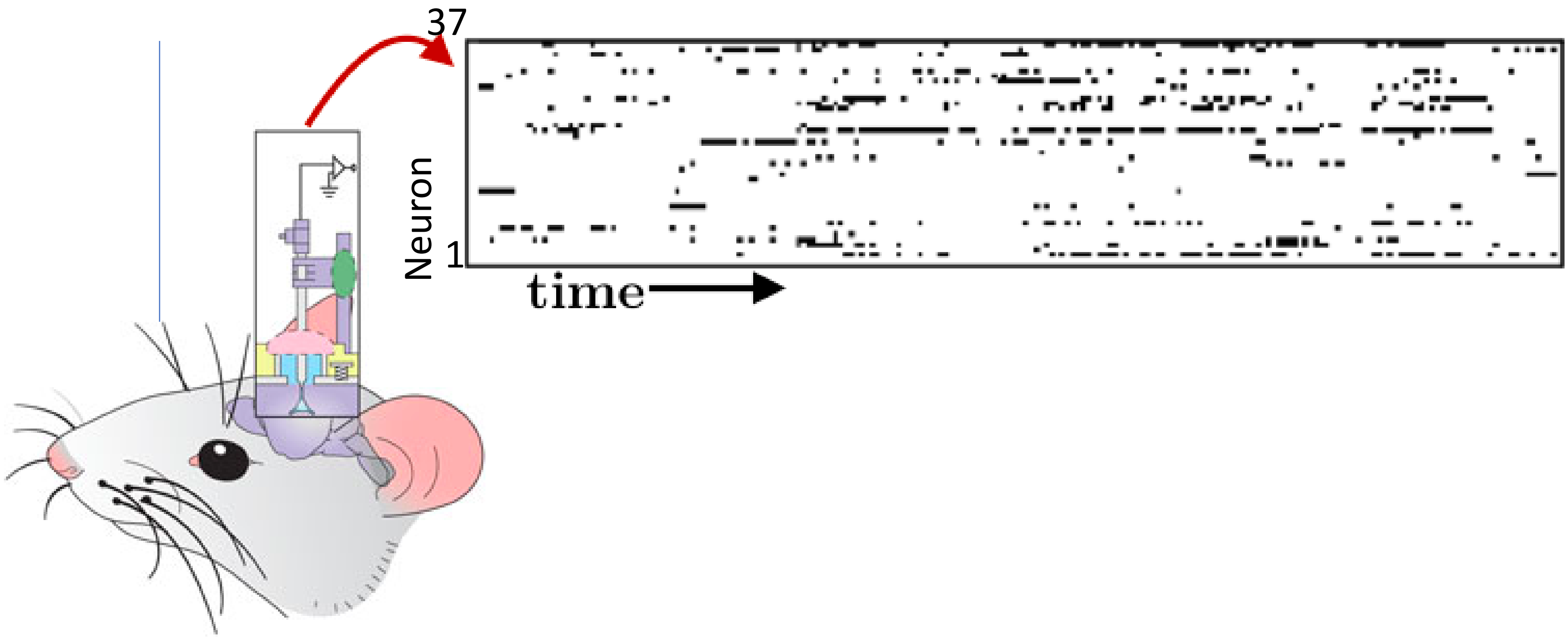
Dimensional Reduction, Clustering, Principal Component Analysis

Simona Cocco (Physics Department of ENS)
simona.cocco@phys.ens.fr

Gabrielle Girardeau (Sorbonne University)
gabrielle.girardeau@inserm.fr

I-BIO-2023 School Banyuls
Advanced Computational Analysis for Behavioral and Neurophysiological Recordings

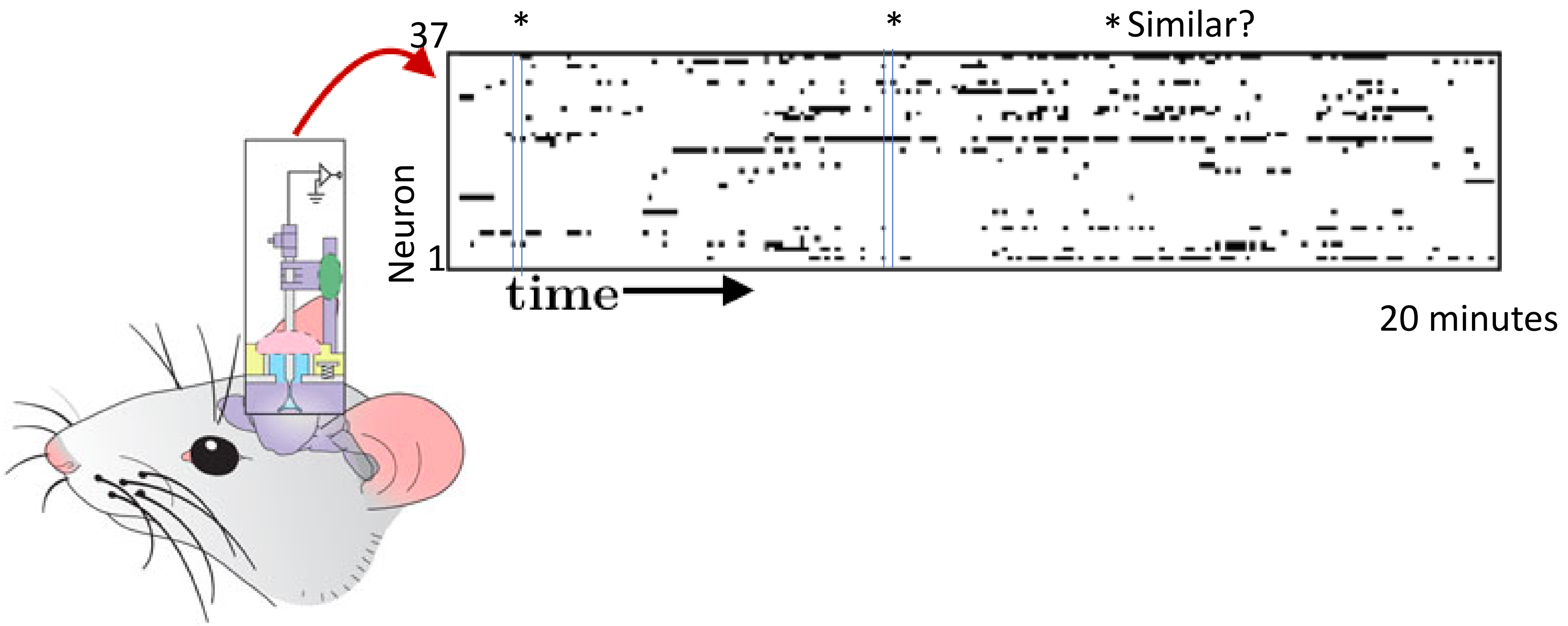
Extract patterns of the network activity from multi-electrode recording



How can we decode neural data to extract patterns of a network activity?

High dimensional problem: large number of neurons

Extract patterns of the network activity from multi-electrode recording



How can we decode neural data to extract patterns of a network activity?

