

Unsupervised Learning and Dimension Reduction

Part II: Principal Component Analysis and Independent Component Analysis

Karthik Devarajan, Ph.D.

Department of Biostatistics & Bioinformatics

Fox Chase Cancer Center, Temple University Health System

Center for High-Dimensional Statistics, Department of Statistics

Lewis Katz School of Medicine, Temple University, Philadelphia

Data Science Institute, University of Delaware

June 28, 2025

Background

Singular Value Decomposition

Principal Component Analysis

Independent Component Analysis

Singular Value Decomposition

- ▶ Singular Value Decomposition (SVD) is a linear decomposition of a high-dimensional matrix X .
- ▶ $X_{p \times n} = H_{p \times k} D_{k \times k} W'_{k \times n}$ where D is a non-negative, diagonal matrix such that $k = \min(p, n)$.
- ▶ For our applications, $p \ll n$ and we will take $k = p$.
- ▶ SVD also known as the Karhunen-Loeve transform in machine learning and the Hotelling transform in statistics.

Interpretation of decomposition

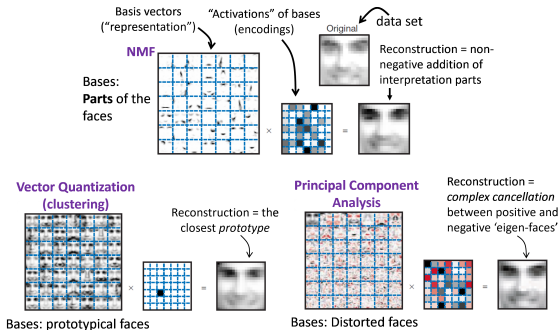
- ▶ SVD is unique up to permutations of the columns of W and H
- ▶ Columns of H and rows of W' represent the left and right singular vectors, respectively, in the decomposition.
- ▶ W and H are both orthogonal such that $W'W = H'H = I$ where I is the identity matrix.

Interpretation of decomposition

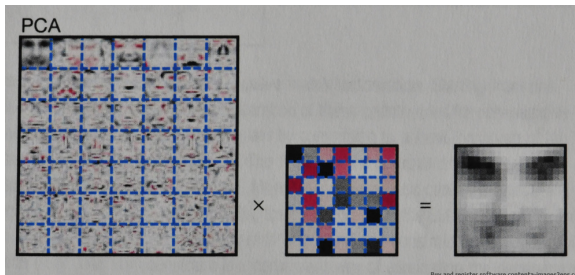
- ▶ Superposition of orthogonal components result in a distributed representation.
- ▶ Orthogonal components are typically non-overlapping with mixed signs.
- ▶ Closely related to Principal Component Analysis (PCA).

Parts-based, local representation

The non-negative matrix factorization algorithm (NMF) can be a dimensionality reduction technique that learns **parts-based representation** of the data set (Lee & Seung, *Nature*, 1999)



Distributed representation



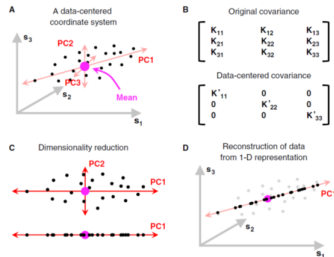
Principal Component Analysis

- ▶ PCA seeks to find a small number of orthogonal linear combinations or principal components (PC) that explain most of the variation in the data.
- ▶ This is achieved via the SVD of the data matrix X .
- ▶ In the least squares sense, PCA is the most optimal linear dimension reduction method.

Principal Component Analysis

PCA is a common dimensionality reduction method for identifying a low-dimensional representation of data

The PCA Problem:

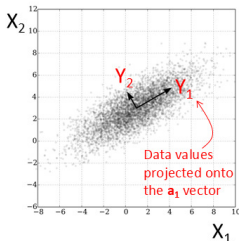


Pang, Lansdell, Fairhall (2016) *Current Biology*

Principal Component Analysis

- PCA seeks to explain the variance-covariance structure of a set of variables through a few linear combinations of these variables.
- In doing so, PCA achieves dimensionality reduction by finding a representation of the linear subspace.

