns).
- Compute covariance matrix, diagonalize it, compute P :

$$C = \frac{MM'}{n-1} = VDV' = PP' \quad \text{where} \quad P = V \sqrt{D}$$

Computing PCA using SVD

- In terms of the SVD factorization $M = USV'$, covariance is

$$\begin{aligned} C &= \frac{MM'}{n-1} = \frac{(USV')(VS'U')}{n-1} = \frac{U(SS')U'}{n-1} \\ &= UDU' \quad \text{where} \quad D = \frac{SS'}{n-1} \\ &= PP' \quad \text{where} \quad P = \frac{US}{\sqrt{n-1}} \end{aligned}$$

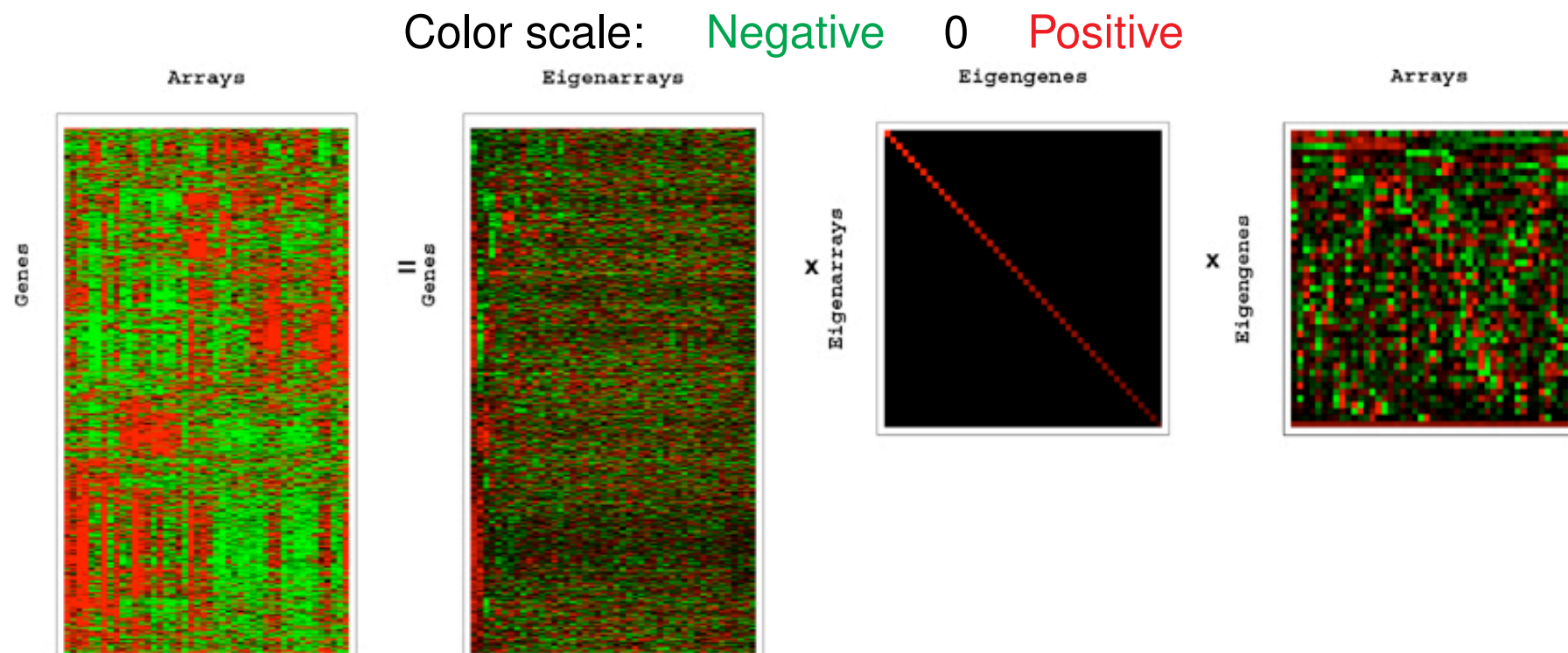
- Variance for i th component is $\frac{s_i^2}{n-1}$
- Note:* there were minor notation adjustments to deal with $n-1$.

SVD in microarrays

- Nielsen et al.¹ studied tumors in six types of tissue.
- 41 tissue samples and 46 microarray slides
- They switched microarray platforms in the middle of the experiment:
 - The first 26 slides have 22,654 spots (22K).
 - The next 20 slides have 42,611 spots (42K) (mostly a superset).
 - Five of the samples were done on both 22K and 42K platforms.
- 7425 spots were in common to both platforms, had good signal across all slides, and had sample variance above a certain threshold. So M is 7425×46 .