High dimensional problem: large number of neurons

High Dimensional data-set

=0000100000000001000

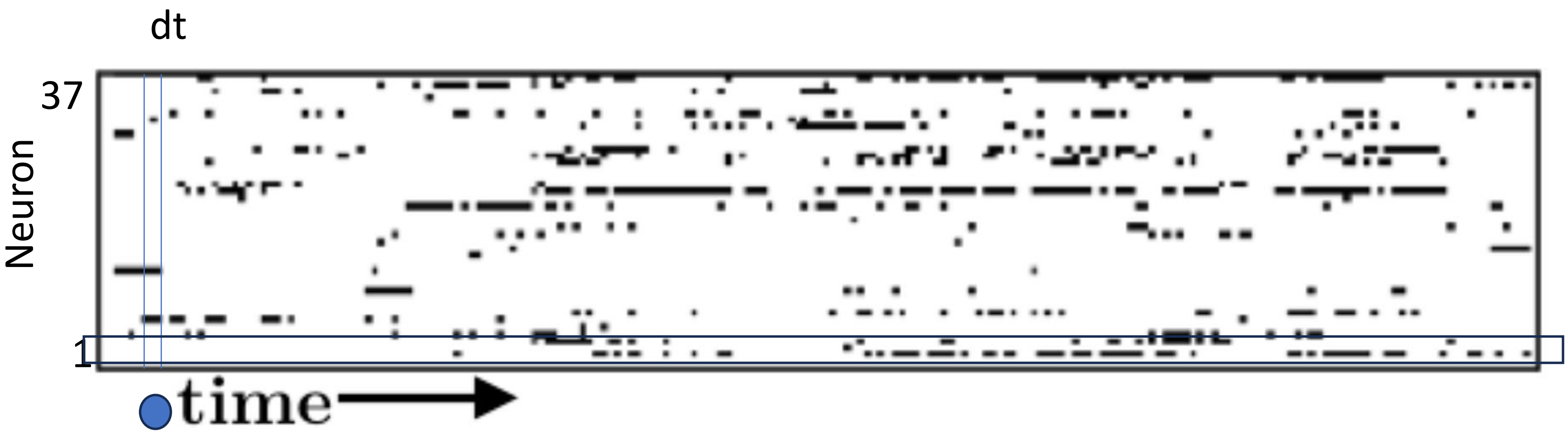
20 minutes
10^4 time windows

Each neuron can be active or non active in a time bin,

Configuration of the activity $\mathbf{y}=(y_1,y_2 \ldots y_N)$

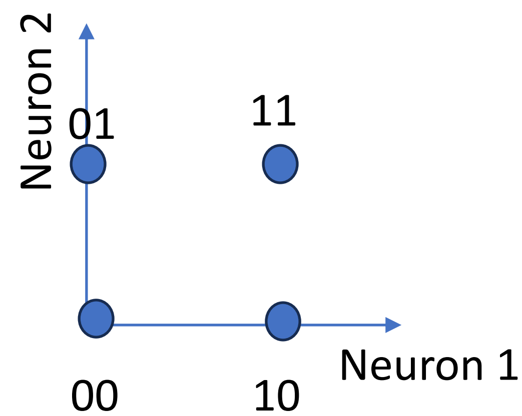
How to represent it?

High Dimensional data-set



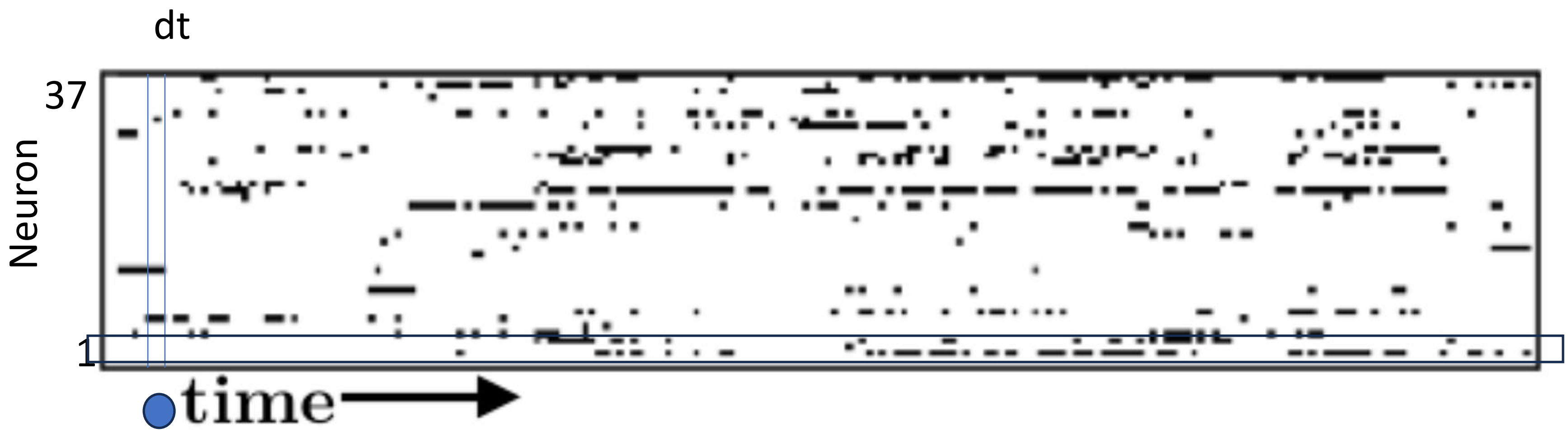
Each neuron can be active or non active in a time bin,

Case N=2 easy,
there are 2^2=4 possible configurations:



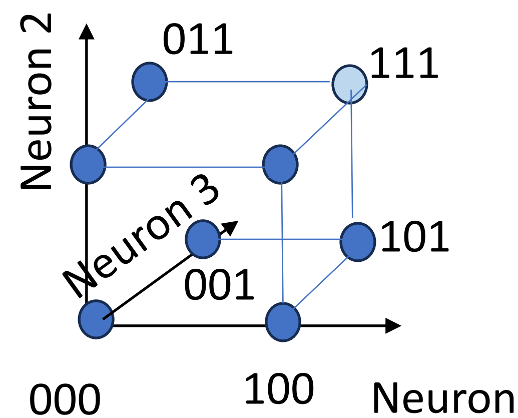
00=

High Dimensional data-set



Each neuron can be active or non active in a time bin,

Case N=3 easy,
there are 2^3=8 possible configurations:



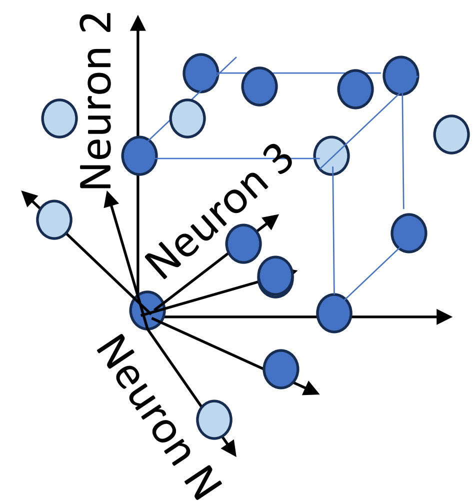
000=

High Dimensional data-set

=000010000000001000

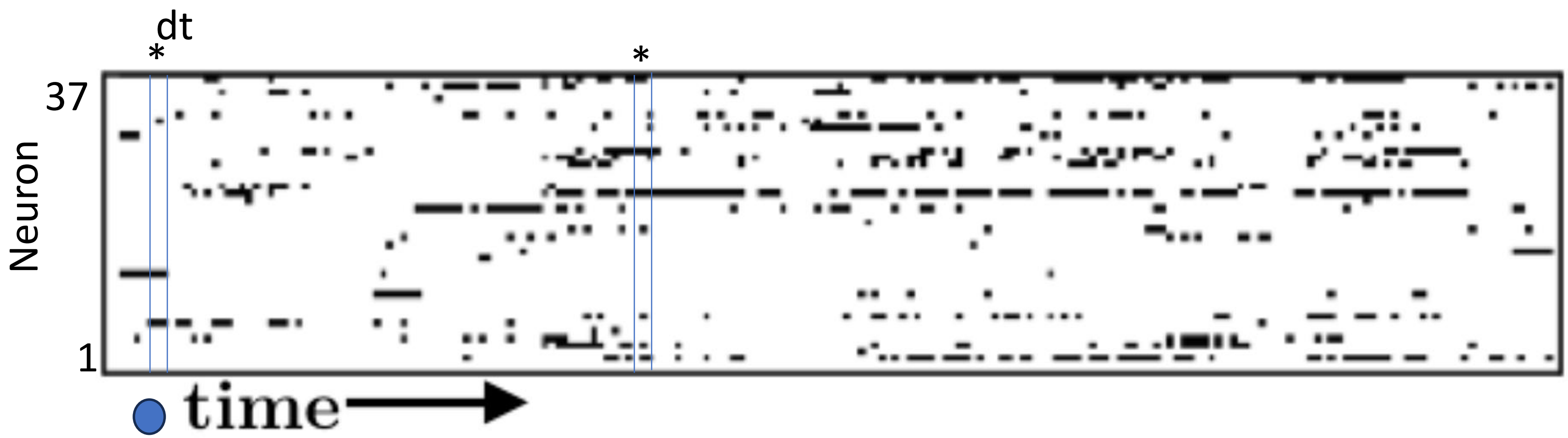
Each neuron can be active or non active in a time bin,

case N=37, difficult
there are $2^{37}=10^{11}$ possible configurations:



● $=10^{11}$
time bin $= 10^4$

High Dimensional data-set

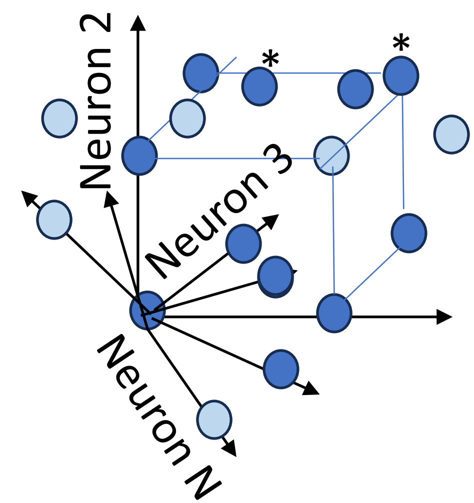


0001000000001000=

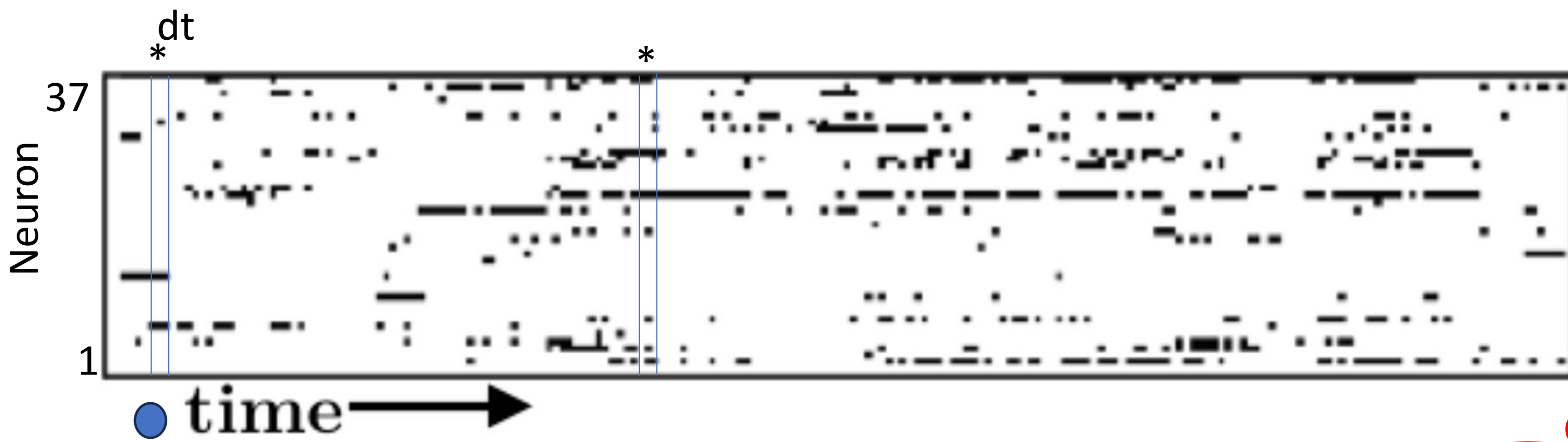
Each neuron can be active or non active in a time bin,

case N=37, difficult
there are $2^{37}=10^{11}$ possible configurations:

* similar? , nearby?



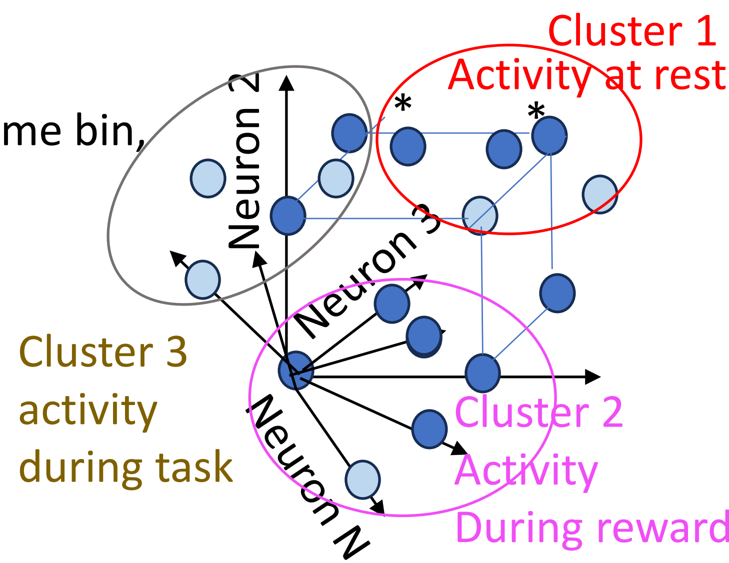
High Dimensional data-set



Each neuron can be active or non active in a time bin,

case N=37, difficult
there are $2^{37}=10^{11}$ possible configurations:

* similar? , nearby?, same cluster?



Other example: diff activity in response sounds
cfr lecture of Brice Bathellier

Plan of the lecture

- K-means

- Principal Component Analysis (PCA)

  geometrical, algebraical interpretation

- PCA on Multi-Variate Gaussian distribution

- PCA on Multi-Electrode recordings

- How many principal component to chose ? Null model based on random matrix theory, Marcenko Pastur model.

K-means

Aim: partition the observed (eg. $M=10^4$) neural configurations in K clusters , in such a way that each configuration belongs to the cluster with the nearest mean (cluster center).

K-means

Aim: partition the observed (eg. $M=10^4$) neural configurations in K clusters , in such a way that each configuration belongs to the cluster with the nearest mean (cluster center).

Algorithm is based on the iterations of 2 steps:

1. assign each configuration to the cluster with nearest center
2. compute the cluster center as the **mean** of the configurations in the cluster

-> repeat until convergence

configurations **mean** μ

010010000000100000

000010000000001000

0
0.5
0
0
1
0
0
0
0
0
0.5
0
0.5
0
0
0

K-means

Aim: partition the observed (eg.$M=10^4$) neural configurations in K clusters , in such a way that each configuration belongs to the cluster with the nearest mean (cluster center).

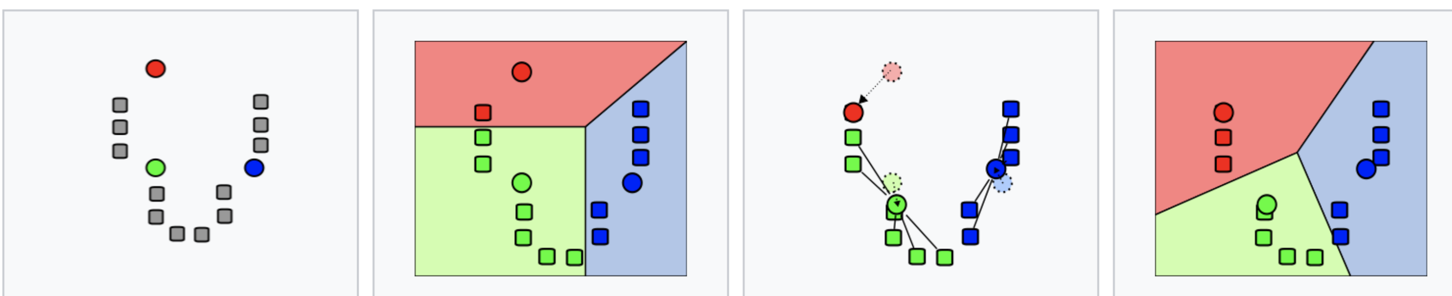
Algorithm is based on the iterations of 2 steps:

1. assign each configuration to the cluster with nearest center
2. comput the cluster center as the mean configuration

> repeat until convergence

Case K=3

Demonstration of the standard algorithm



0. initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).

1. clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

2. The centroid of each of the *k* clusters becomes the new mean.

4. Steps 2 and 3 are repeated until convergence has been reached.

From wikipedia

Limitatons of K-means

- Results may depend on Initial clusters
- Convergence to the global optimum is not guaranteed
- K is to be fixed a priori , it is a hyper-parameter :

there are empirical criteria e.g. elbow method (tutorial on zebrafish)

It is possible to reduce first the dimensionality of the space considering only the most informative directions in the N dimensional space, dicarding non informative directions (considering sampling limitations): Principal Component Analysis

Principal Component Analysis (PCA)

Goals :

- Reduce the dimensionality of a data-set.
- Increase the interpretability of data while preserving the maximum amount of information
- Help the visualization and clusterization of multi-dimensional data:

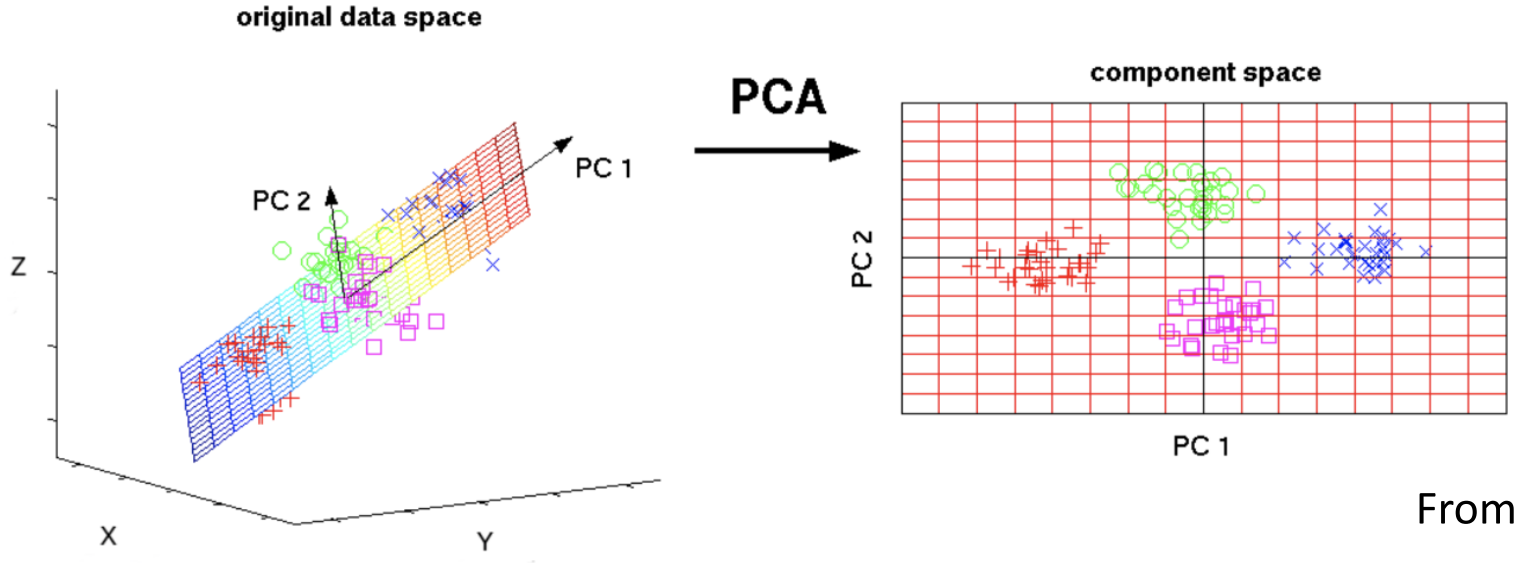
In many applications the first two components of the PCA are used to represent the data, identify clusters of closely related data points, and represent neural trajectories during the recording.

How?

- A linear transformation in a new coordinate system where most of the variation of the data can be explained with fewer dimensions than initial data.
- In practice find the eigenvectors and associated eigenvalues of the correlation matrix.

Principal Component Analysis (PCA)

A change of variables in a new coordinate system: **n**ew directions which constitute an **orthonormal** basis.



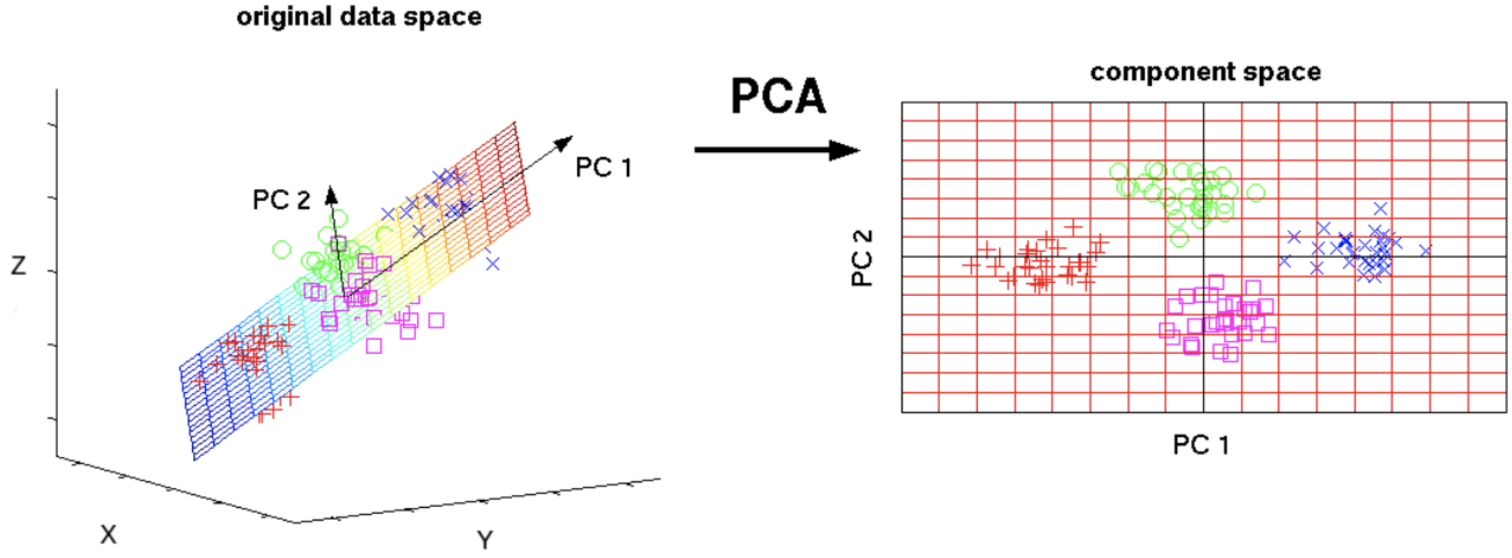
From wikipedia

- The first principal component of a set of variable is the variable formed by a linear combination of the original variables and explaining as much variance in the data as possible

- The second component explains most variance in the data in the space orthogonal to the first direction and so on..

Connection between PCA and K-means

Principal directions connect the center of the cluster.



PC1 is the line connecting the 2 clusters which are the most distant : maximal Variability of the data. PC2 direction connecting the 2 clusters.

By reducing the dimensionality the distance between cluster is kept while the distance between points in the same cluster is reduced through the projection. This allows to reduce noise.

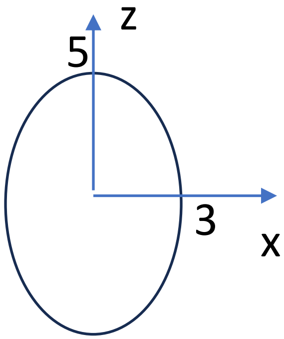
PCA in geometry and linear algebra

PCA was invented in 1901 by Karl Pearson as an extension of the principal axis theorem in geometry and linear algebra.