The PCA Problem



- Let the $p \times n$ data matrix X have covariance matrix Σ whose eigenvalues are $\lambda_1 \geq \lambda_2 \geq \dots \lambda_p \geq 0$
- For $i=1, \dots, p$, consider the linear combinations for each column of X

$$Y_i = \mathbf{a}_i^T \mathbf{X} = a_{i1}X_1 + a_{i2}X_2 + \dots a_{ip}X_p$$
- Then, $\text{Var}(Y_i) = \mathbf{a}_i^T \Sigma \mathbf{a}_i$
- Identify the PCs that are **uncorrelated** linear combinations, $Y_1, Y_2, \dots, Y_p$, such **that their variances are as large as possible**, subject to the constraints: $\mathbf{a}_i^T \mathbf{a}_i = 1$, and the $\mathbf{a}_i$ s are perpendicular to each other

Principal Component Analysis

One solution to the PCA problem is to find the **eigenvalue-eigenvector** pairs of the **data covariance matrix**

Let Σ be the covariance matrix associated with X . Let Σ have the eigenvalue-eigenvector pairs $(\lambda_1, \mathbf{e}_1), \dots, (\lambda_n, \mathbf{e}_n)$ where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$, and $\mathbf{e}_i^T \mathbf{e}_i = 1$. The i th element of the principal component is given by

$$Y_i = \mathbf{e}_i^T \mathbf{X} = e_{i1}X_1 + e_{i2}X_2 + \dots + e_{ip}X_p, \quad i = 1, \dots, p$$

$$\begin{aligned} \text{Also, } \text{Var}(Y_i) &= \mathbf{e}_i^T \Sigma \mathbf{e}_i = \lambda_i, & i = 1, \dots, p \\ \text{Cov}(Y_i, Y_k) &= 0, & i \neq k \end{aligned}$$

Note: if $E = [\mathbf{e}_1 | \mathbf{e}_2 | \dots | \mathbf{e}_p]$

$$\begin{array}{ccc} & \begin{matrix} p \times p & p \times n \end{matrix} & \\ & \downarrow & \swarrow \\ p \times n & Y = E^T X & \end{array}$$

Since E is an orthonormal square matrix, $E^{-1} = E^T$,

$$Y = E^{-1}X$$

$$X = EY \text{ where } E \equiv H \text{ and } Y = H^T X$$

Principal Component Analysis

- ▶ Standardize X such that the covariance matrix does not depend on the scale of the expression measurements.
- ▶ The first PC is the linear combination with the largest variance; the second PC is the linear combination with the second largest variance and orthogonal to the first PC and so forth.
- ▶ The rows of $Y = E'X = H'X$ are the PCs and they are ordered based on decreasing variance.
- ▶ Typically, the first few PCs explain most of the variation in the data and are usually sufficient.

Principal Component Analysis

- ▶ The total variation is equal to the sum of the eigenvalues of the $p \times p$ covariance matrix given by $\Sigma = \frac{1}{n}XX' = HD^2H'$ where D^2 is a diagonal matrix of the ordered eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$.
- ▶ Columns of H are the eigenvectors of Σ and are known as the principal component directions of X .
- ▶ PCA is a second-order method due to its dependence on the covariance matrix of the data.

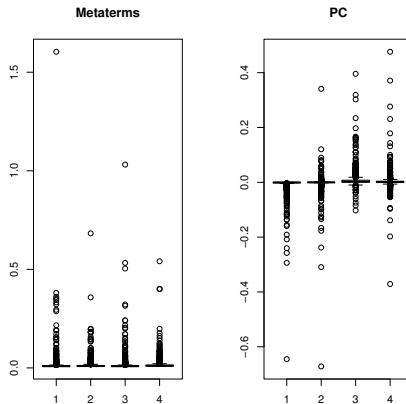
Principal Component Analysis

- ▶ PCs are uncorrelated and have variances equal to the eigenvalues of Σ .
- ▶ $\sum_{i=1}^n \lambda_i = \text{trace}(\Sigma) = \sum_{i=1}^n \text{Var}(Y_i)$
- ▶ Relative significance $\psi_i = \frac{\lambda_i}{\text{trace}(\Sigma)}$ is the cumulative proportion of variance explained by the first i PCs.
- ▶ Scree plot: ψ_i vs. i facilitates choice of the appropriate number of PCs that are required to be kept in order to explain a given proportion of the total variation.
- ▶ Search for an elbow in the plot reveals the number of PCs that explain most of the variation.

