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

Prof. Tesler

Math 283
November 25, 2013

Covariance

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

$$\mathrm{Cov}(X,Y) = E\left((X-\mu_X)(Y-\mu_Y)\right)$$

and that an alternate formula is

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

- Previously we used

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

and

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

Covariance properties

Covariance properties

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

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

- ***When* $\mathbf{Cov}(X,Y)$ *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* $\mathbf{Cov}(X,Y)$ *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* $\mathbf{Cov}(X,Y) = \mathbf{0}$*:***
  a) X and Y **might** be independent, but it's not guaranteed.
  b) $\mathrm{Var}(X+Y) = \mathrm{Var}(X) + \mathrm{Var}(Y)$

Sample variance

Variance of a random variable:

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

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

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

Vector formula:

Centered data: $M = \begin{bmatrix} x_1-\bar{x} & x_2-\bar{x} & \cdots & x_n-\bar{x} \end{bmatrix}$

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

Sample covariance

Covariance between random variables X, Y:

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

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

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

Vector formula:

$$\begin{aligned} M_X &= \begin{bmatrix} x_1-\bar{x} & x_2-\bar{x} & \cdots & x_n-\bar{x} \end{bmatrix} \\ M_Y &= \begin{bmatrix} y_1-\bar{y} & y_2-\bar{y} & \cdots & y_n-\bar{y} \end{bmatrix} \end{aligned}$$

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

Covariance matrix

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

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

and then form a *covariance matrix:*

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

In matrix/vector notation,

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

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

Often there's one vector with all the variables:

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

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

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

Covariance matrix properties

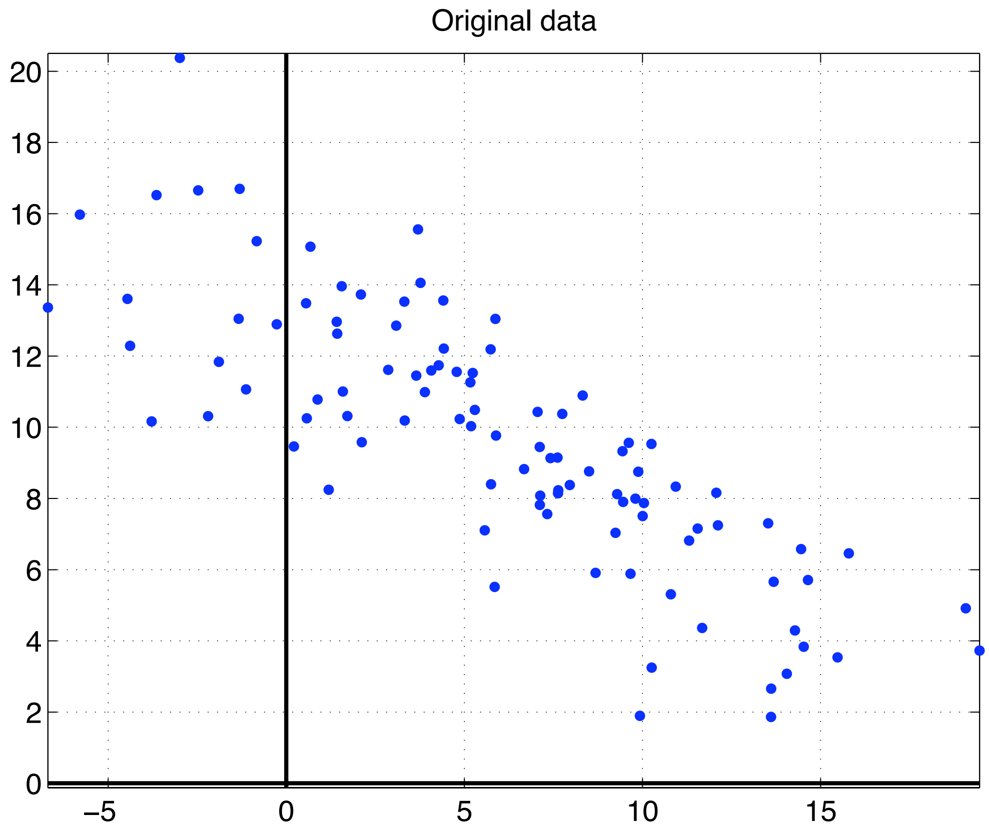
$$\begin{aligned}
\mathrm{Cov}(\vec{X},\vec{Y}) &= \mathrm{Cov}(\vec{Y},\vec{X})' \\[1ex]
\mathrm{Cov}(A\vec{X}+\vec{B},\vec{Y}) &= A\,\mathrm{Cov}(\vec{X},\vec{Y}) \\
\mathrm{Cov}(\vec{X},C\vec{Y}+\vec{D}) &= \mathrm{Cov}(\vec{X},\vec{Y})C' \\
\mathrm{Cov}(A\vec{X}+\vec{B}) &= A\,\mathrm{Cov}(\vec{X})A' \\[1ex]
\mathrm{Cov}(\vec{X}_1+\vec{X}_2,\vec{Y}) &= \mathrm{Cov}(\vec{X}_1,\vec{Y})+\mathrm{Cov}(\vec{X}_2,\vec{Y}) \\
\mathrm{Cov}(\vec{X},\vec{Y}_1+\vec{Y}_2) &= \mathrm{Cov}(\vec{X}_1,\vec{Y}_1)+\mathrm{Cov}(\vec{X}_2,\vec{Y}_2)
\end{aligned}$$

