Statistical Methods for High Dimensional Biology

STAT/BIOF/GSAT 540

Lecture 12 – Unsupervised learning: clustering and PCA

Sara Mostafavi

Feb 13, 2019

Slide credits: Dr. Paul Pavlidis

Unsupervised learning

- “Automated” way of finding patterns/structure in the data – without specifying what those patterns should tell you about! The distinction with supervised learning become more clear in future lectures.

- Two general frameworks:

1. Dimensionality reduction (e.g., PCA)
2. Clustering (next class)

Unsupervised learning

- "Automated" way of finding patterns/structure in the data – without specifying what those patterns should tell you about! The distinction with supervised learning become more clear in future lectures.

- Two general frameworks:

1. Dimensionality reduction (e.g., PCA)
   - Visualization, compression, hypothesis generation
2. Clustering (next class)

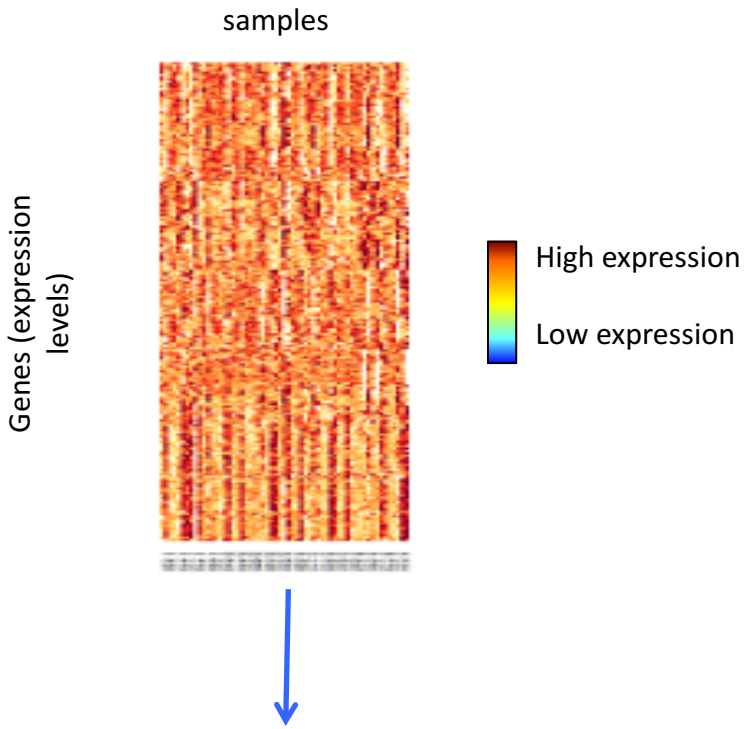
Today

- PCA
  - Covariance matrices
  - Basis span and matrix rank
  - SVD
  - Applications

“Don’t try to do surgery before knowing how to use a band-Aid”. L . Wasserman (paraphrased)

New notation and data representation

- From now on, we are going to represent all of our *measured* data in a matrix called X.

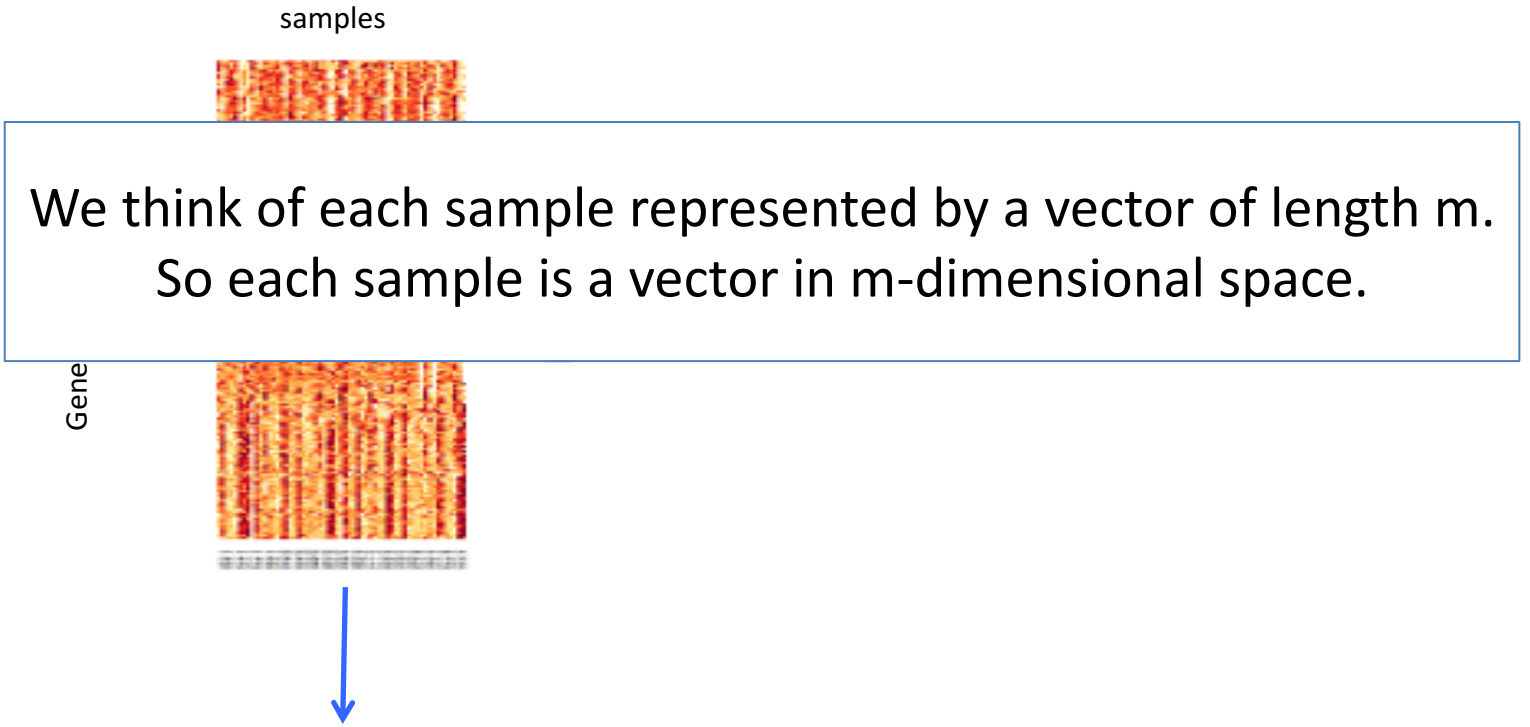


Matrix X has m rows (number of genes) and n columns (number of samples)

- X has no information about experimental design (i.e., meta-data is not part of X)
- Hence “unsupervised analysis”

New notation and data representation

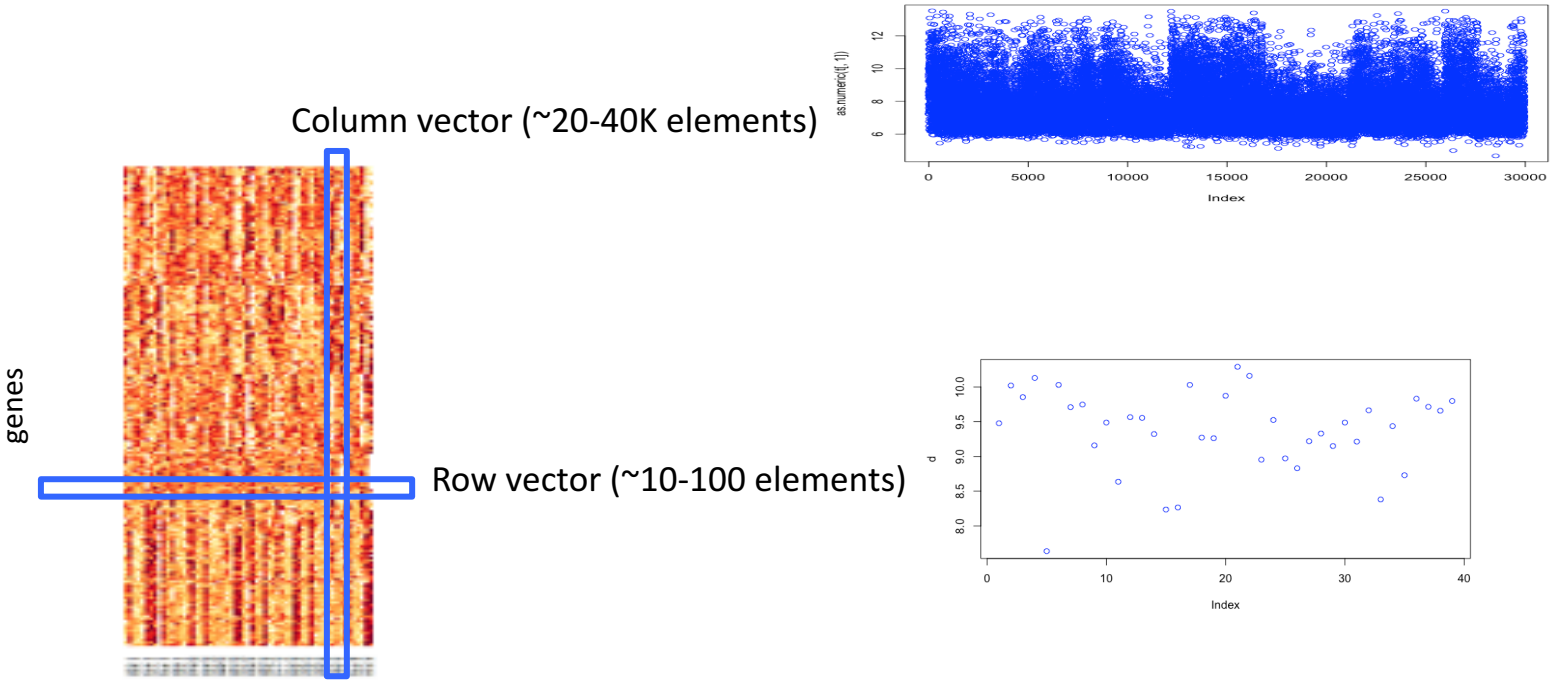
- From now on, we are going to represent all of our *measured* data in a matrix called X.