Principal Component Analysis

- ▶ For gene expression data, it can be used as an exploratory tool to graphically represent the correlations of PCs with genes.
- ▶ Plot correlations of the first PC with each gene against correlations of the second PC with each gene.
- ▶ Distance of each gene from the origin could be interpreted as its amplitude of expression in the subspace spanned by the two PCs.
- ▶ Angular distance of each gene from the x-axis could be interpreted as its phase in the transition between expression patterns defined by PCs 1 & 2.
- ▶ Same exploratory approach as above can be used in other applications as well (such as EMG data etc.).

Principal Components

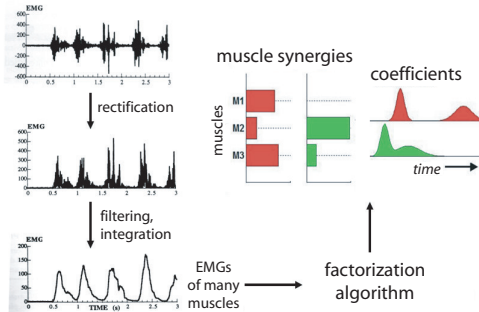


Application to EMG Data

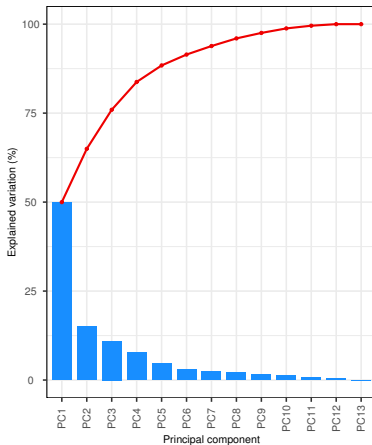
- ▶ Intramuscular EMG signals collected from all major hindlimb muscles (13) of the American bullfrog during natural behaviors (jump and swim)
- ▶ Natural locomotive behaviors constructed by motor primitives (muscle synergies) organized within the spinal cord
- ▶ Structures of these primitives centrally organized (i.e., independent of sensory feedback)
- ▶ EMGs recorded before and after the surgery of deafferentation (which removed sensory feedback)
- ▶ Four frogs each in jump or swim group before and after deafferentation
- ▶ Total of 16 data sets containing temporal EMGs for 13 muscles
- ▶ Goal is to isolate these centrally organized muscle synergies

EMG data decomposition

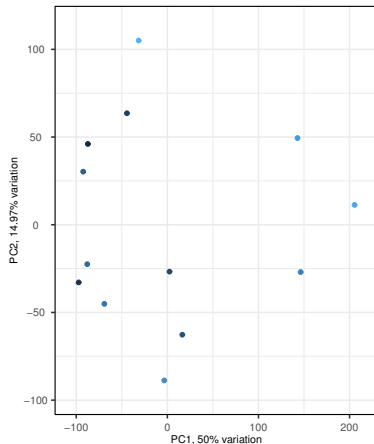
Identifying muscle synergies from EMGs using a factorization algorithm