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

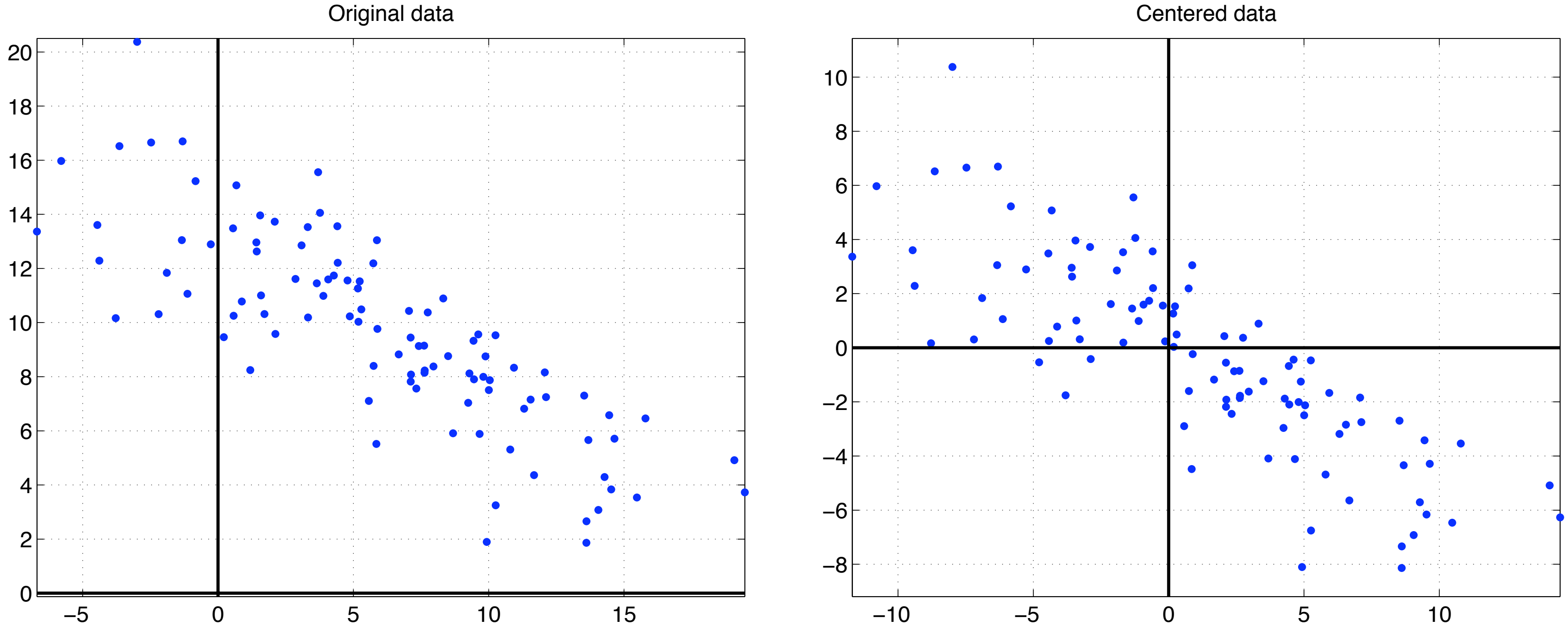
Example (2D, but works for higher dimensions too)

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

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



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:**

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

Diagonalizing the sample covariance matrix C

- C is a real-valued symmetric matrix, so it can be diagonalized $C = VDV'$ where $V' = V^{-1}$ (V transpose equals V inverse, meaning the columns of V are *orthonormal*):

$$\begin{array}{ccccc} C & = & V & D & V' \\ \begin{bmatrix} 31.9702 & -16.5683 \\ -16.5683 & 13.0018 \end{bmatrix} & = & \begin{bmatrix} -0.8651 & -0.5016 \\ 0.5016 & -0.8651 \end{bmatrix} & \begin{bmatrix} 41.5768 & 0 \\ 0 & 3.3952 \end{bmatrix} & \begin{bmatrix} -0.8651 & 0.5016 \\ -0.5016 & -0.8651 \end{bmatrix} \end{array}$$

- Recall *orthonormal* means the columns are unit vectors, and dotting any two of them together gives 0.
- It is conventional to put the eigenvalues into D in decreasing order $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant 0$.
- *Note:* C is positive semidefinite (all eigenvalues are $\geqslant 0$) because for all vectors $\vec{w}$,

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

For eigenvector equation $C\vec{w} = \lambda\vec{w}$, we have $\vec{w}\,'C\vec{w} = \lambda|\vec{w}|^2$.
So $\lambda|\vec{w}|^2 = \vec{w}\,'C\vec{w} \geqslant 0$, giving $\lambda \geqslant 0$.

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	$\begin{bmatrix}-0.5016\\-0.8651\end{bmatrix}$	$\begin{bmatrix}-0.9242\\-1.5940\end{bmatrix}$