Matrix X has m rows (number of genes) and n columns (number of samples)

- X has no information about experimental design (i.e., meta-data is not part of X)
- Hence “unsupervised analysis”

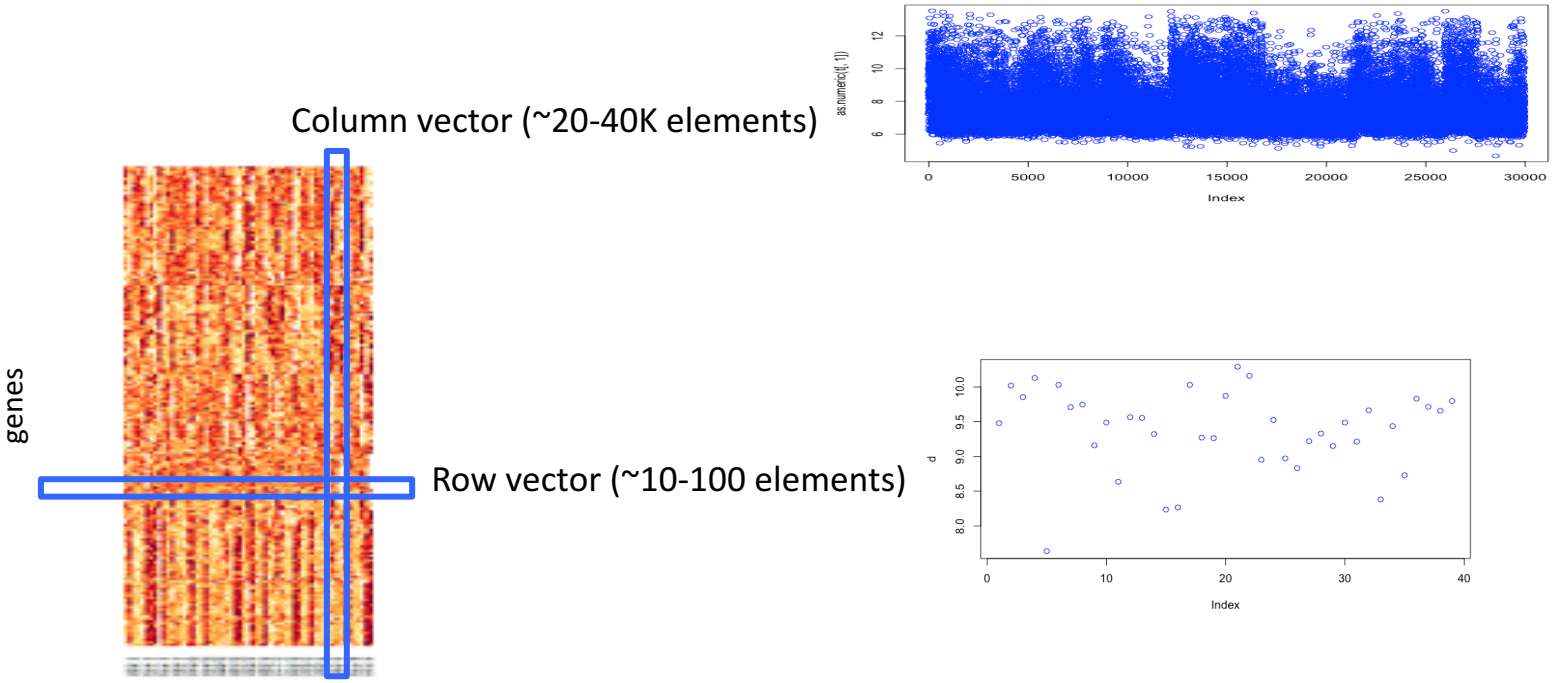
Finding structure in high-dimensional data



To get a sense of the data, we can look at “part of it”:

- The expression level of one gene across *all samples*
- The expression levels of *all genes* for one sample

Finding structure in high-dimensional data

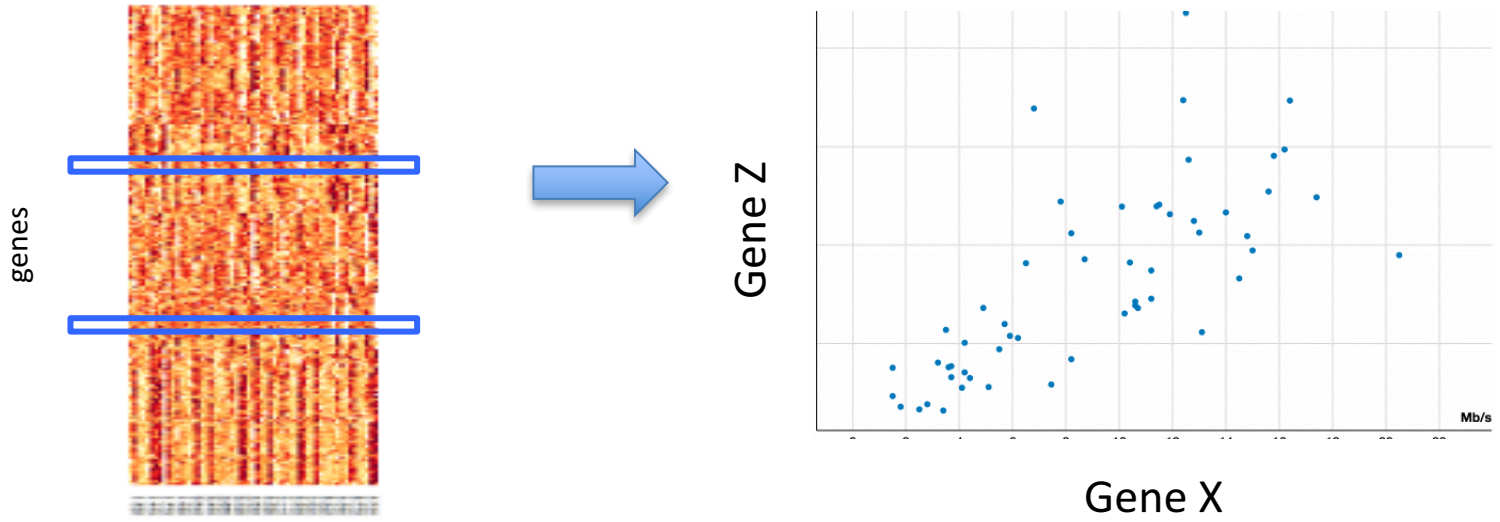


To get a sense of the data, we can look at "part of it":

- The expression level of one gene across *all samples*
- The expression levels of *all genes* for one sample

These correspond to 1-d representation of "data points"

You can do a little better than 1-d representation….



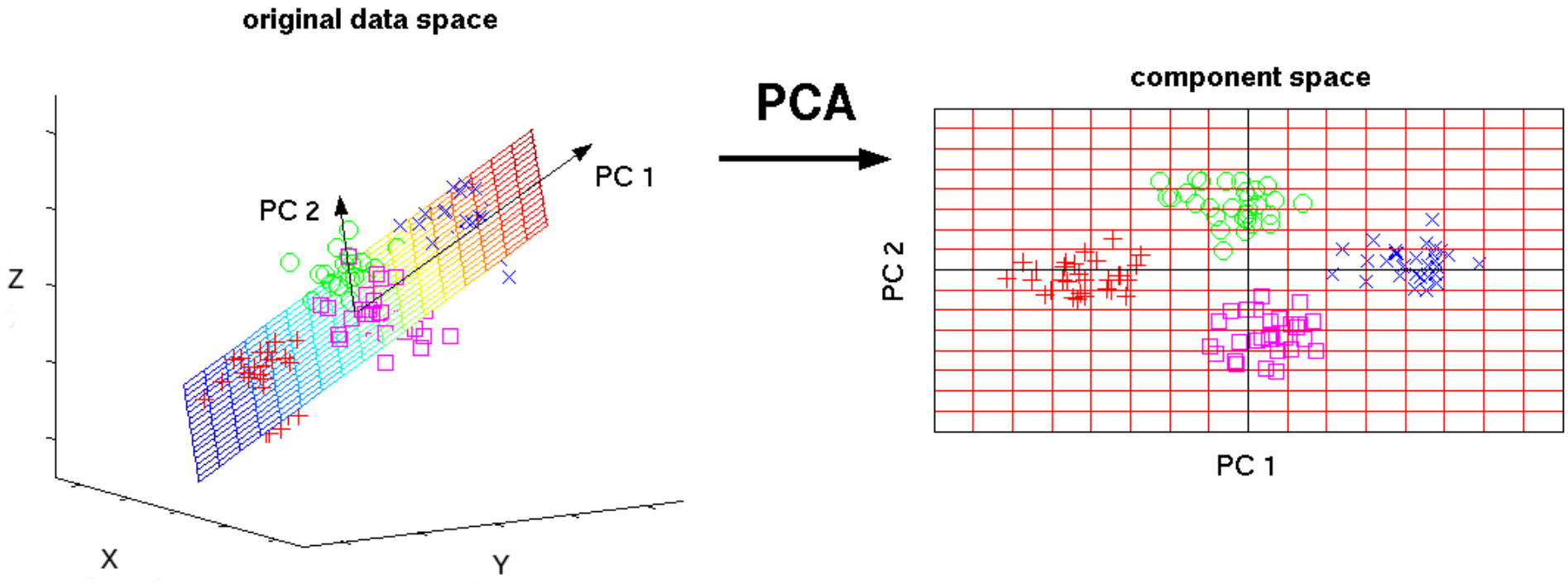
We can represent each sample in a 2-d space

Principal Component Analysis (PCA)

- PCA is a widely used example of dimensionality reduction that allows us to project data from higher dimensional space to a lower dimensional space.