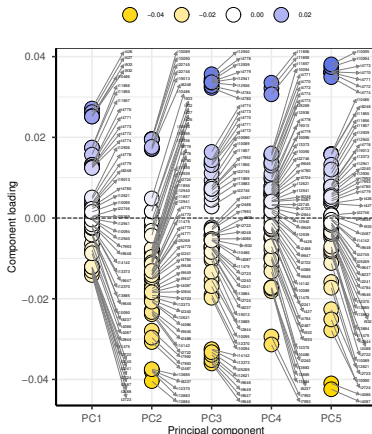
Scree plot: FB4, deafferented jump



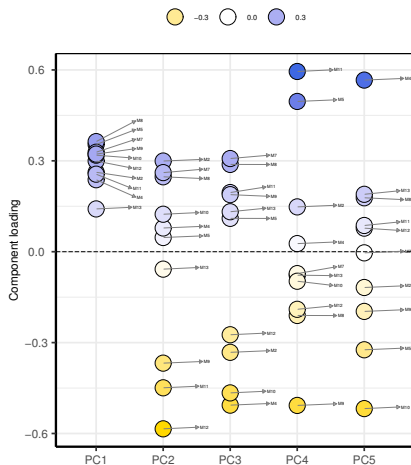
Bi plot: FB4, deafferented jump



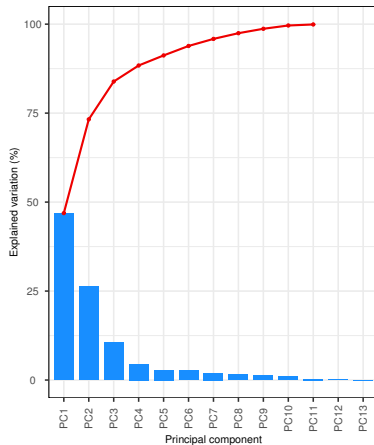
Loadings plot: FB4, deaffferented jump



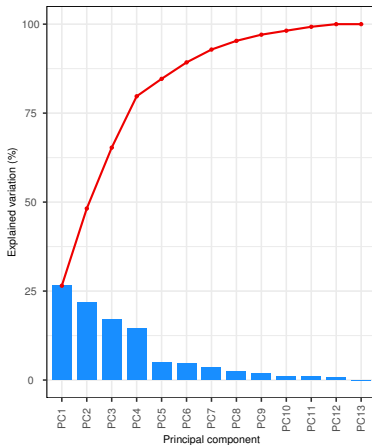
Loadings plot (T): FB4, deaffferented jump



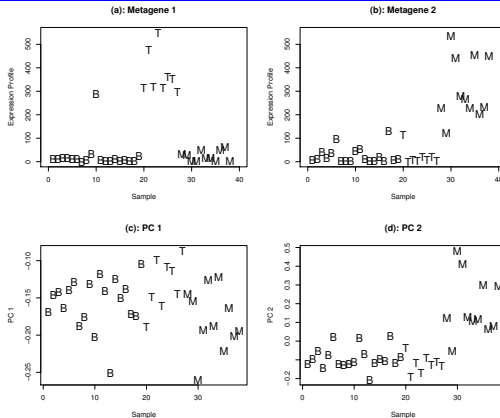
Scree plot: FB4, deafferented swim



Scree plot: FB5, deafferented swim



Metagenes and metagenes expression profiles

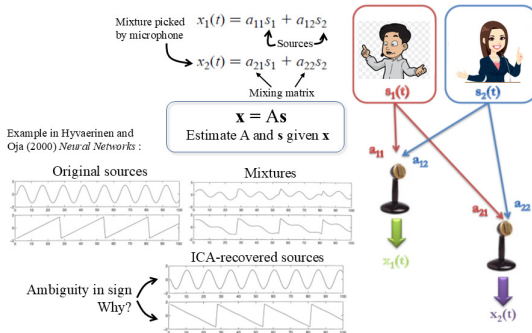


Summary of PCA

- ▶ Implementation is fast and easy; available in most standard computing packages
- ▶ Orthogonality of basis vectors guaranteed by design
- ▶ Distributed representation of basis vectors may not be realistic or desirable
- ▶ Directions with largest variance may not be physiologically or behaviorally interpretable
- ▶ Optimal number of PCs cannot be determined systematically
- ▶ Implicitly assumes temporal independence across samples

Independent Component Analysis (ICA)

Independent component analysis (ICA) solves the “cocktail party problem” by recovering the original sources from mixed signals. It is *assumed* that the number of mixtures is equal to the number of sources.



Independent Component Analysis

- ▶ Cocktail party problem: Different microphones pick up mixtures of various independent sources of sound such as music, voices from different speakers etc.
- ▶ ICA exploits the independence and non-Gaussianity of original sources to perform a blind source separation
- ▶ ICA is a higher-order method that seeks linear projections that are not orthogonal to each other necessarily but are statistically independent.
- ▶ Statistical independence between components is a much stronger assumption than uncorrelatedness and requires higher-order statistics.
- ▶ Contrasts with PCA which requires only second-order statistics and orthogonal components.