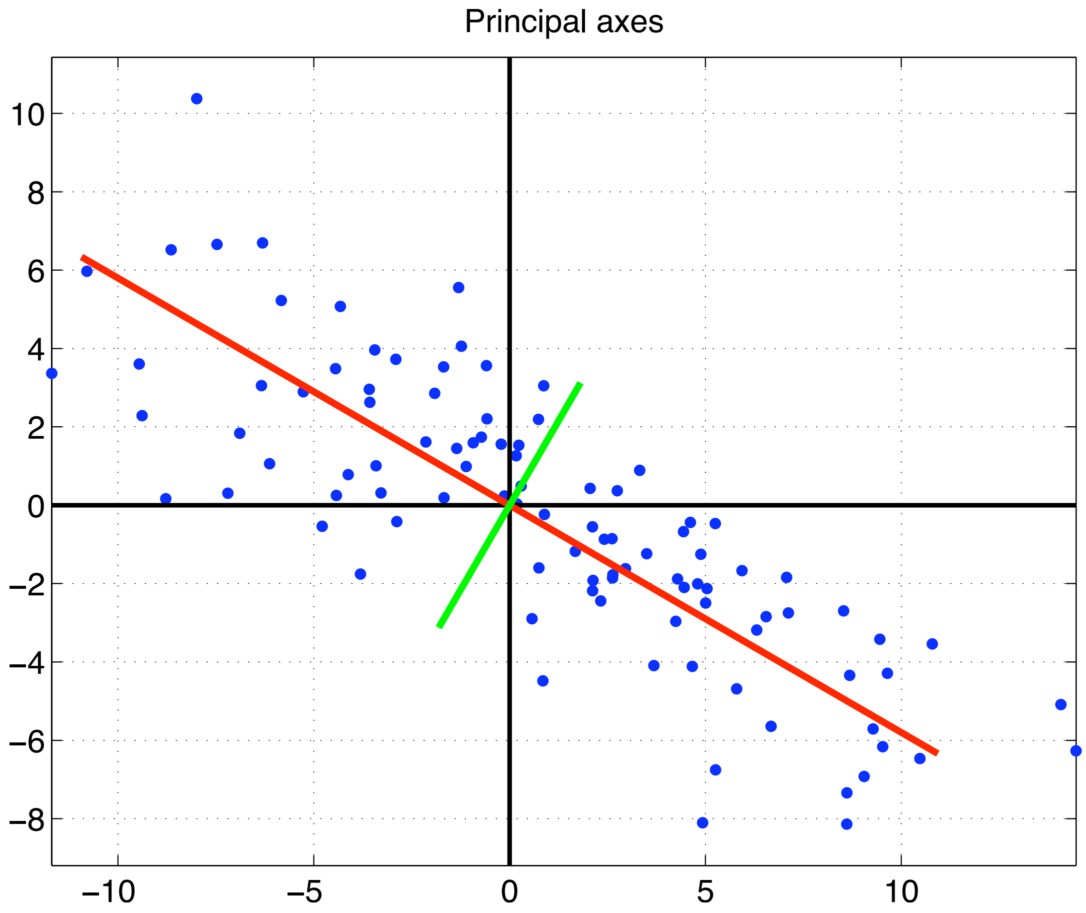
- Put them into the columns of a matrix:

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

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

Principal axes

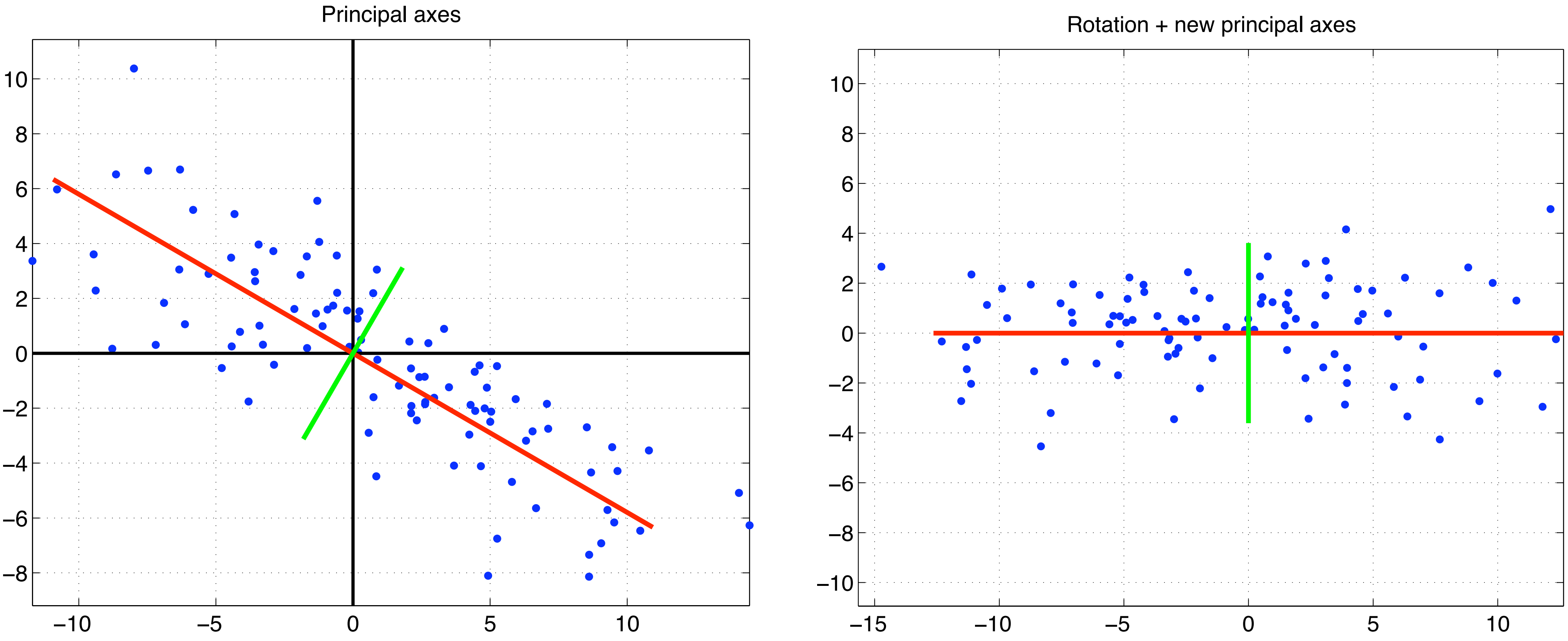
- Plot the centered data with lines along the principal axes:



- Sum of squared perpendicular distances of data points to first PC line (red) is minimum among all lines through origin.
- ith 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 the new columns:



- $C = VDV'$ so $V'CV = D$. (*Note*: V is orthonormal so $V' = V^{-1}$.)
- $C = \frac{MM'}{n-1}$ so $D = V'CV = \frac{V'MM'V}{n-1} = \frac{(V'M)(V'M)'}{n-1}$
- 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.