- To project data, PCA **decomposes** data into **components**:
  - “Compact” representation
  - Components help with visualization and data interpretation.

Principal Component Analysis (PCA)



[google images]

Definitions of PCA

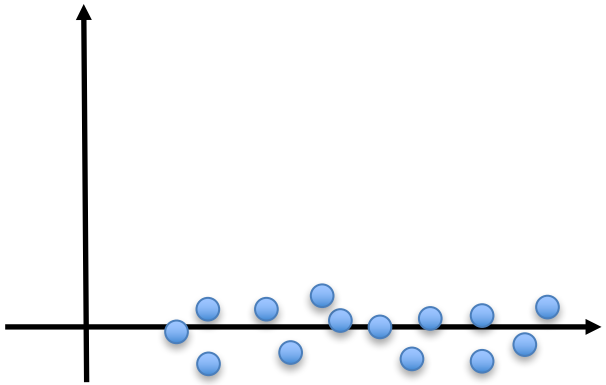
- Two common definition that give rise to the same algorithm:
  - Orthogonal projection of data into a lower dimensional space, known as *principal sub-space*, such that the total variation of the projected data is maximized (*Hotelling 1933*)
  - Projection that minimizes the projection cost, defined as the mean squared distance between projected data and original data (*Pearson 1901*)

[Bishop 2006]

Maximum variance formulation

- Consider data m-dimensional vectors $\overrightarrow{x_i}$, representing the expression levels of all genes for each sample
- We want to construct “proxy” (projected) vectors $\overrightarrow{y_i}$ that are k-dimensional, where k<<m
- Then we can visualize the data in k-dimensional space.

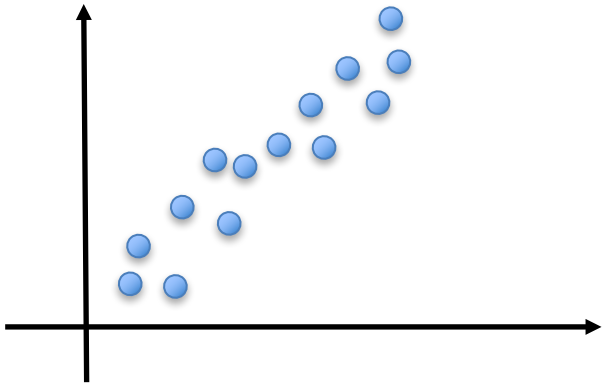
Intuition for sub-space representation



A projection on the x-axis will not ”lose” too much information

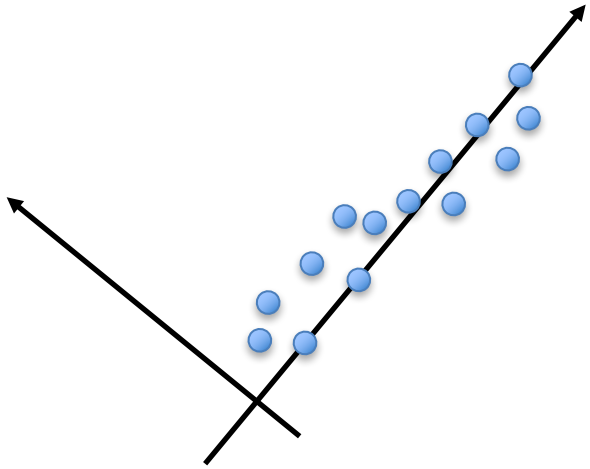
i.e., only one of the two dimensions seems useful

Intuition for sub-space representation



A projection on the x-axis will lose more information

But we could ”rotate” the data first or equivalently rotate the axis



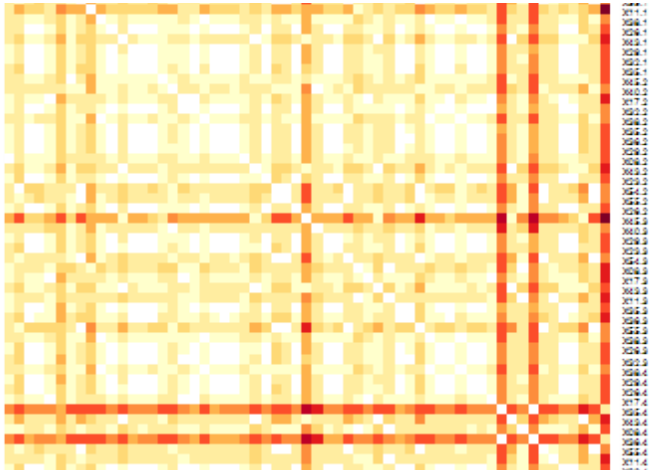
Review: sample covariance matrix

K_{ij} = covariance between data point i and data point j
K_{ii} = Variance of data point i

$$\mathrm{Cov}(\mathbf{x}, \mathbf{y}) = \frac{\sum_i^N (x_i - \bar{x})(y_i - \bar{y})}{N - 1}$$

Are you "far" from your mean when I am far from my mean?

K



- X^TX on centered data gives you the covariance matrix
- **Row vs column** covariance matrix
- Correlation matrix: X^TX when data is standardized across columns

Review: dot product & covariance

$$\vec{x}.\vec{y} = \sum_{i=1}^{n} x_i y_i = \cos\theta \,||\, x \,||||\, y \,||$$

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

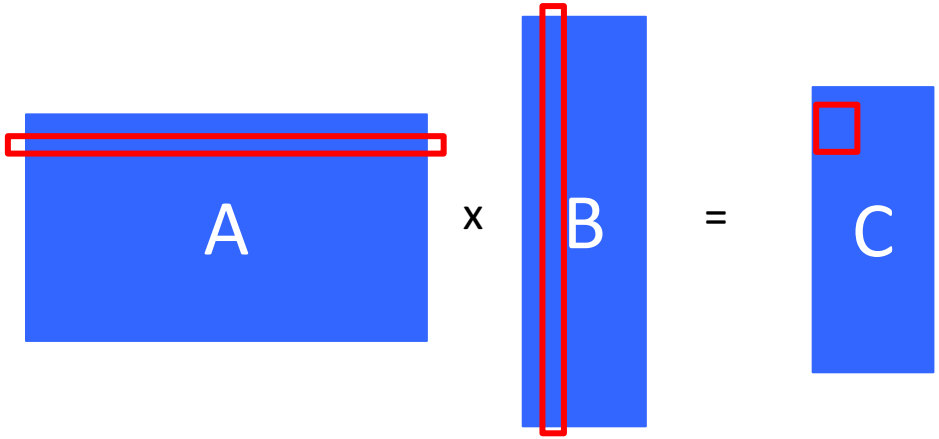
$$\text{cov}(\vec{x}, \vec{y}) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2 (y_i - \bar{y})^2}{n-1}$$

$$\text{Corr}\ (\vec{x}, \vec{y}) = \frac{cov(\vec{x},\vec{y})}{\sqrt{var(\vec{x})\ var(\vec{y})}}$$

Review: Matrix multiplication

A systematic way of computing dot products between rows of A and columns of B

A x B

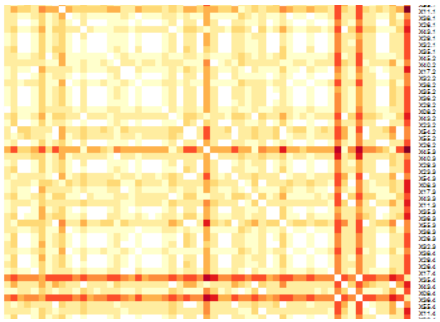


- $C_{2,3}$ is the dot product between row 2 of A and column 3 of B
- Dot product between rows of A and columns of B
- In R: %*% (?matmult)

Column covariance matrix

- Going back to our gene expression matrix X.
- If we **center** the matrix X: subtract mean from each column.
- $X^TX = K$ which is the sample (column) covariance matrix
- K_{ij} => dot product between column i and column j