Independence and non-Gaussianity

Solving the ICA problem relies on the assumption that the sources are **statistically independent** of each other

Definition: Variables y_1 and y_2 are **independent** if information on the value of y_1 **does not give any information** on the value of y_2 , and vice versa

- ❖ Properties of independence
 - $P(y_1, y_2) = P_1(y_1)P_2(y_2)$ ◦ *factorizable PDFs*
 - $E\{h_1(y_1)h_2(y_2)\} = E\{h_1(y_1)\}E\{h_2(y_2)\}$ ◦ *factorizable expectation values of any of their functions*
- ❖ Independence implies **uncorrelatedness** (i.e., covariance = 0)

$$E\{y_1 y_2\} - E\{y_1\}E\{y_2\} = 0$$
- ❖ Uncorrelatedness does *not* imply independence

Independence and non-Gaussianity

- ▶ Independence implies uncorrelatedness; in the case of the Gaussian distribution they are equivalent.
- ▶ Independence requirement suggests that ICA may be appropriate for data that are not normally distributed.
- ▶ May not be suitable for analyzing highly correlated data., i.e identification of independent components may not be practically possible in such cases.

Some basic concepts related to ICA

- ▶ ICA is based on differential entropy H
- ▶ For a random variable Y with density $g(y)$, $H(Y) = - \int g(y) \log g(y) dy$
- ▶ Mutual information, $I(Y)$, between components of a random vector is given by $I(Y) = \sum_{j=1}^n H(Y_j) - H(Y)$.
- ▶ $I(Y)$ is the Kullback-Leibler distance between the density of Y , $g(y)$, and its independent version $\prod_{j=1}^n g_j(y_j)$.
- ▶ Among all random variables with equal variance, the Gaussian has maximum entropy.

The objective function in ICA

- ▶ Given a (whitened) data matrix X , find an orthogonal A such that the components of $\mathbf{y} = A'\mathbf{x}$ are independent and non-Gaussian.
- ▶ If $\text{Cov}(X) = I$, A is orthogonal and $Y = A'X$, then
$$I(Y) = \sum_{j=1}^n H(Y_j) - H(X) - \log |\det A| = \sum_{j=1}^n H(Y_j) - H(X).$$
- ▶ The solution to $\text{argmin}_A I(A'X)$ seeks the orthogonal transformation resulting in the most independent components.
- ▶ Equivalent to minimizing the sum of entropies of the components of Y (or, maximizing their departures from Gaussianity).

Some ICA Algorithms

- ▶ FastICA: Fast and robust fixed-point algorithms for ICA (Hyvarinen, 1999)
- ▶ ICA via InfoMax: An information-maximization approach to ICA (Bell & Sejnowski, 1995)
- ▶ ICA via JADE: Joint Approximate Diagonalization of Eigenmatrices approach (Cardoso & Souloumiac, 1993, 1996)

Some ICA Algorithms

- ▶ FastICA: Finds the orthogonal rotation matrix A that maximizes negentropy of the estimated source signals Y .
- ▶ InfoMax: Finds the orthogonal rotation matrix A that maximizes the joint entropy of a nonlinear function of the estimated source signals.
- ▶ JADE: Finds the orthogonal rotation matrix A that diagonalizes the cumulant array of the source signals.

FastICA

- ▶ Based on negentropy
- ▶ Negentropy is non-negative and measures the departure of component Y_j from Gaussianity.
- ▶ Negentropy of component Y_j is defined as $J(Y_j) = H(Z_j) - H(Y_j)$ where Z_j is Gaussian with the same variance as Y_j .
- ▶ Implemented using the approximation $J(Y_j) = [E(G(Y_j)) - E(G(Z_j))]^2$ where $G(u) = \frac{1}{a} \cosh(au)$, $a \in [1, 2]$.

Illustration of ICA

Negentropy can be used as a measure of non-Gaussianity for implementation of the ICA algorithm

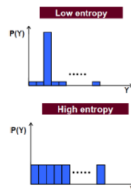
- The *entropy* of a random variable is the degree of information that observations of the variable give. The more unpredictable, the larger the entropy

For a discrete random variable Y , the entropy H is defined as:

$$H(Y) = -\sum_i P(Y = a_i) \log P(Y = a_i)$$

where a_i are the possible values of Y .