The principal axis are lines   generalizing of the major and minor axis of an ellipse. They are perpendicular, and can be found by a matrix diagonalization.

Example 1:

$$\frac{x^2}{9} + \frac{z^2}{25} = 1$$



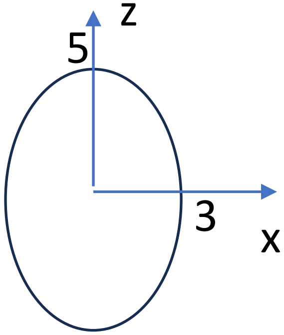
PCA in geometry and linear algebra

PCA was invented in 1901 by Karl Pearson as an extension of the principal axis theorem in geometry and linear algebra.

The principal axis are lines generalizing of the major and minor axis of an ellipse. They are perpendicular, and can be found by a matrix diagonalization.

Example 1:

$$\frac{x^2}{9}+\frac{z^2}{25}=1 \quad = \sum_{i,j} y_i \, T_{ij} \, y_j$$ with



$\mathbf{y} = \begin{pmatrix} x \\ z \end{pmatrix}$ and $\widehat{T} = \begin{pmatrix} 1/9 & 0 \\ 0 & 1/25 \end{pmatrix}$

PCA in geometry and linear algebra

PCA was invented in 1901 by Karl Pearson as an extension of the principal axis theorem in geometry and linear algebra.

The principal axis are lines generalizing of the major and minor axis of an ellipse. They are perpendicular, and can be found by a matrix diagonalization:

Example 2:

$$5\,x^2 + 8\,x\,z + 5\,z^2 = 1$$

This corresponds to matrix $\widehat{T} = \begin{pmatrix} 5 & 4 \\ 4 & 5 \end{pmatrix}$

PCA in geometry and linear algebra

PCA was invented in 1901 by Karl Pearson as an extension of the principal axis theorem in geometry and linear algebra.

The principal axis are lines generalizing of the major and minor axis of an ellipse. They are perpendicular, and can be found by a matrix diagonalization

Example 2:

$$5\,x^2 + 8\,x\,z + 5\,z^2 = 1 \qquad \widehat{T} = \begin{pmatrix} 5 & 4 \\ 4 & 5 \end{pmatrix}$$

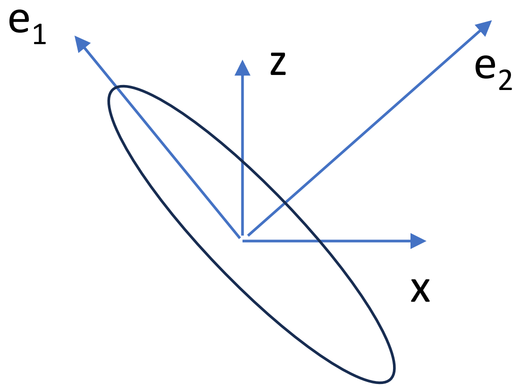
Diagonalizing matrix : solving

$$\sum_j \widehat{T_{ij}}\, e_j = \lambda\, e_i$$

λ_1=1 $\quad \mathbf{e_1} = \frac{1}{\sqrt{2}}\begin{pmatrix} -1 \\ 1 \end{pmatrix}$

λ_2=9 $\quad \mathbf{e_2} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$

e: eigenvectors, λ:eigenvalues



PCA in geometry and linear algebra

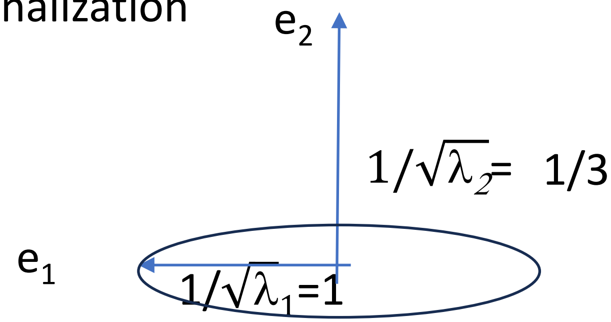
PCA was invented in 1901 by Karl Pearson as an extension of the principal axis theorem in geometry and linear algebra.

The principal axis are lines generalizing of the major and minor axis of an ellipse. They are perpendicular, and can be found by a matrix diagonalization

Example 2:

$$\lambda_1 {e_1}^2 + \lambda_2 {e_2}^2 \ = 1$$

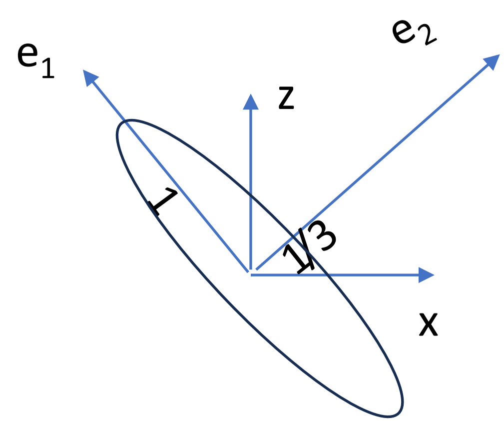
$$\widehat{T} = \begin{pmatrix} 1 & 0 \\ 0 & 9 \end{pmatrix}$$



$$\lambda_1 (x+z)^2 + \lambda_2 (x-z)^2 \ = 1$$

λ_1=1 $\mathbf{e_1} = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 1 \end{pmatrix}$

λ_2=9 $\mathbf{e_2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$



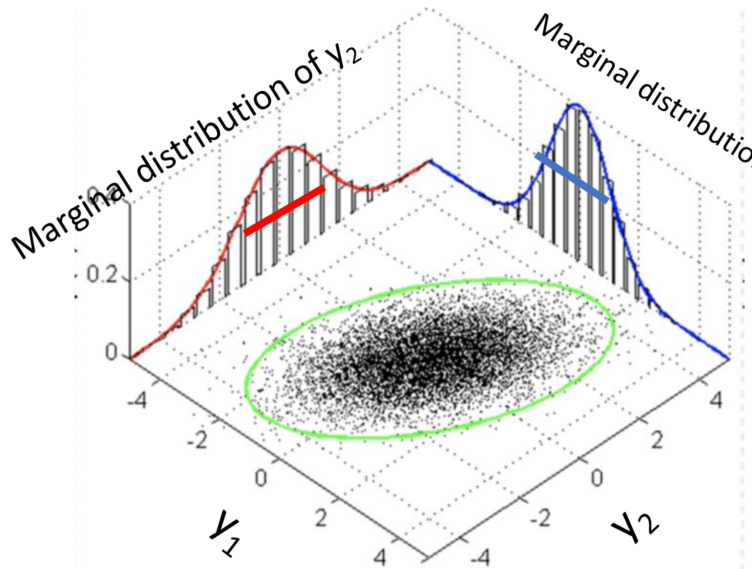
PCA and Multivariate Gaussian distributions

Simple model of stochastic variables which are correlated and in a high dimensional space:

Joint distribution of **y**=(y_1,y_2,…y_N):

$$P(\mathbf{y}|\hat{T}) \propto e^{-\frac{1}{2}\sum_{i,j} y_i T_{ij} y_j}$$

Drawing values from the distribution for N=2



PCA and Multivariate Gaussian distributions

Simple model of stochastic variables which are correlated and in a high dimensional space:

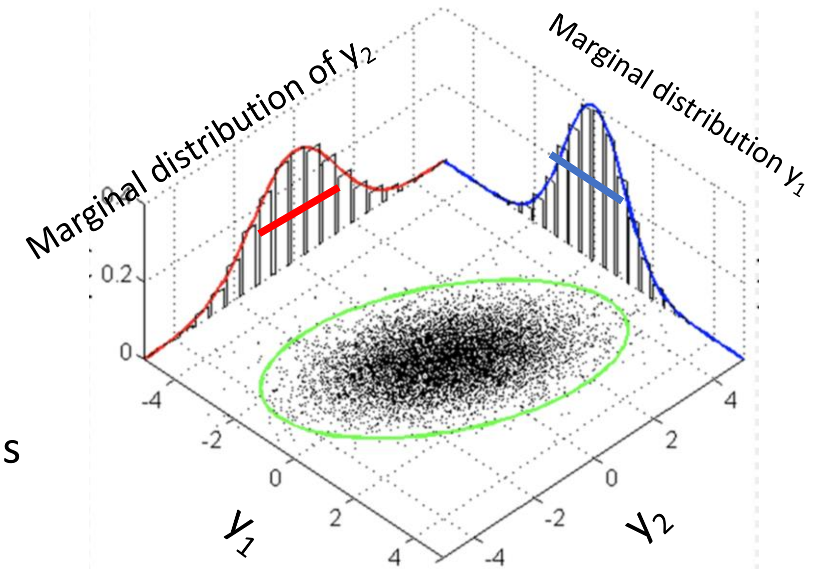
Joint distribution of $\mathbf{y}=(y_1,y_2,\ldots y_N)$:

$$P(\mathbf{y}|\hat{T}) \propto e^{-\frac{1}{2}\sum_{i,j} y_i T_{ij} y_j}$$

$\mu_i = \int d\mathbf{y}\, P(y|\hat{T})\, y_i = 0$ Averages

$C_{ij} = \int d\mathbf{y}\, P(y|\hat{T})\, y_i y_j = [\hat{T}^{-1}]_{ij}$ Covariations

Drawing values from the distribution for N=2



$$\hat{T} = \begin{array}{c|c} 3.5 & 1 \\ \hline 1. & 1.5 \end{array} \qquad \hat{C} = \begin{array}{c|c} 0.35 & -0.23 \\ \hline -0.23 & 0.82 \end{array}$$

PCA and Multivariate Gaussian distributions

Simple model of stochastic variables which are correlated and in a high dimensional space:

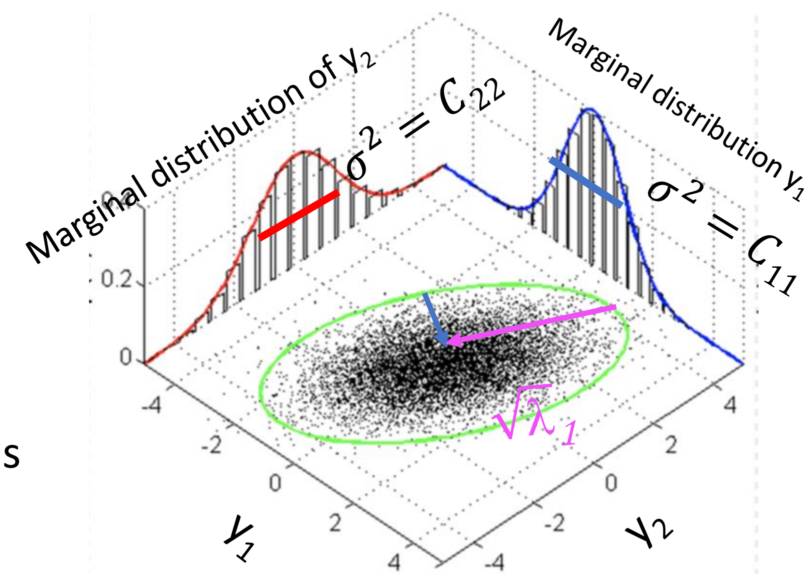
Joint distribution of $\mathbf{y}=(y_1,y_2,\ldots y_N)$:

$$P(\mathbf{y}|\hat{T}) \propto e^{-\frac{1}{2}\sum_{i,j} y_i T_{ij} y_j}$$

$\mu_i = \int d\boldsymbol{y}\, P(y|\hat{T})\, y_i = 0$ Averages

$C_{ij} = \int d\boldsymbol{y}\, P(y|\hat{T})\, y_i y_j = [\hat{T}^{-1}]_{ij}$ Covariations

Drawing values from the distribution for N=2



$\hat{T}$ =

3.5	1
1.	1.5

$\hat{C}$ =

0.35	-0.23
-0.23	0.82

$\lambda^c_1=0.92$, $\lambda^c_2=0.26$,

Principal components are (principal axis of the ellipse, or direction of maximal variability) are the eigenvectors of the covariance matrix $\hat{C}$ (same as the ones of $\hat{T}$), and the eigenvalues of $\hat{C}$ (inverse of the ones of $\hat{T}$) give the variances along them.

PCA and Multivariate Gaussian distributions

N= 3 variables

$$P(\mathbf{y}|\hat{T}) \propto e^{-\frac{1}{2}\sum_{i,j} y_i T_{ij} y_j}$$

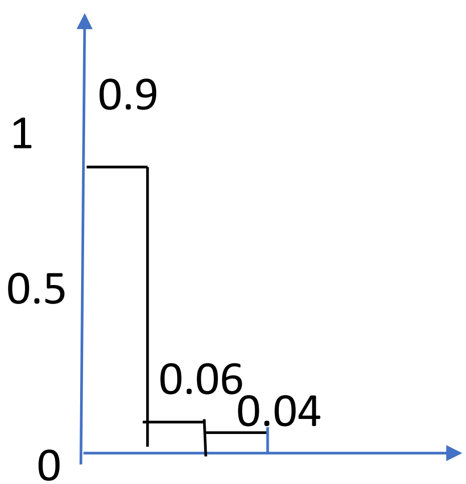
$\hat{T}$ =

3.5	-2	0
-2	1.5	0.5
0	0 .5	2.5

Eigenvalues and eigenvectors of the covariance matrix:

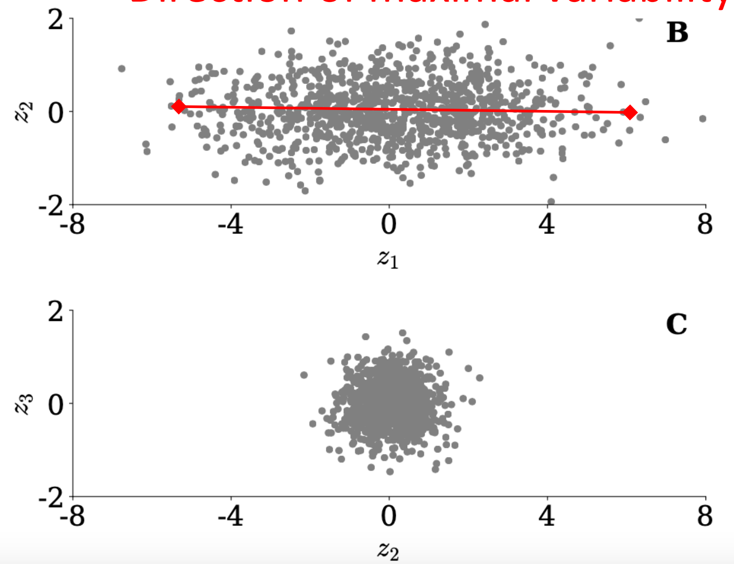
$\lambda^c{}_1 = 5.4 \quad \lambda^c{}_2 = 0.39 \quad \lambda^c{}_3 = 0.21$ $\quad \mathbf{e}_{1,}\mathbf{e}_{2,}\mathbf{e}_3$

Fraction of explained variability $\frac{\lambda_i}{\sum \lambda_i}$



Direction of maximal variability

$$z_\mu = \mathbf{e}_\mu^T \cdot \mathbf{y}$$

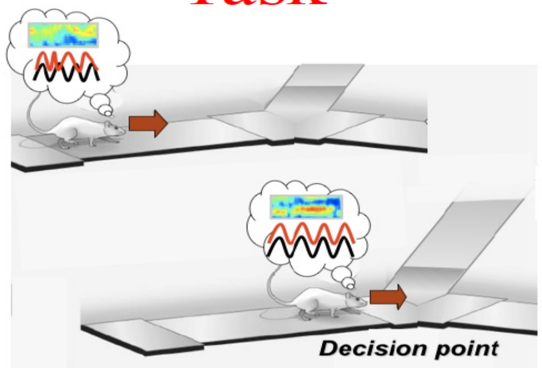


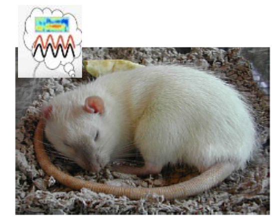
$\sqrt{\lambda^c{}_1/\lambda_2{}^c}$ =4.7