New coordinates

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

$$\begin{bmatrix} \mathrm{Var}(T) & \mathrm{Cov}(T,U) \\ \mathrm{Cov}(T,U) & \mathrm{Var}(U) \end{bmatrix} = \begin{bmatrix} 41.5768 & 0 \\ 0 & 3.3952 \end{bmatrix}$$

- The *total variance* is $\lambda_1 + \lambda_2 + \cdots = \mathrm{Tr}(D) = \mathrm{Tr}(C)$.
- Here, the total variance is $\mathrm{Var}(T) + \mathrm{Var}(U) = 44.9720$.
- 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%

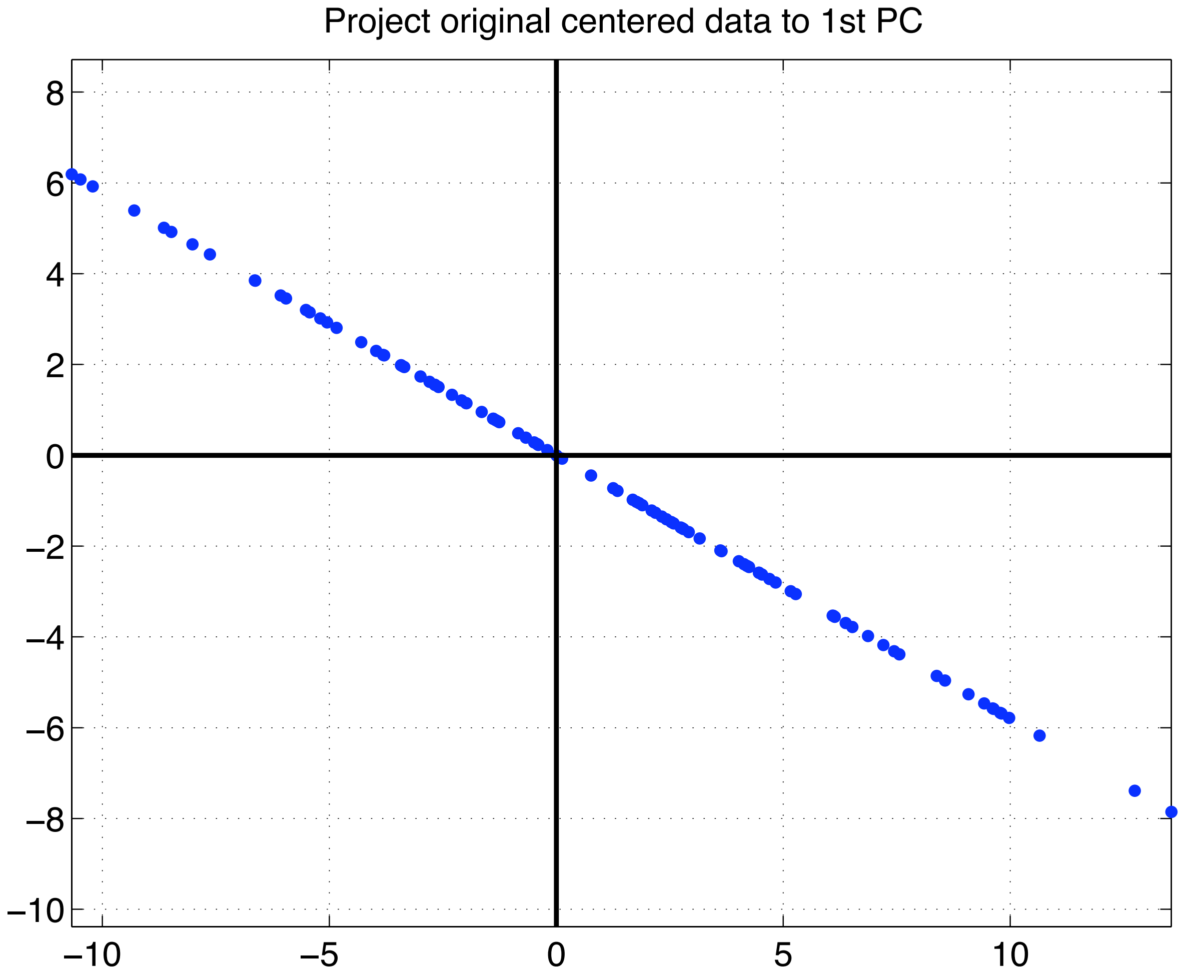
- This is an application of $\mathrm{Cov}(A\vec{X}) = A\,\mathrm{Cov}(\vec{X})A'$:

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

Dimension reduction

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

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



Dimension reduction

- Say we want to keep enough information to explain 90% of the variance.
- Take enough top PCs to explain $\geqslant 90\%$ 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 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. This is especially an issue with physical quantities with different units.
- It was originally designed for measurements in ordinary space, so, e.g., all axes would represent cm or inches and equivalent results would be obtained no matter what units were used.
- Length of (a,b) in (seconds,mm): $\sqrt{a^2+b^2}$.
  Convert to (hours,miles): $(a/3600, b/1609344)$ with length $\sqrt{(a/3600)^2+(b/1609344)^2}$.
  Angles are also distorted by this unit conversion.
- Length of $(0\ {}^\circ C, 0\ {}^\circ C)$ is 0,
  vs. length of $(32\ {}^\circ F, 32\ {}^\circ F)$ is 32.
  Both systems use an arbitrary zero offset instead of absolute zero.
- Typically addressed by replacing each row with Z-scores.

Microarrays

- Before we were interested in finding single genes where “red” or “green” (positive or negative expression level) distinguished between 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.

- We want to find linear combinations of genes that so that $L > C$ 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; we just want to see how to determine them when they are.

Microarrays

- Consider an experiment with 80 microarrays with 10000 spots on each.
- M is 10000×80.
- $C = \frac{MM'}{80-1}$ is $10000 \times 10000!$
- M has rank $\leqslant 80$ (actually $\leqslant 79$ since centering made the row sums $= 0$).
- C has the same rank as M.
  So at least $10000 - 80 = 9920$ of its eigenvalues are 0.
- It turns out the other 80 eigenvalues of MM' (10000×10000) are the same as in $M'M$ (80×80).