A Gaussian variable has the largest entropy among all random variables of equal variance.



Negentropy can be defined as follows:

$$J(y) = H(y_{\text{gauss}}) - H(y)$$

Gaussian random variable
with same covariance as y



- $J(y)$ is non-negative, and = 0 if y is Gaussian
- ICA: find y so that $J(y)$ is maximized

Illustration of ICA

ICA implementation that maximizes negentropy is equivalent conceptually to **minimizing the mutual information** between the estimated sources

- The mutual information I between m random variables, $y_1, \dots, y_m$, is

$$I(y_1, y_2, \dots, y_m) = \sum_{i=1}^m H(y_i) - H(y)$$

Joint entropy of $y_1, \dots, y_m$

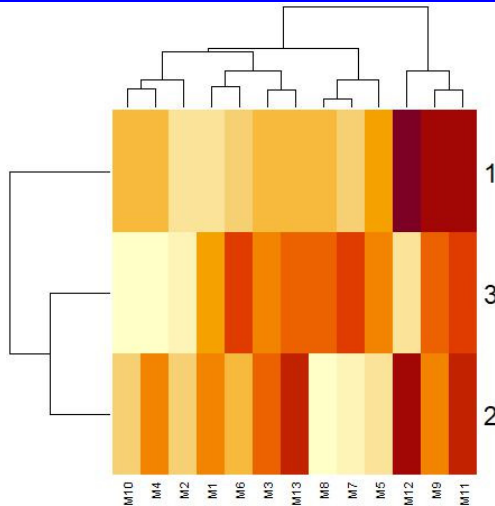
- Mutual information is non-negative and is zero iff the variables are **statistically independent**
- A natural measure of the *dependence structure* of the variables

It can be shown (Hyvärinen and Oja, 2000) that if the y_i s are uncorrelated and are of unit variance,

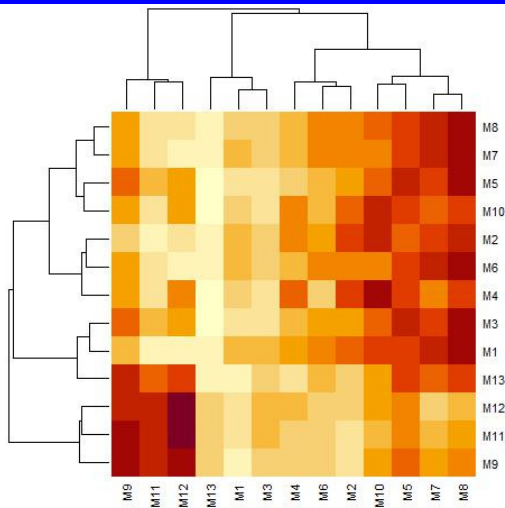
$$I(y_1, \dots, y_n) = \underbrace{C}_{\text{constant}} - \sum_i J(y_i) \leftarrow \text{negentropy}$$

Thus, in ICA, finding S that maximizes negentropy is equivalent to minimizing the mutual information between the sources

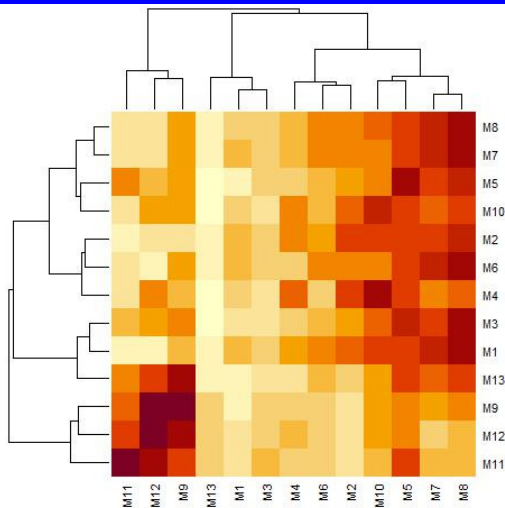
Heat map of mixing matrix A: FB4, Deaff Jump