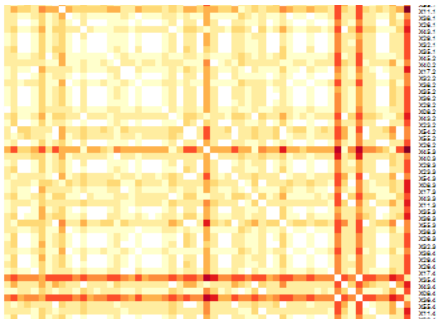
K



Column covariance matrix

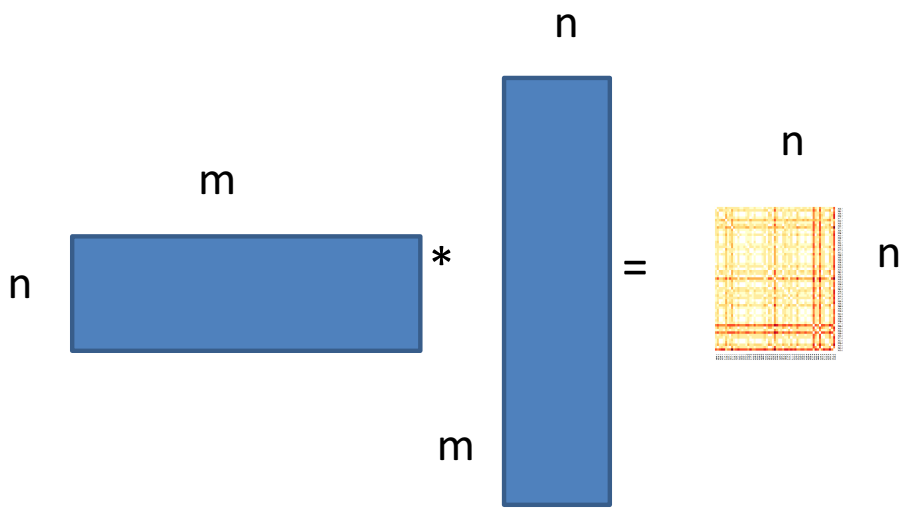
- Going back to our gene expression matrix X.
- **Correlation matrix can be computed as X^TX if we first standardize the columns of matrix X**
- $X^TX = K$ which is the sample (column) covariance matrix
- K_{ij} => dot product between column i and column j

K



Three thought experiments

- Think about the correlation structure of the data X_{mxn}, and understand that they can be described by at least *m independent* vector (where m<n)



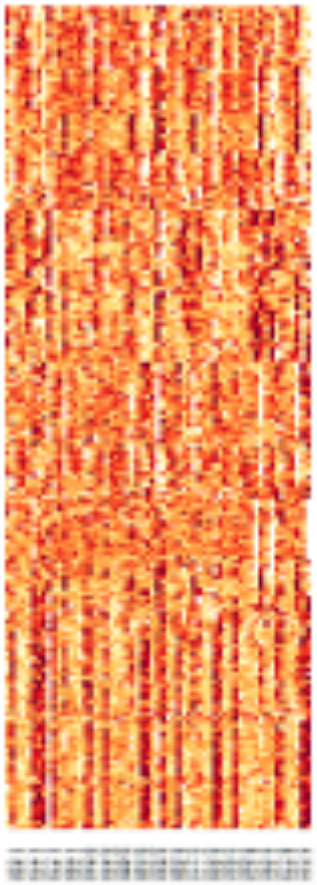
What does the column data look like?

X has m rows and n columns

1. All samples are perfectly correlated?

2. No correlation among samples? (they are perfectly uncorrelated)

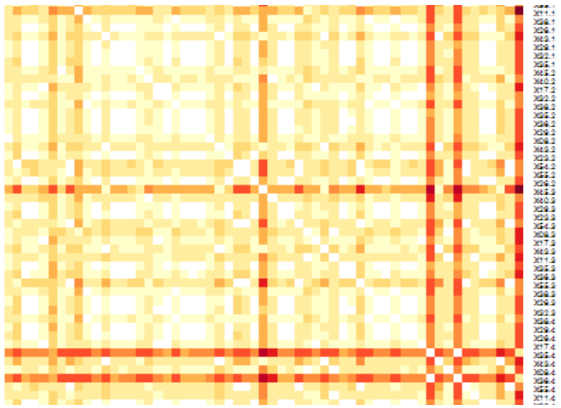
3. There are two types of samples, which are perfectly correlated within group and completely uncorrelated outside group?



What happens to the column correlation matrix?

X has m rows and n columns

1. All samples are perfectly correlated?
2. No correlation among samples? (they are prefect uncorrelated)
3. There are two types of samples, which are perfectly correlated within group and completely uncorrelated outside group?



Discussion of the second case

- If all samples are perfectly uncorrelated, they are vectors all at right angle from each other (orthogonal).

- While the column vectors are of length m, the m orthogonal vectors only *span* a subspace of R^m

Definition: Basis and Span

- A basis for R^m (m-dimensional space) is (roughly) a set of *m* vectors from which you can "make" any vector in R^m

- An orthogonal basis is a nice kind of basis: the *m* vectors are orthogonal.
  - E.g., [0,2] and [2,0]
  - **Any pair of orthogonal 2-vectors is a basis in R^2**

- Even nice is an **orthonormal basis**, where the vectors and normalized to length 1:
  - E.g., [0,1] and [1,0]

Example: 2-d basis and a 1-d subspace

- x=[1,0] and y=[0,1] are examples of orthonormal basis
- Any vector *v* in R^2 can be constructed by a weighted combination of these two vectors:

$$v = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- Sub-space: the "x-axis" while two dimensional only spans a line.

$$v = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

- **You need at least m vectors to span ("fill") an m-dimensional space.**

- The situation for the columns of the X matrix: they live in at most an n-dimensional space (where n<<m)

Vector span and matrix rank

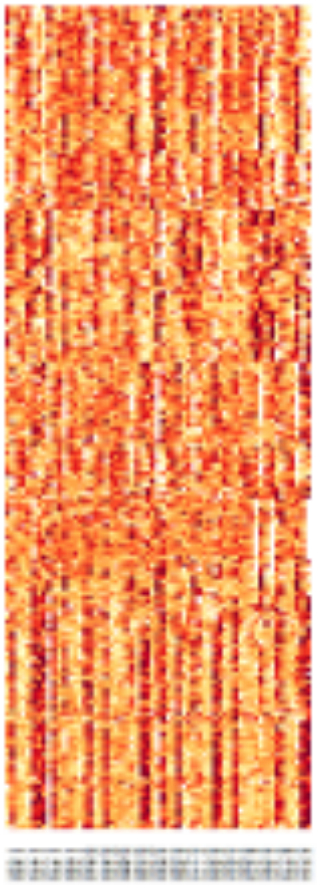
- You need at least *m* vectors to span an m-dimensional space (R^m)

- With data matrices $\mathbf{X}_{\mathbf{nxm}}$ that are "rectangular" the rank (max number of linearly independent column) is min(n,m)

What does the **row data** look like?

X has m rows and n columns

1. All samples are perfectly correlated?
2. No correlation among samples?
3. There are two types of samples, which are perfectly correlated within group and completely uncorrelated outside group?



Discussion of second case for **row data**

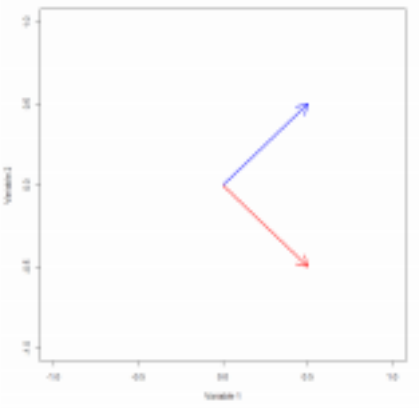
- The rows are more obviously “n-dimensional” – they each contain n values!

- Each row is a linear combination of at most *n* orthogonal vectors.

- This guarantees that there will be some correlated row (gene) vectors. (bc there are m>>n rows)

2-d example

- In 2-d, each point has an “X” and “Y” coordinate.
- If we have more than two vectors in 2-d, they can’t all be at a right angle to each other. (some of them will be correlated)
- This is the same situation for thousands of rows in X.



Summary so far…

- We can think of both genes and samples as combination of no more than *n* vectors that form an orthonormal basis.

- In PCA, we want to find such orthonormal basis that preserve the variance in the data.
  - These are called *row and column* **eigenvectors**

Eigenvectors

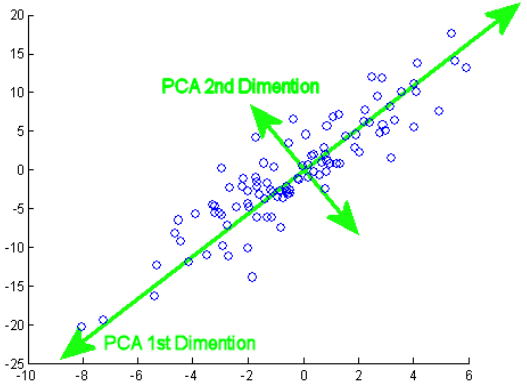
(only defined for square matrices)

- A special vector that is only stretched by linear transformations defined by the matrix; not rotated.

- The relative amount of stretching is the eigenvalues.

Eigenvectors and PCA

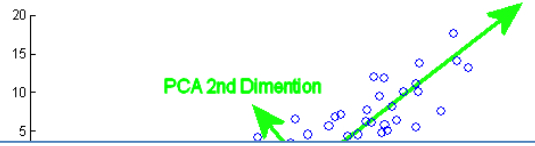
- We can compute the eigenvector of the covariance matrix (as it's a square matrix)
- These define the **principal components (PC)**
- Give us the direction of maximum variance – projections on these vectors has max. variance



- Intuition: covariance matrix has information about variance.