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, $s_1 \geqslant s_2 \geqslant \cdots \geqslant 0$.
- If M is 5×3, this would look like

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

“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, $s_1 \geqslant s_2 \geqslant \cdots \geqslant 0$.
- 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

- $M'M = (VS'U')(USV') = V(S'S)V' = V \begin{bmatrix} {s_1}^2 & 0 & 0 \\ 0 & {s_2}^2 & 0 \\ 0 & 0 & {s_3}^2 \end{bmatrix} V'$
- $MM' = (USV')(VS'U') = U(SS')U' = U \begin{bmatrix} {s_1}^2 & 0 & 0 & 0 & 0 \\ 0 & {s_2}^2 & 0 & 0 & 0 \\ 0 & 0 & {s_3}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} U'$
- **Compute the SVD using whichever gives smaller dimensions!**
- If $p \geqslant q$:
  - Diagonalize $M'M = VDV'$.
  - Compute S from D (S is a $p \times q$ matrix with square root of diagonal entries of D and 0's elsewhere).
  - Compute $U = MVS^{-1}$.
  - Set "$\frac{1}{0} = 0$" in diagonal of "S^{-1}" if a singular value is 0.
- The procedure when $q \geqslant p$ is analagous: diagonalize $MM' = UDU'$, then compute S from D, then compute V.
- `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

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

- Break U and V into columns

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

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

  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:

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

Computing PCA using SVD

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

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

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

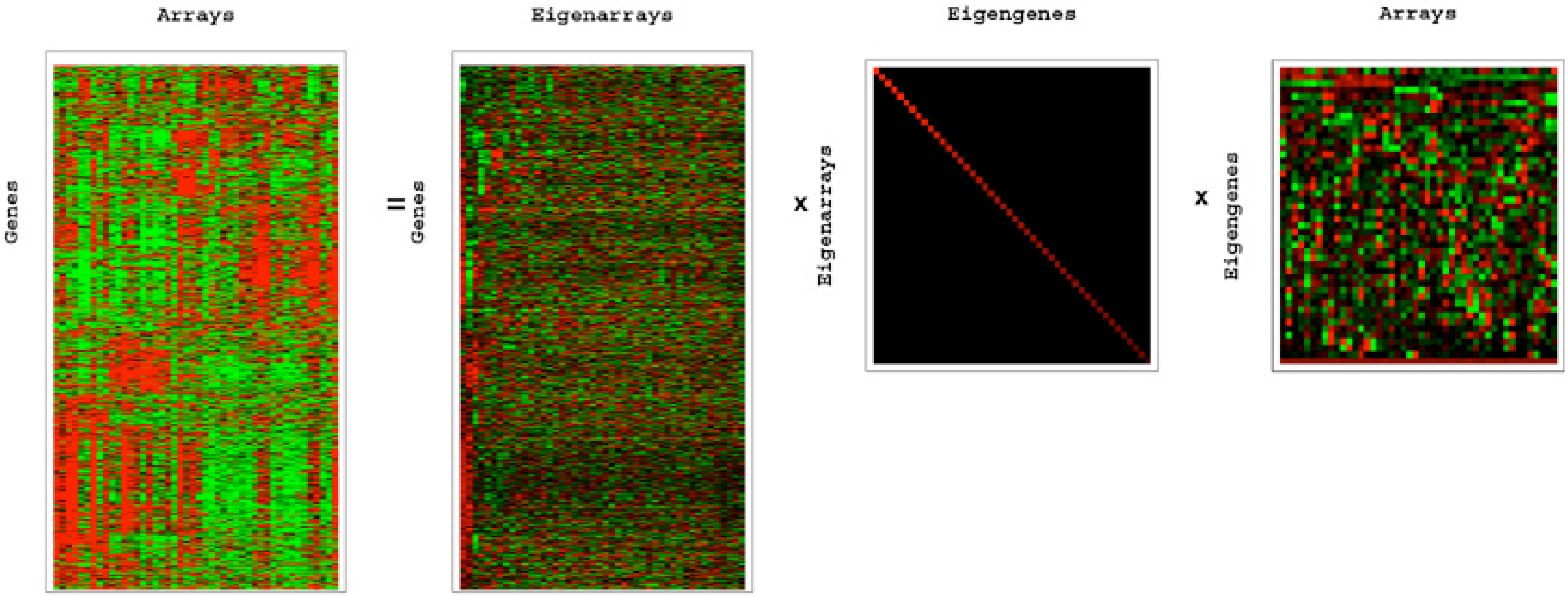
SVD in microarrays

- 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

- The compact form $M = USV'$ is shown below. They call the columns of U *"eigenarrays"* and the columns of V (rows of V') *"eigengenes."*
- Color scale: Negative 0 Positive



Webfigure 3a

Nielsen et al., supplementary material.
`http://genome-www.stanford.edu/sarcoma/Supplemental_data.shtml`