¹ *Molecular characterisation of soft tissue tumours: a gene expression study*, Lancet (2002) 359: 1301–1307.

SVD in microarrays



Webfigure 3a

Nielsen et al., supplementary material.

http://genome-www.stanford.edu/sarcoma/Supplemental_data.shtml

- The compact form $M = USV'$ is shown above.
- They call the columns of U “*eigenarrays*” and the columns of V (rows of V') “*eigengenes*.”
 - An *eigenarray* is a linear combination of arrays.
 - An *eigengene* is a linear combination of genes.

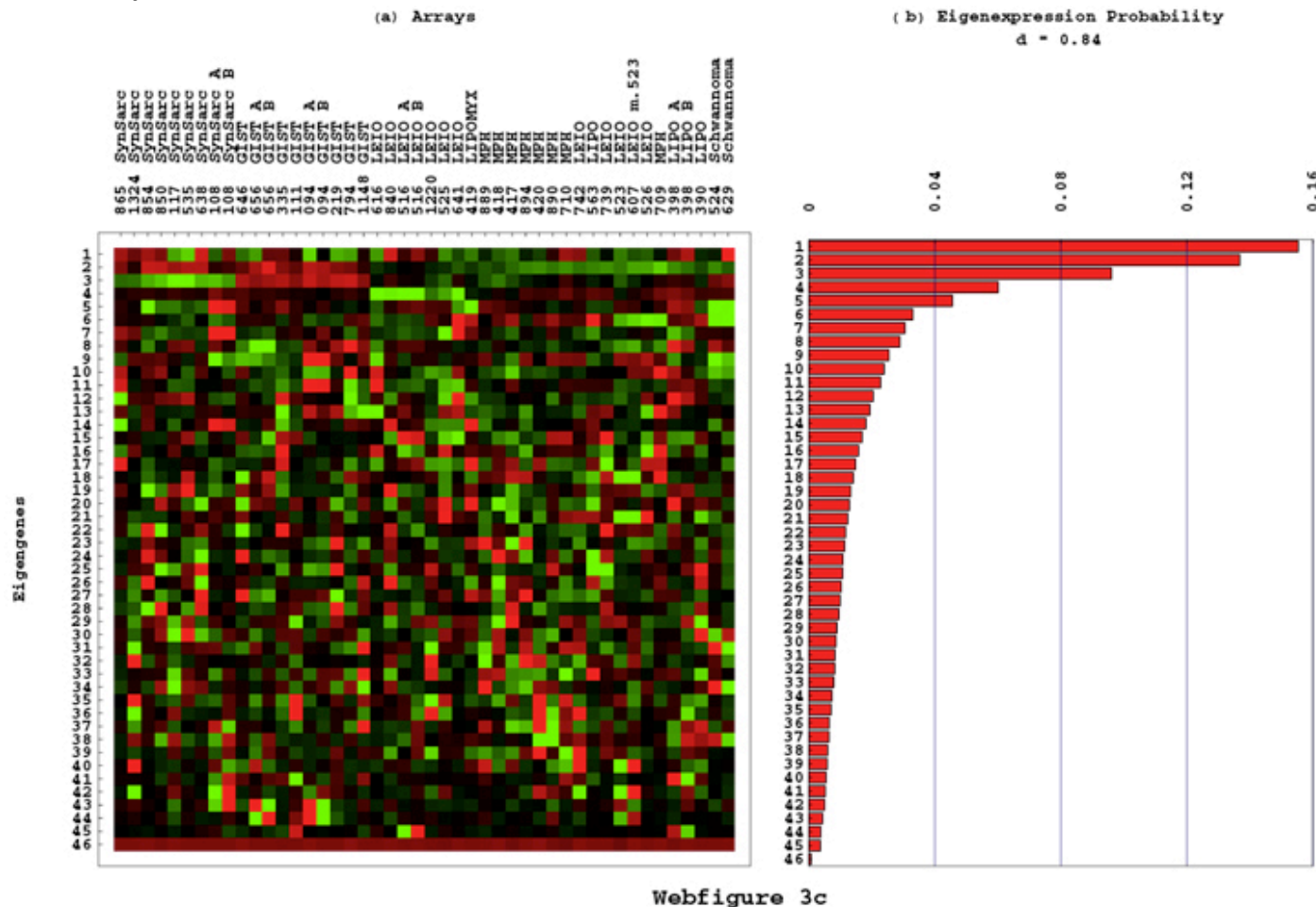
SVD in microarrays

Sample covariance matrix: $C = M M' / 45 = U S S' U' / 45$

Sample variance of i th component: $s_i^2 / 45$.