Eigenvectors and PCA

- We can compute the eigenvector of the covariance matrix (as it's a square matrix)
- These define the **principal components (PC)**
- Give us the direction of maximum variance – projections on these vectors has max. variance



Eigenvalue associated with each eigenvector *can* tells us about the proportion of variance in original data explained by projecting in each PC.

- Intuition: covariance matrix has information about variance.

SVD

- In practice for non square matrices we use SVD.

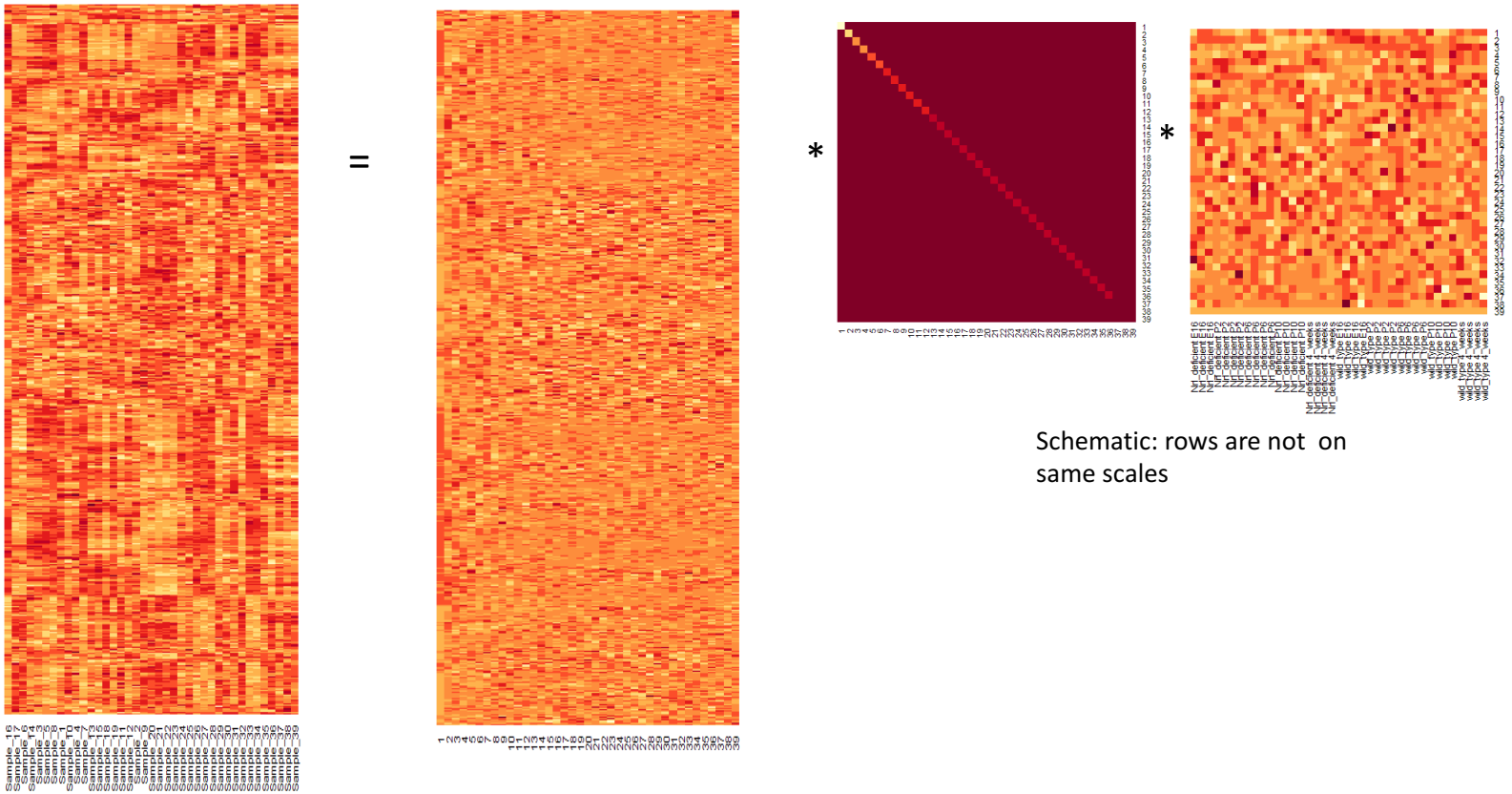
- $X_{mxn}=UDV^T$

- U_{mxk} left singular vectors – orthonormal basis of row space of X
- V_{mxk} right singular vectors – orthonormal basis of column space of X
- D diagonal matrix of eigenvalues

Original data

Decomposition

X = **U** * **D** * **V^T**



Schematic: rows are not on same scales

- The columns of U are “eigensamples” : samples are mixtures of them
- The columns of V are “eigengenes” (rows of V’): genes are mixtures of them
- D tells us how much of each eigensample or eigengene needs to be mixed together to reconstruct each gene or sample vector.

Linear algebra manipulations…

Each column of X is constructed as:

$$\vec{x}_{:,i} = UD\vec{v}_{:,i} = \sum_{j=1}^{n} U_{:,j} d_{jj} v_{j,i}$$

i.e., each column of X is a sum of weighted U column vectors.

Linear algebra manipulations…

Each column of X is constructed as:

$$\vec{x}_{:,i} = UD\vec{v}_{:,i} = \sum_{j=1}^{n} U_{:,j} d_{jj} v_{j,i}$$

i.e., each column of X is a sum of weighted U column vectors.

Matrix X reconstructed from outer product of rank-1 vectors

$$X = \sum_{j=1}^{n} d_{jj} U_{:,j} V_{j,:}$$

d_{jj} is the variance of X that is explained by the j^{th} left and right eigenvectors.

Application on photoreceptor data

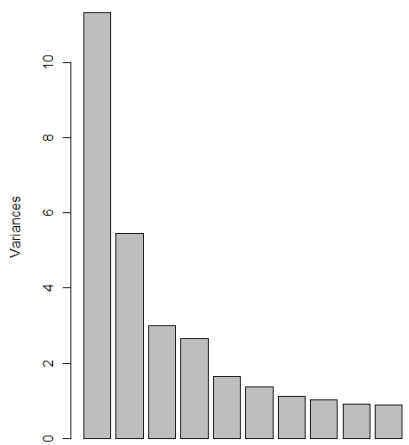
1. Data represented as sum of components
2. Projection of samples (principal components, columns of V^T)
3. Projection of genes (factor loadings, rows of U)

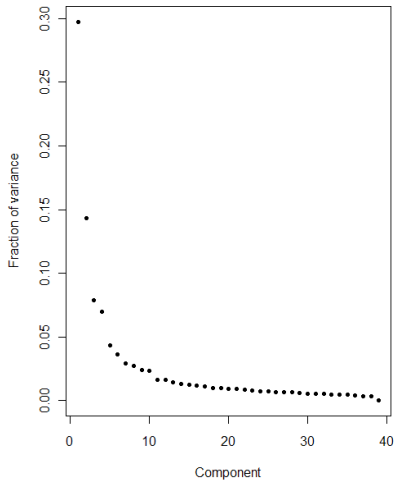
Practical stuff

- Before applying SVD/PCA standardize your data
  - Otherwise the first PC captures the average expression/intensity, this constraints the next direction (recall orthogonally)
  - You can standardize each feature (e.g., gene) or sample, or both.
  - The number of eigenvalues corresponds to the smaller dimension min(n,m)
- R notes:
  - svd(dat)
  - prcomp(dat): calls svd(dat) and gives you stdev(square roots of eigenvalues) and rotation(column of eigenvectors) a.k.a loadings
- Do you start from X^T or X:
  - You can the same decomposition (but transposed), pay attention to the dimensionality of the returned matrices

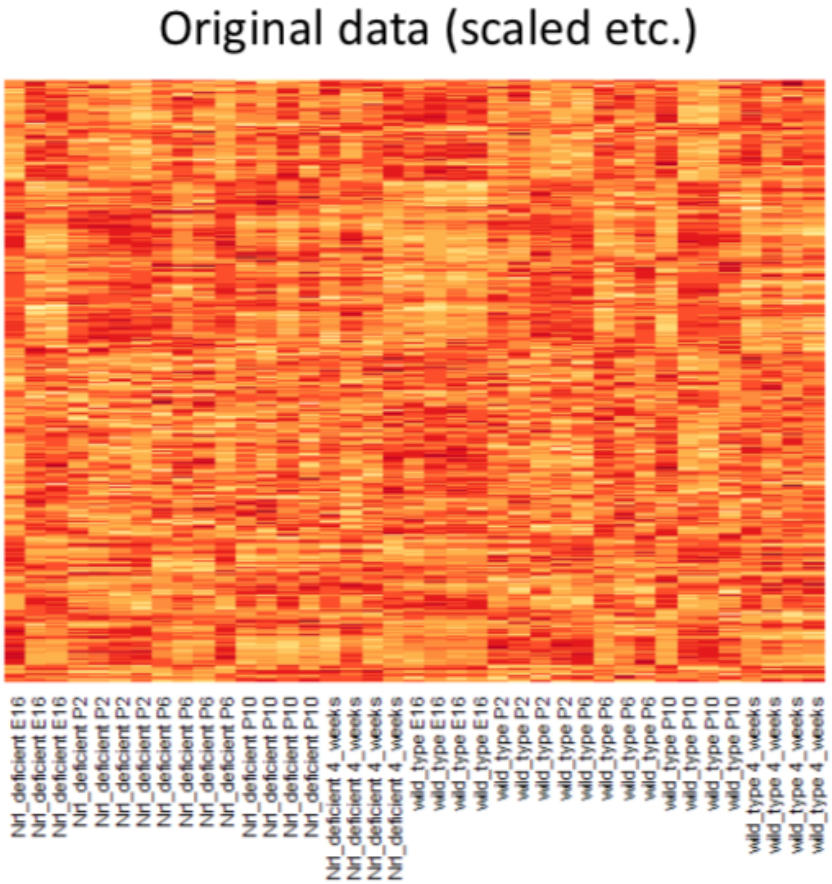
Scree plot

- Shows the relative magnitudes of the eigenvalues (can translate to proportion of variance explained very easily)