$\sqrt{\lambda_2{}^c\ /\lambda_3{}^c}$ =1.4

Application of PCA to neural data

Sleep Pre → Task → SleepPost





Replay of rule-learning related neural patterns in the prefrontal cortex during sleep

nature neuroscience (2009)

Adrien Peyrache[1], Mehdi Khamassi[1,2], Karim Benchenane[1], Sidney I Wiener[1] & Francesco P Battaglia[1,3]

Peyrache et al JCN 2009

Application of PCA to neural data

Renormalise the variable be centered in 0 and with unitary variance : the z-score

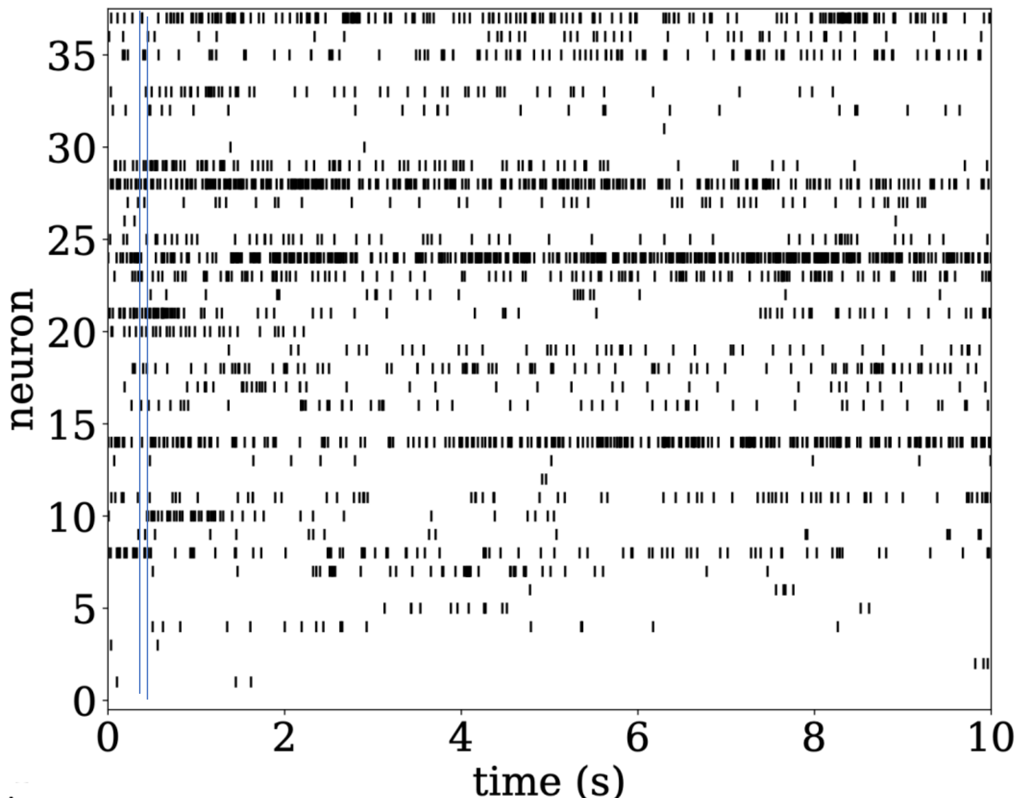
Number of spikes in a time window:
$s_i(t_b)=0,1,2,\ldots 10$

$$\mu_i = \langle s_i \rangle = \frac{1}{M} \sum_{b=1}^{M} s_i\,(b)$$

$$\sigma_i^2 = \frac{1}{M-1} \sum_{b=1}^{M} \left[s_i\,(b) - \mu_i \right]^2$$

z-score

$$y_i(b) = \frac{s_i\,(b) - \mu_i}{\sigma_i}\,,$$



$$\Delta t = 100 \text{ ms}$$

The variable y are well described by stochastic variables from a multivariate Gaussian distribution

Application of PCA to neural data

Renormalise the variable be centered in 0 and with unitary variance : the z-score

Number of spikes in a time window:
$s_i(t_b)=0,1,2,\ldots10$

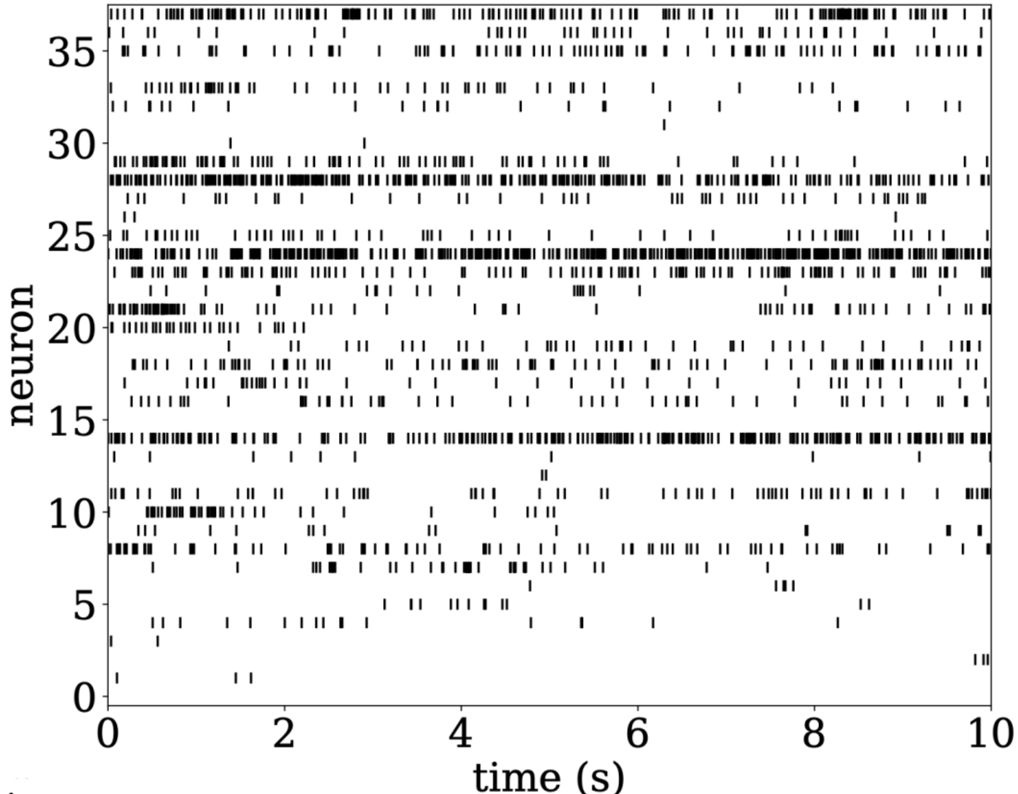
z-score

$$y_i(b) = \frac{s_i(b) - \mu_i}{\sigma_i} ,$$

We can estimate the covariance matrix from data:

$$\widehat{C}_{ij} \equiv \int d\boldsymbol{y}\, P(\boldsymbol{y}|\widehat{T})\, y_i y_j$$