SVD in microarrays

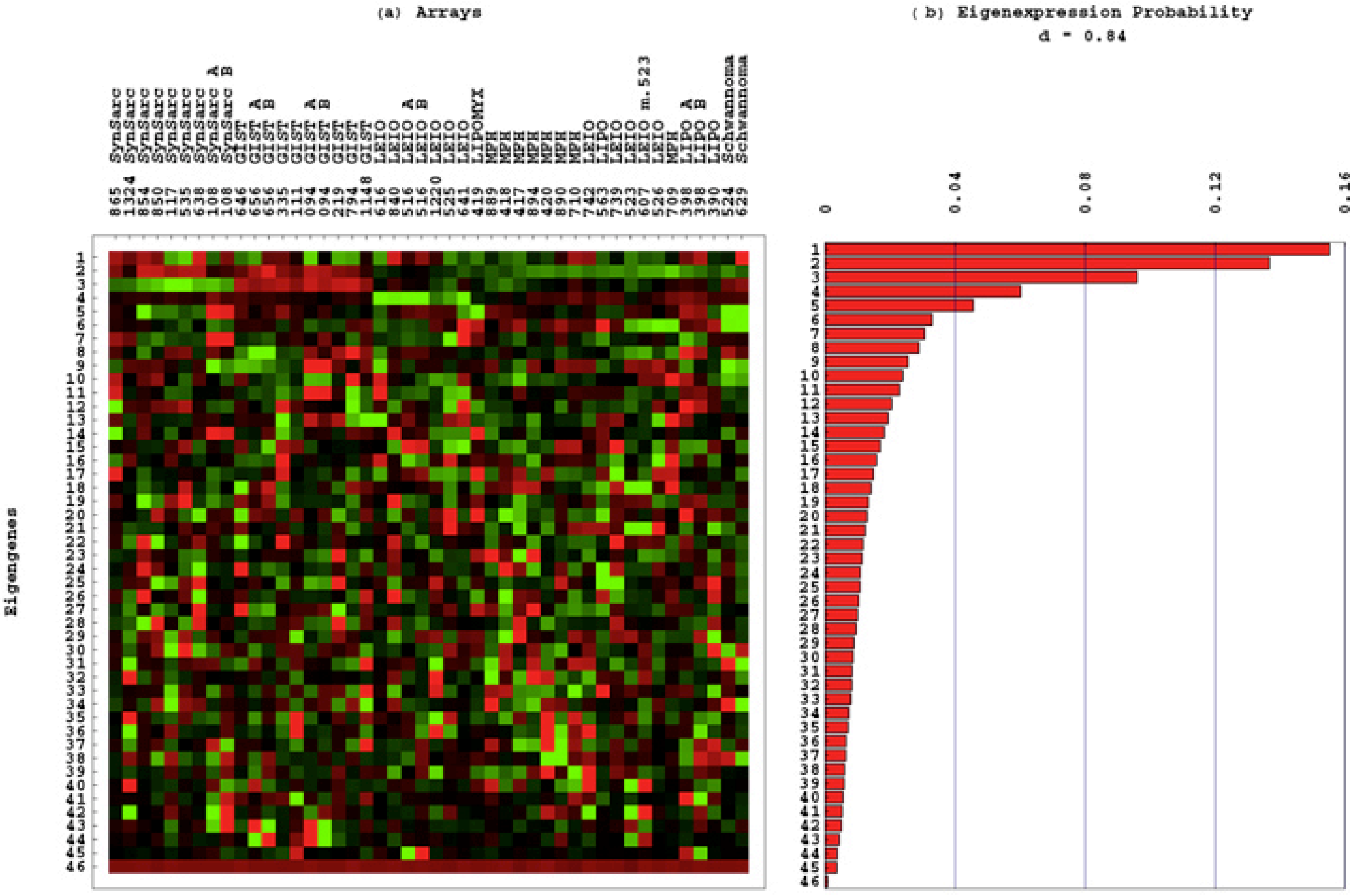
Sample covariance matrix: $C = M\,M'/45 = U\,S\,S'\,U'/45$

Sample variance of ith component: ${s_i}^2/45$.

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

Here is V' and the explained fractions ${s_i}^2/({s_1}^2 + \cdots + {s_{46}}^2)$

Nielsen et al., supplementary material.



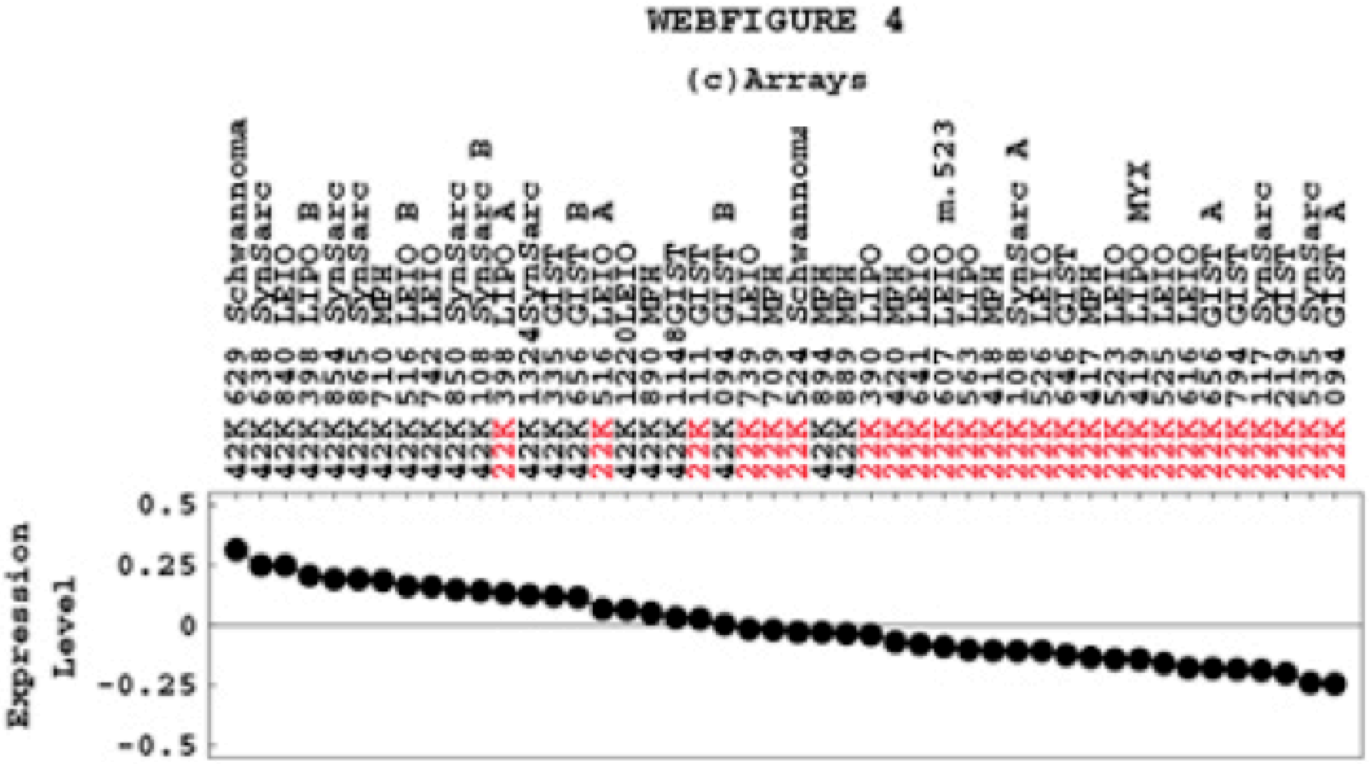
Webfigure 3c

“Expression level” of eigengenes

- The expression level of gene i on array j is M_{ij}.
- **Interpretation of change of basis $\boldsymbol{S = U'MV}$:** the ith eigenarray only detects the ith eigengene, and has 0 response to other eigengenes.
- **Interpretation of $\boldsymbol{V' = U'MS^{-1}}$:** 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).
- The 42K arrays tend to have positive expression level and the 22K arrays tend to have negative expression level.