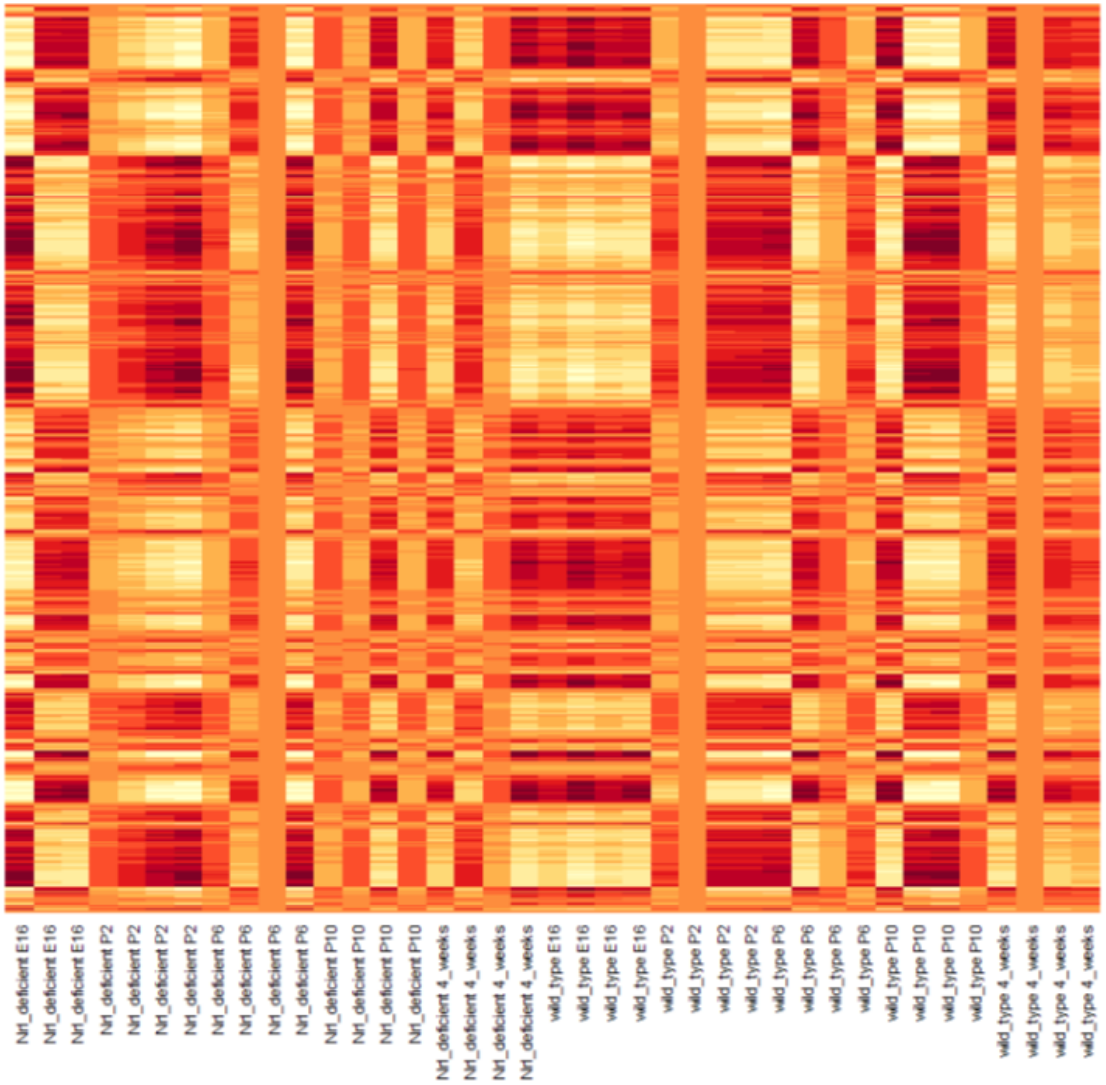


Reconstruction of data matrix X

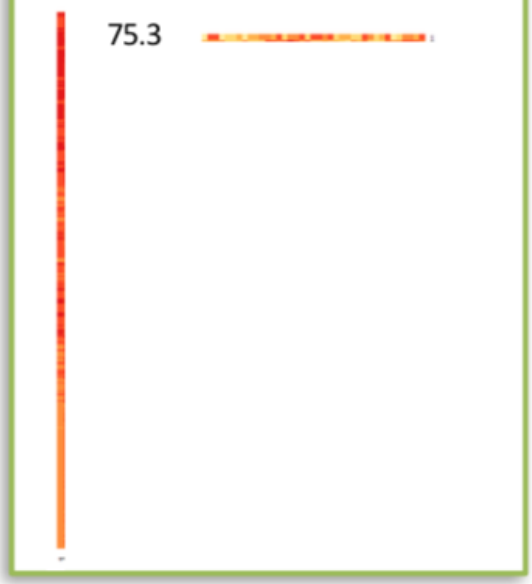


First component

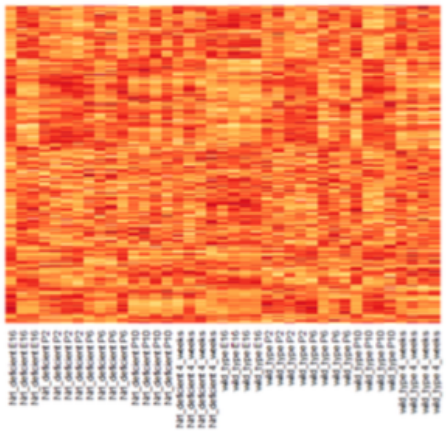
(30% of the variance)