Total sample variance: $(s_1^2 + \dots + s_{46}^2) / 45$.

Here is V' and the explained fractions $s_i^2 / (s_1^2 + \dots + s_{46}^2)$
Nielsen et al., supplementary material.



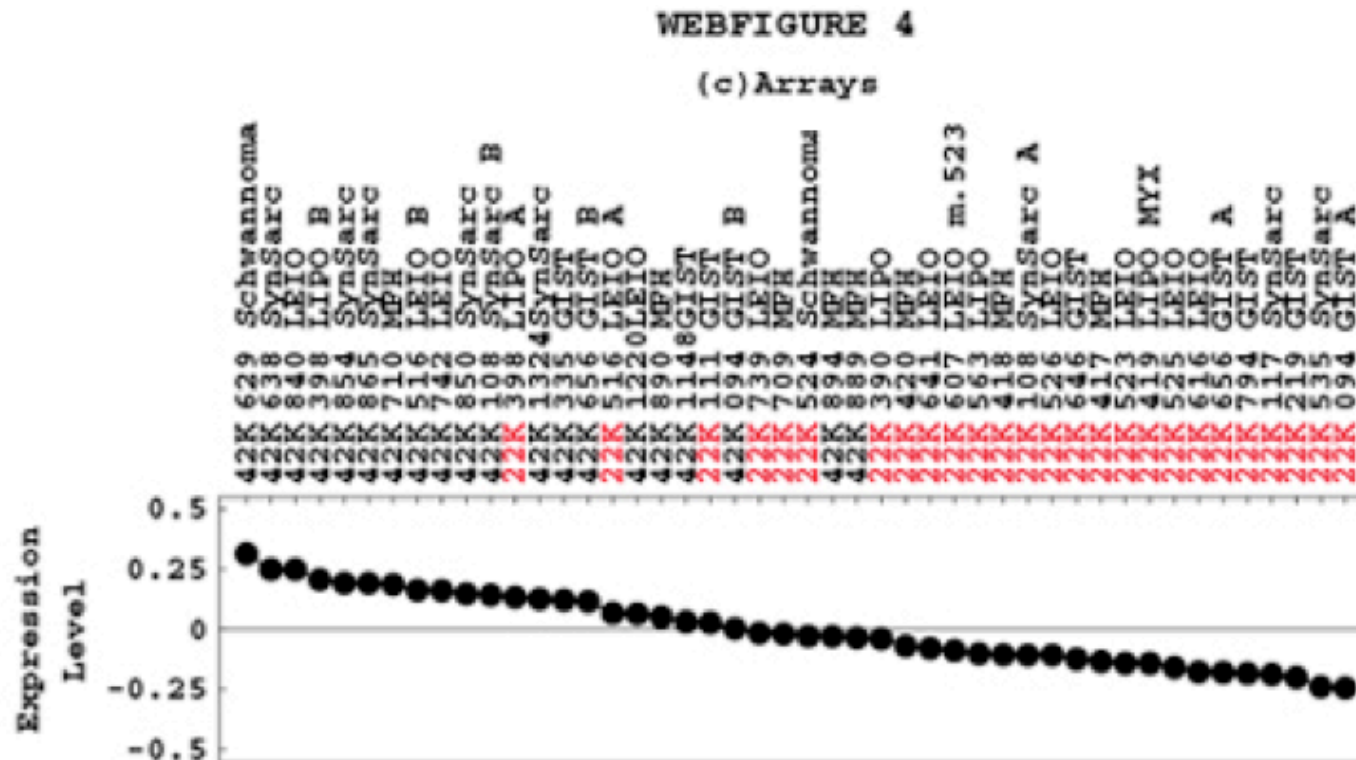
Webfigure 3c

“Expression level” of eigengenes

- The expression level of gene i on array j is M_{ij} .
- **Interpretation of change of basis $S = U'MV$:**
the i th eigenarray only detects the i th eigengene, and has 0 response to other eigengenes.
- **Interpretation of $V' = U'MS^{-1}$:**
The “expression level” of eigengene i on array j is $(V')_{ij} = V_{ji}$.
- Let \vec{m} represent a new array (e.g., a column vector of expression levels in each gene).
The expression level of eigengene i is $(U'\vec{m})_i/s_i$.

Platform bias

- They re-ordered the arrays according to the expression level V_{j1} in the first eigengene (largest eigenvalue).
- They found that V_{j1} tends to be larger in the 42K arrays and smaller in the 22K arrays. This is an experimental artifact, not a property of the specimens under study.