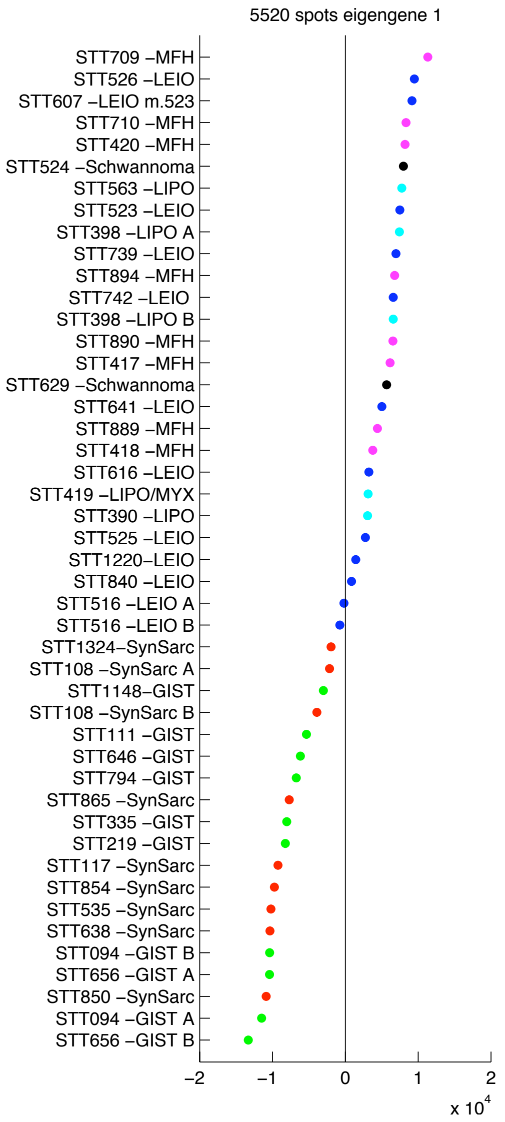
Nielsen et al., supplementary material.

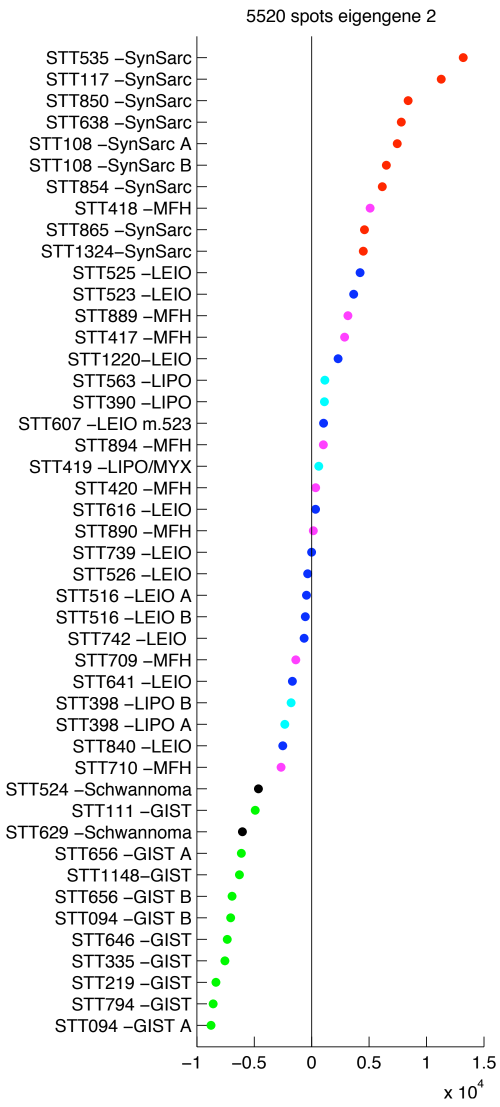
Removing 22K vs. 42K array bias

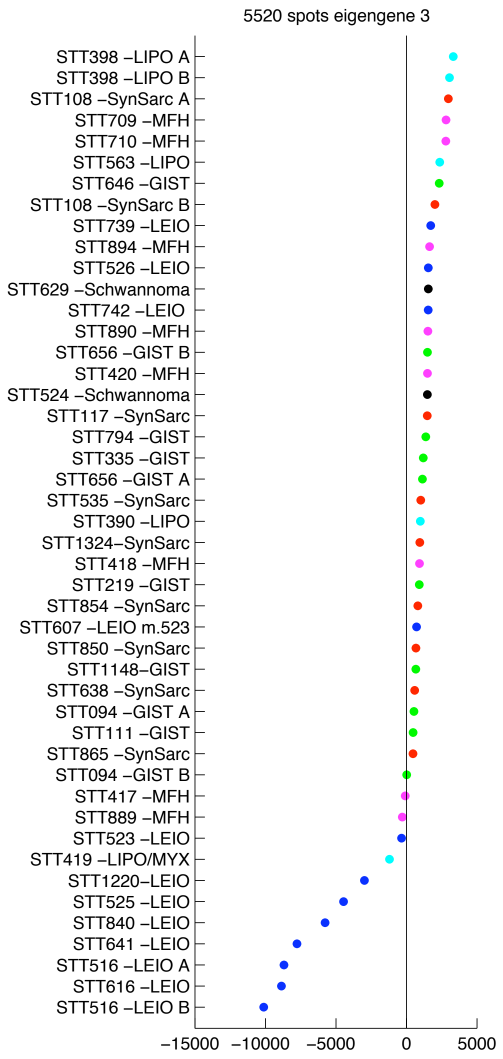
- Let $\widetilde{S}$ be S with the (1,1) entry replaced by 0.
- Let $\widetilde{M} = U\widetilde{S}V'$.
- This reduces the signal and variance in many spots. After removing weak spots, they cut down to 5520 spots, giving a 5520×46 data matrix.