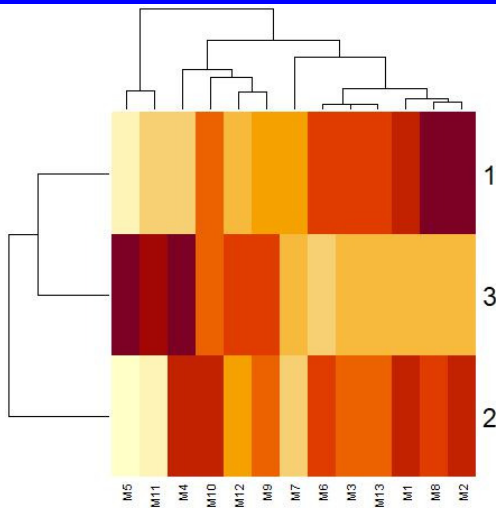
Heat map of $A'A$: FB4, Deaff Jump, #comp. = 3



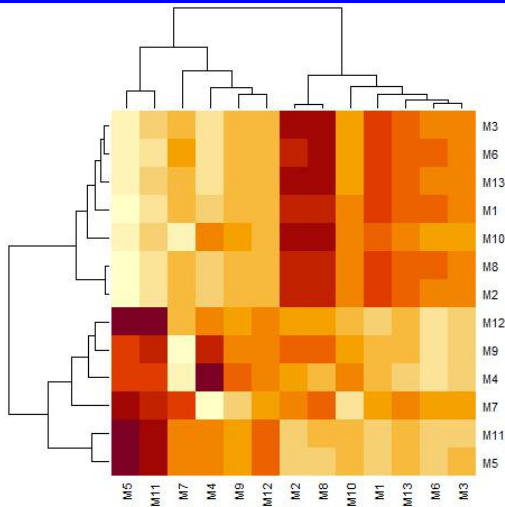
Heat map of $A'A$: FB4, Deaff Jump, #comp.= 4



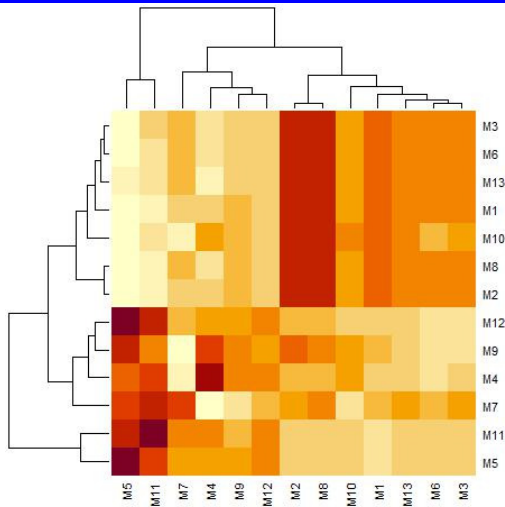
Heat map of mixing matrix A: FB5, Intact Jump



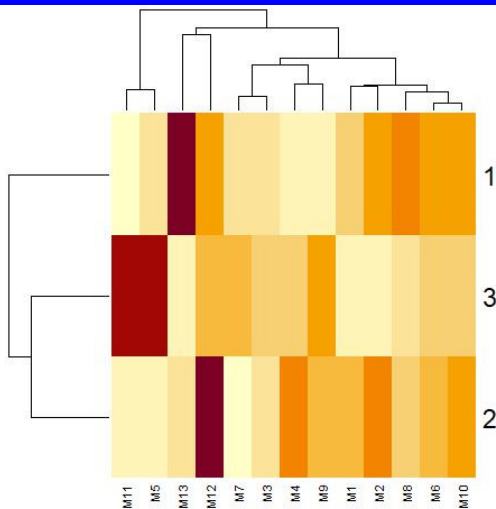
Heat map of $A'A$: FB5, Intact Jump, #comp.= 3



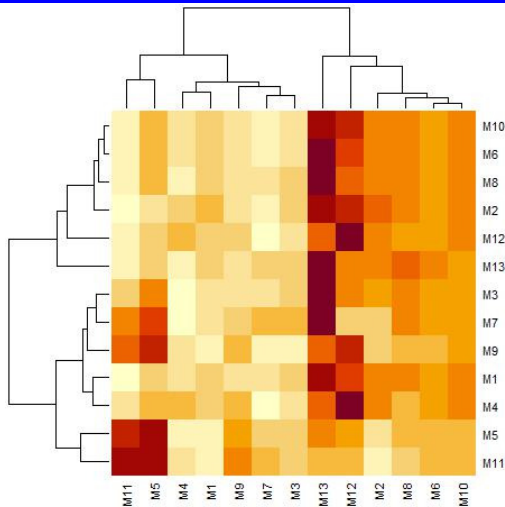
Heat map of $A'A$: FB5, Intact Jump, #comp.= 4