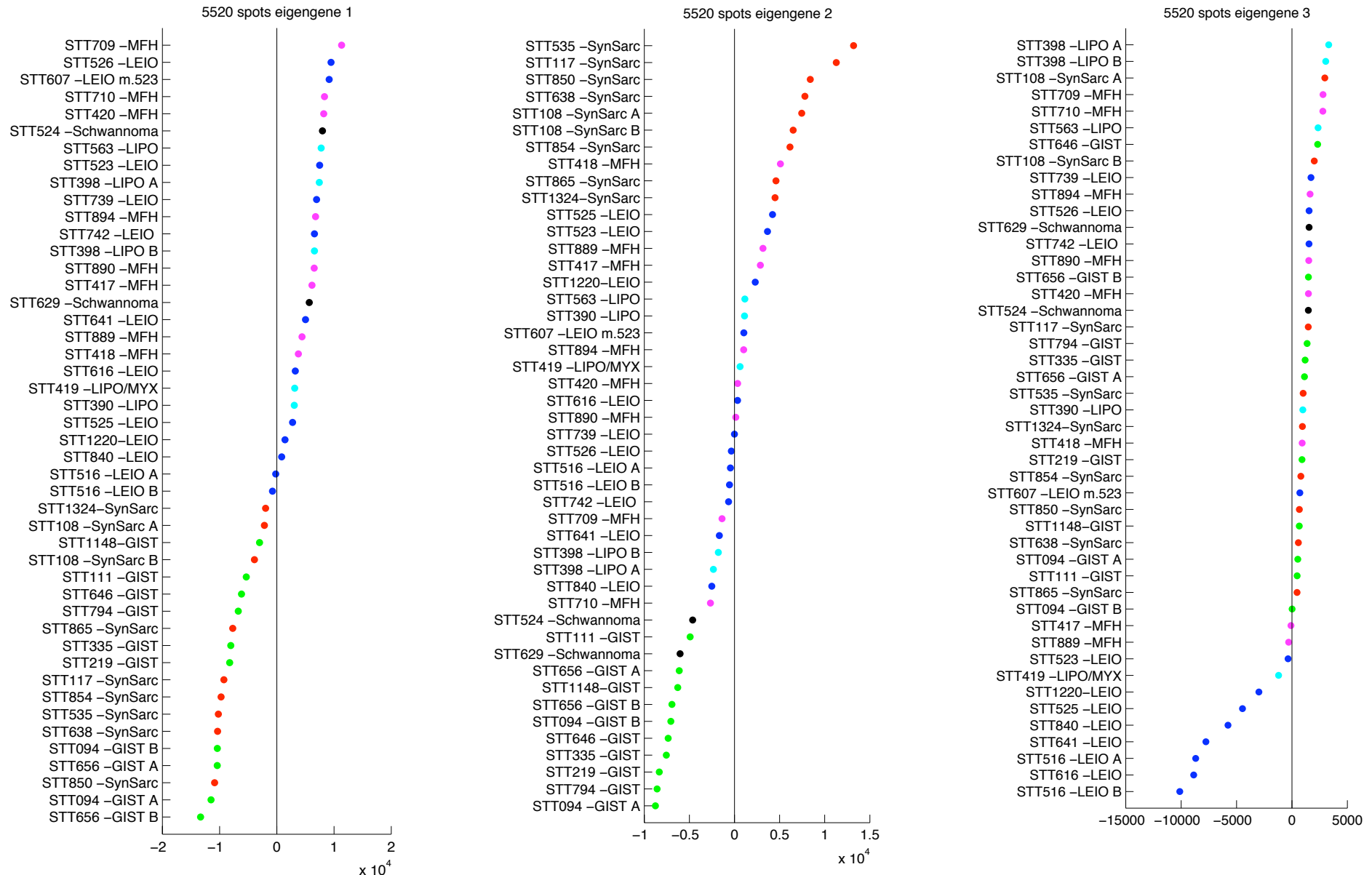
Nielsen et al., supplementary material.

Removing 22K vs. 42K array bias

- Let \tilde{S} be S with the $(1,1)$ entry replaced by 0.
- Let $\tilde{M} = U\tilde{S}V'$.
- This reduces the signal and variance in many spots.
After removing weak spots, they cut down to 5520 spots, giving a 5520×46 data matrix.

Classification — Eigengenes — 1D

For the 5520×46 matrix, the expression levels of the top three eigengenes can be used to classify some tumor types.



Classification — Eigengenes — 2D

$\lambda_1, \lambda_2, \lambda_3$ help distinguish between tumor types