Classification — Eigengenes — 1D

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

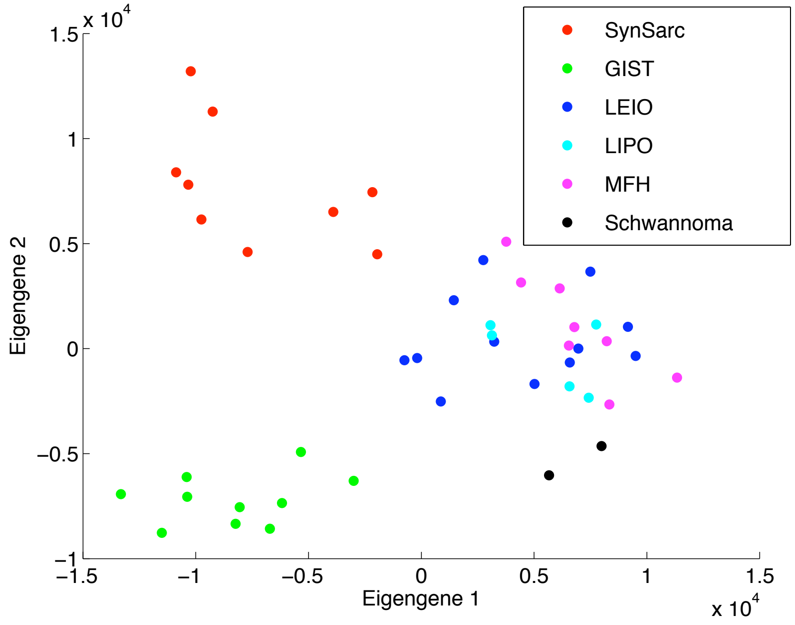


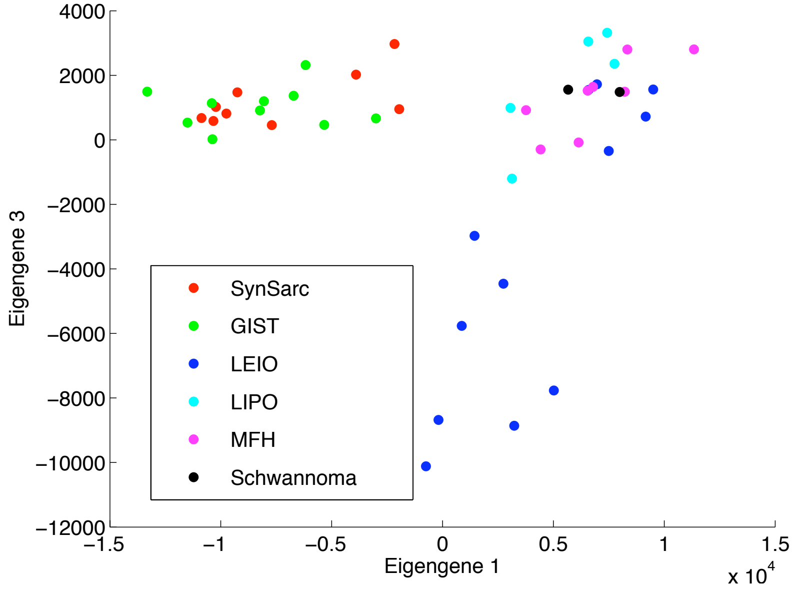


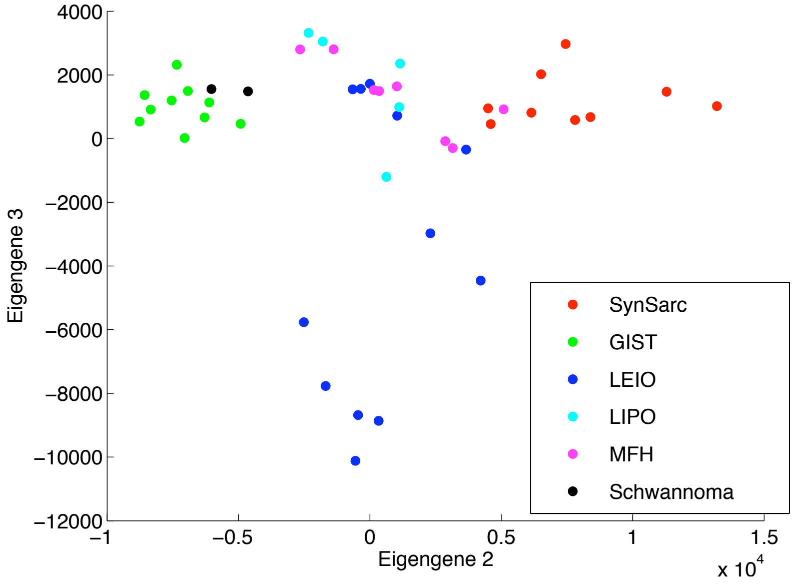


Classification — Eigengenes — 2D

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


x 10^4
Eigengene 2
Eigengene 1
x 10^4
● SynSarc
● GIST
● LEIO
● LIPO
● MFH
● Schwannoma


Eigengene 3
Eigengene 1
x 10^4
● SynSarc
● GIST
● LEIO
● LIPO
● MFH
● Schwannoma


Eigengene 3
Eigengene 2
x 10^4
● SynSarc
● GIST
● LEIO
● LIPO
● MFH
● Schwannoma