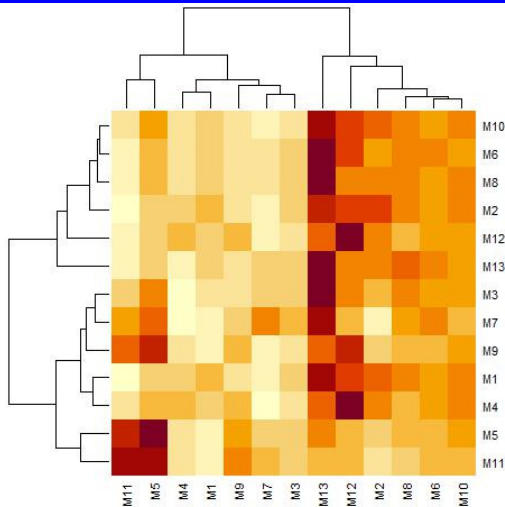
Heat map of mixing matrix A: FB11, Deaff Jump



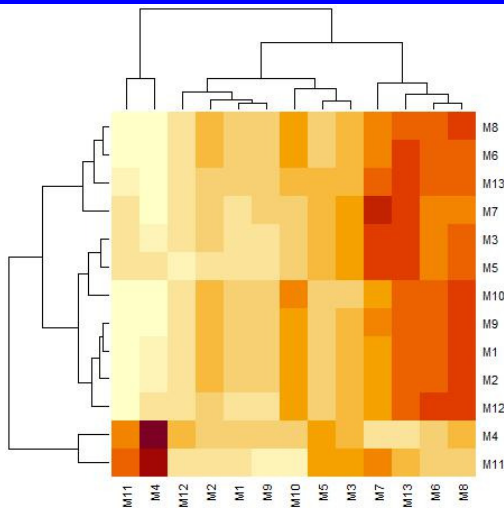
Heat map of $A'A$: FB11, Deaff Jump, #comp.= 3



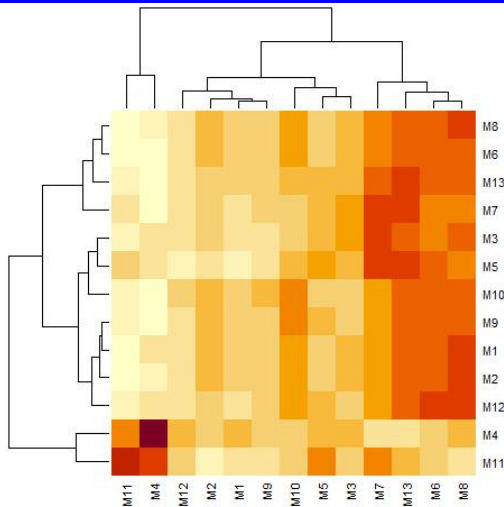
Heat map of $A'A$: FB11, Deaff Jump, #comp.= 4



Heat map of $A'A$: FB11, Intact Swim, #comp.= 3



Heat map of $A'A$: FB11, Intact Swim, #comp.= 4



Summary of ICA

- ▶ ICA seeks a sequence of orthogonal projections such that the projected data are as far from Gaussianity as possible
- ▶ For pre-whitened data, this results in independent components
- ▶ Similar to factor analysis, ICA looks for rotations that lead to independent components
- ▶ Implementation is easy. Many fast ICA algorithms are readily available
- ▶ Assumption of statistical independence between sources may help distinguish artefacts from task-related signals

Summary of ICA

- ▶ Independence assumption can be limiting due to correlated neural inputs
- ▶ No additional assumption on the signs of ICs and mixing matrix corresponds well to the presence of excitatory and inhibitory drives in nervous system
- ▶ Number of sources is assumed to equal the number of mixtures.
- ▶ Determining the optimal number of ICs may not be straightforward