$U_1 * D_1 * V_1^T$

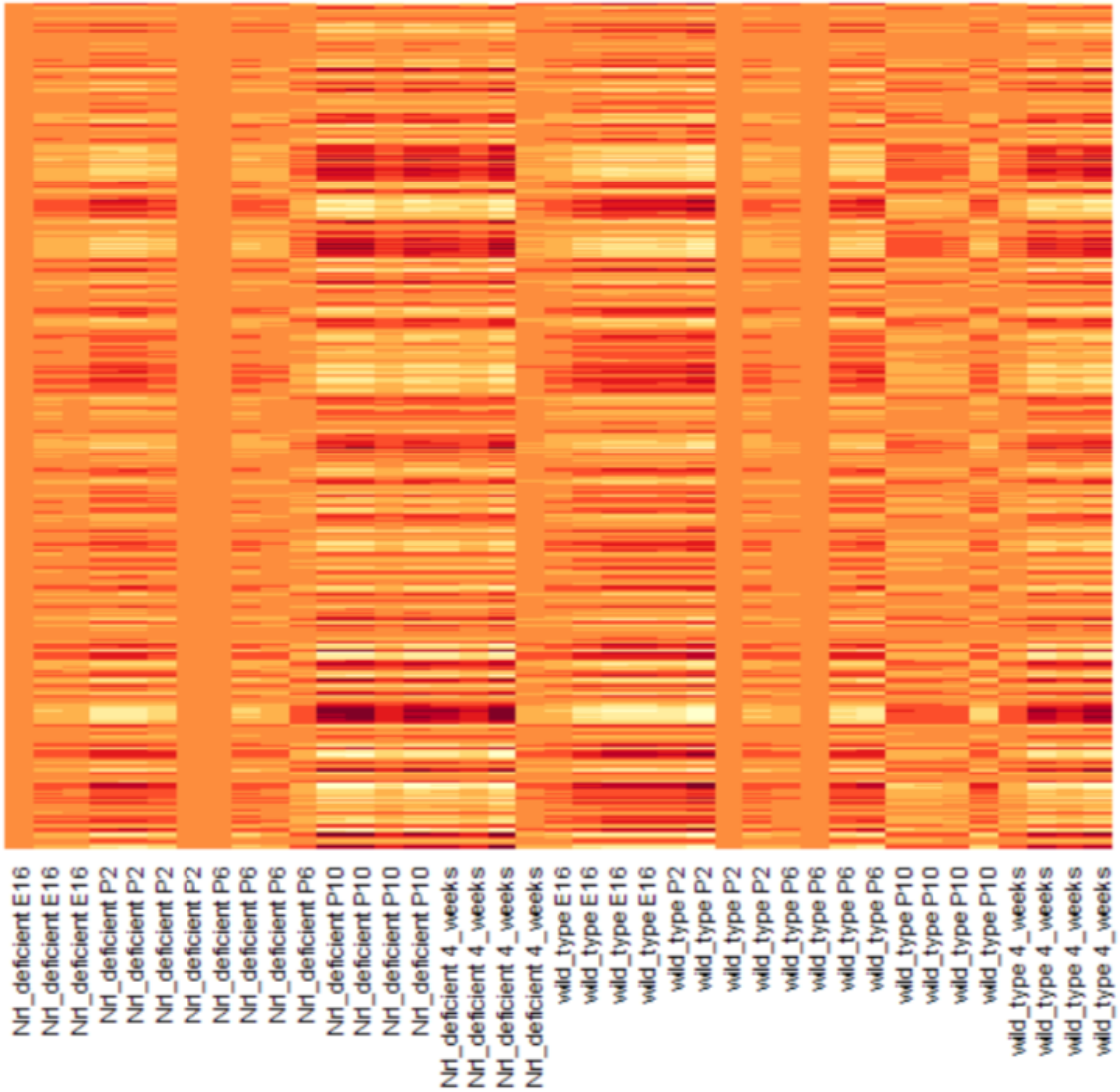


Original data

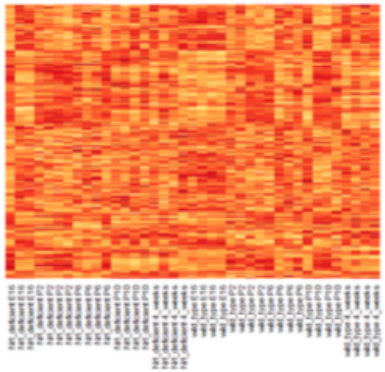


Second component

(15% of the variance)

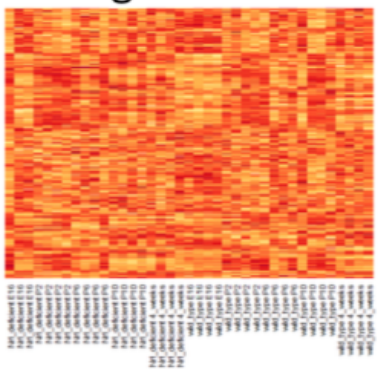


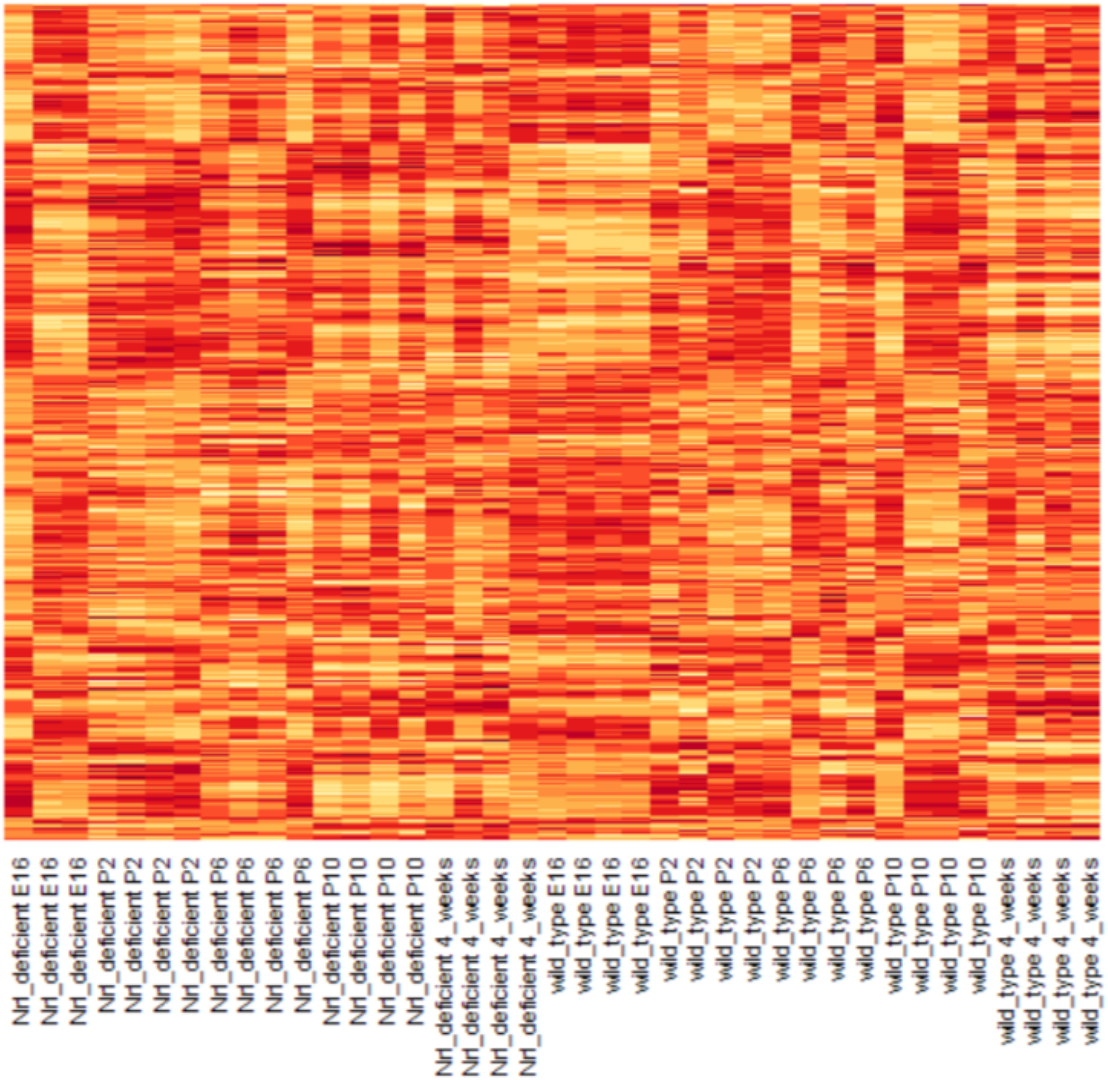
Original data



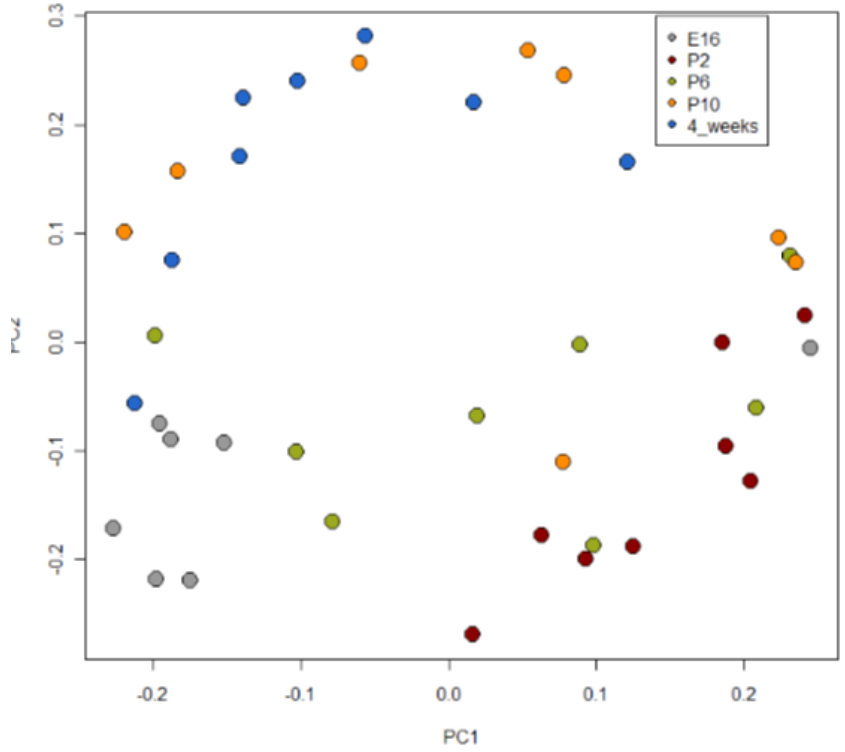
Reconstructing: first 10 components

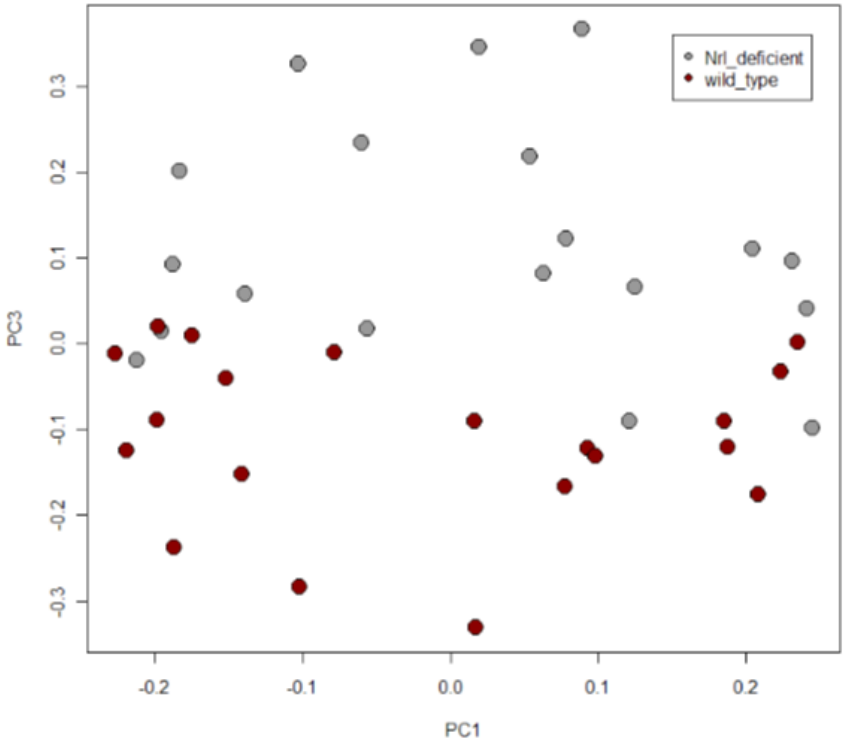
Original data





Visualization of data based on PC1-3 projection

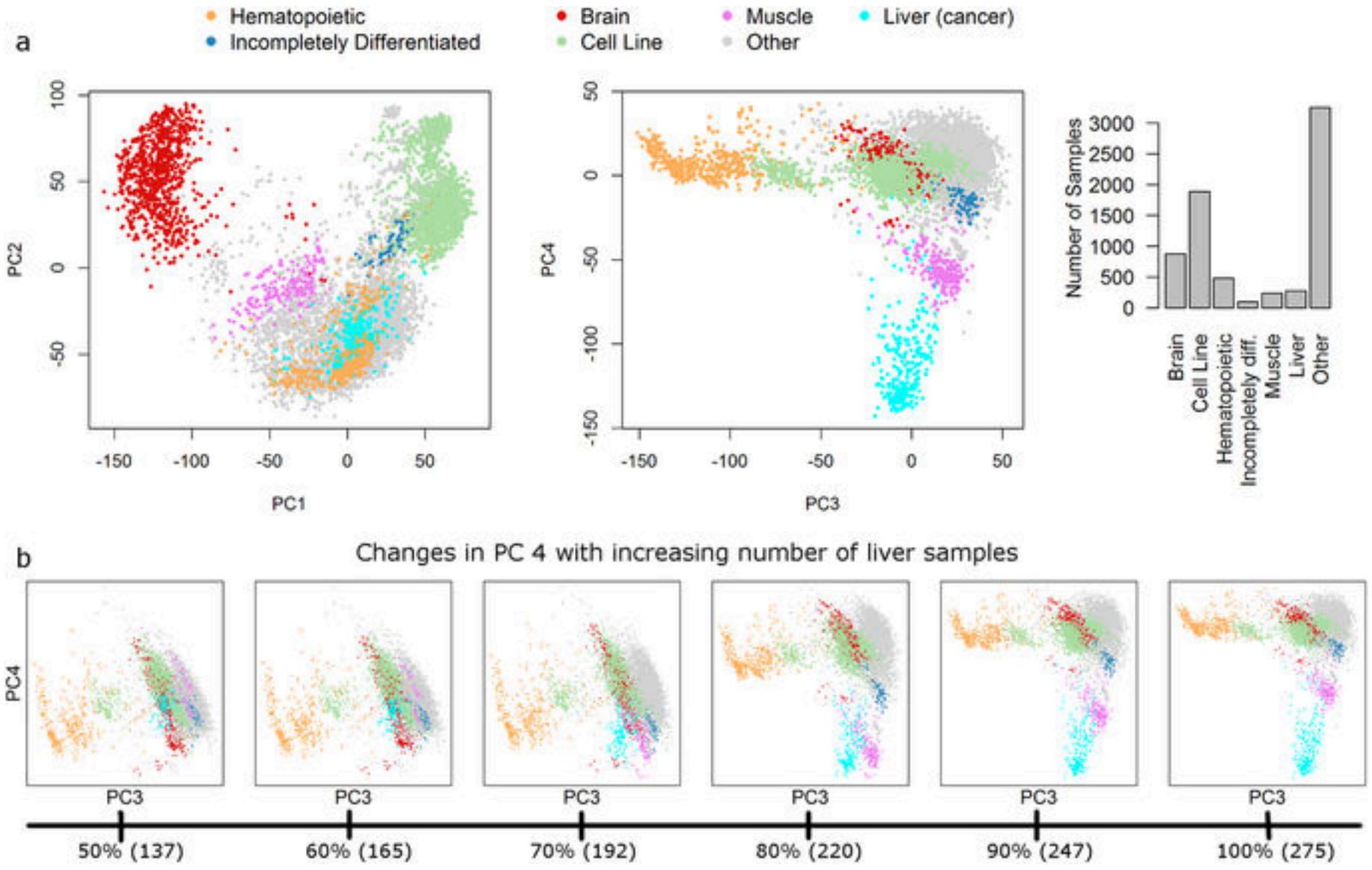




Values plotted are columns of v, as in `plot(v[,1], v[,2])`

Standard usage for PCA

- Visualize data using PCs 1-3.
- Assess correlation between top PCs and “external” variables of interest.
- Identify genes associated with different PCs (need correlation analysis; too hard to interpret the loadings)


a
Hematopoietic
Incompletely Differentiated
Brain
Cell Line
Muscle
Other
Liver (cancer)
PC2
PC1
PC4
PC3
Number of Samples
Brain
Cell Line
Hematopoietic
Incompletely diff.
Muscle
Liver
Other
b
Changes in PC 4 with increasing number of liver samples
PC4
PC3
50% (137)
60% (165)
70% (192)
80% (220)
90% (247)
100% (275)

Other applications of PCA

- SVD as a batch correction approach

- If you suspect or can show that for example PC1 correlates with batch, then you can "remove" it's effect. (show on board)

PC1 and PC2 capture population stratification in GWAS

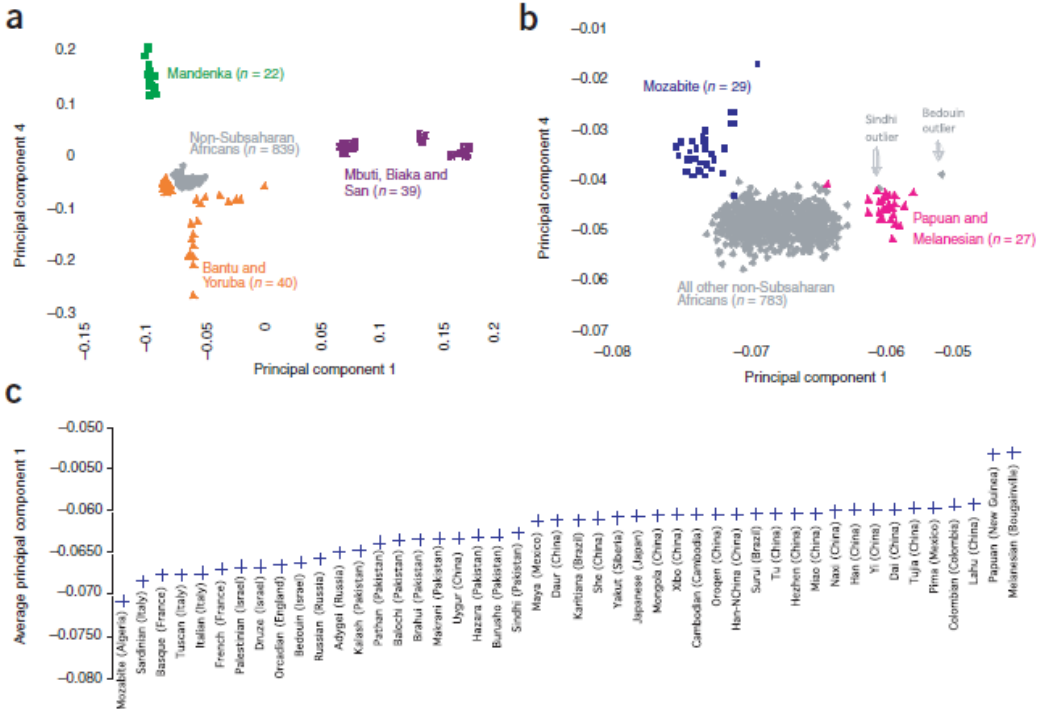


Figure 1 PCA continues to provide evidence of important migration events. (**a**) We carried out PCA on 940 individuals from the Human Genome Diversity Project that were scanned at approximately 650,000 SNPs[11] using data from 101 sub-Saharan African samples to define the PCs (Mandenka, Bantu from Kenya and South Africa, Yoruba, San, Mbuti Pygmy and Biaka Pygmy). We carried out the analysis on samples blinded to population labels (the coloring of samples was only carried out after the analysis). We plotted principal component 1 (negative values are more Bantu-related) and principal component 4 (positive values are more closely related to the Senegalese Mandenka). (**b**) Outlying populations are the Mozabite, who are more Mandenka-related, reflecting recent gene flow across the Sahara, and Papuans and Melanesians, who have inherited less Bantu-related gene flow. (**c**) To reveal the west-to-east gradient of Bantu-related ancestry across Eurasia, we averaged the first PC for each of the non-African populations and plotted the populations in rank order.

Read book chapters on PCA/clustering

- Pattern recognition in Machine Learning (Bishop)
- Elements of statistical learning (Friedman, Tibshirani, Hastie)