$$C_{ij} = \frac{1}{M} \sum_{b=1}^{M} y_{bi} y_{bj} \ .$$

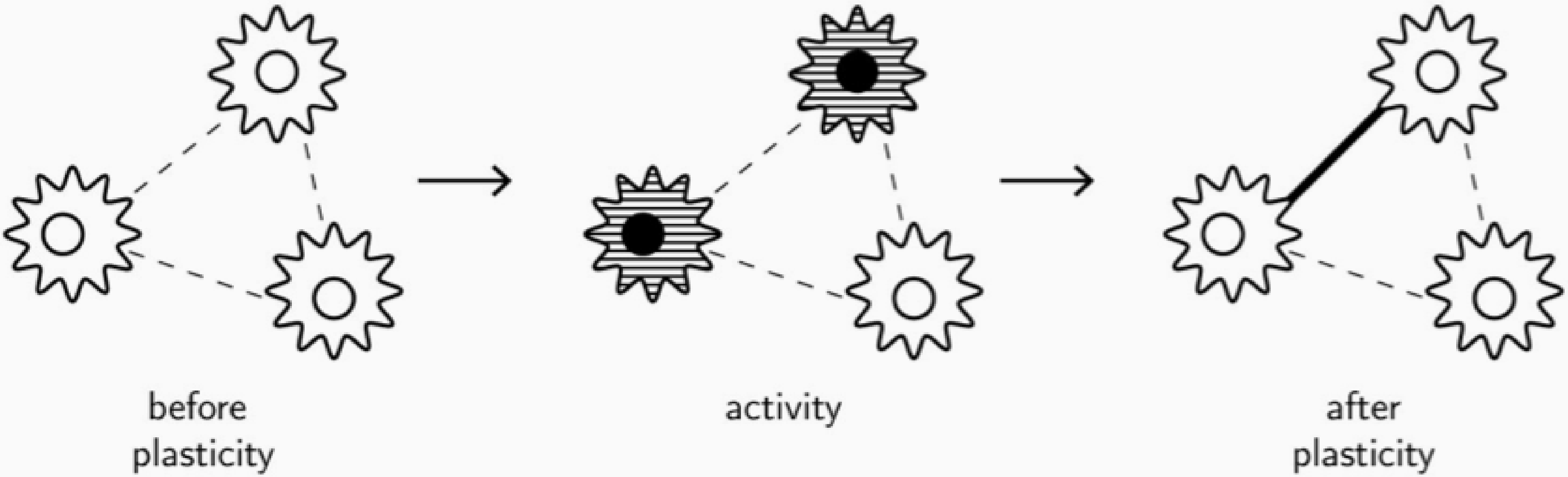


$\Delta t = 100$ ms

The variable y are well described by stochastic variables from a multivariate Gaussian distribution

Hebbian learning

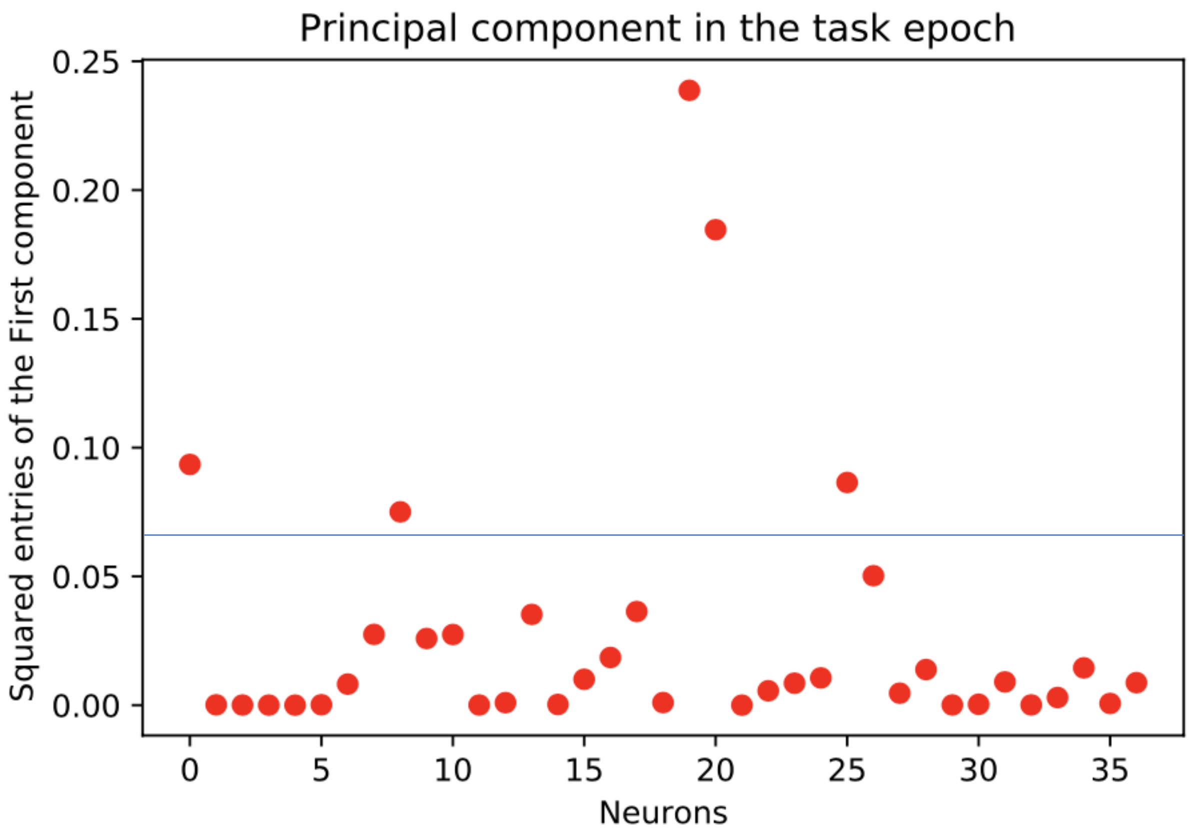
Cells which fire together wire together



Hebb (1949)

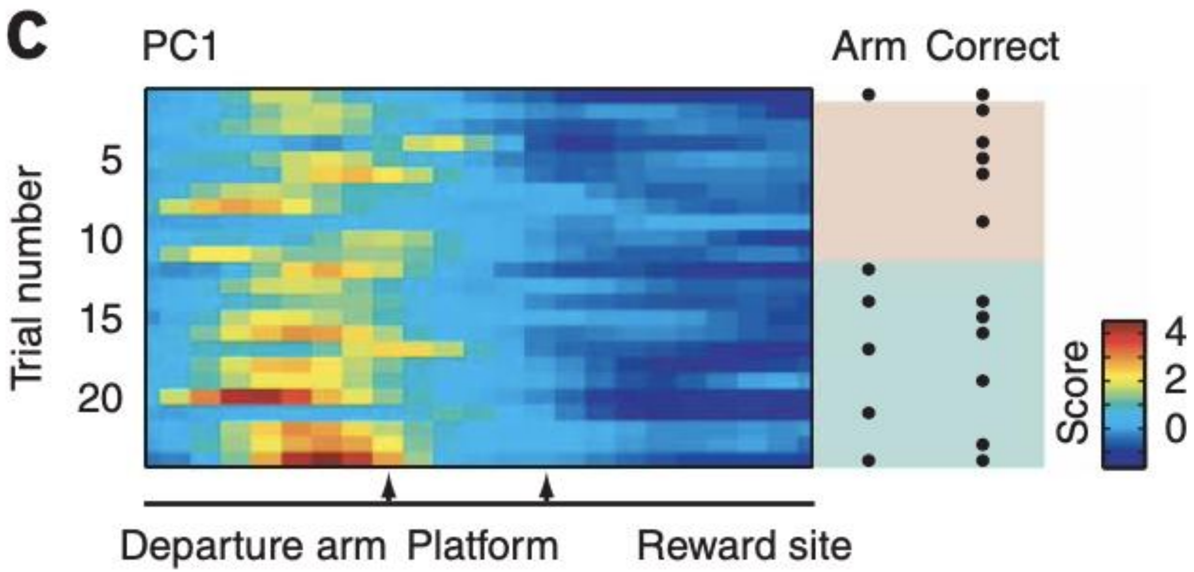
Cell assemblies hypothesis (Hebb 1949): Information is represented , replayed and stored through a group of neurons firing together

Principal Component as a Cell assembly



5 largest neuron component: 1-9-20-21-26 : a cell assembly

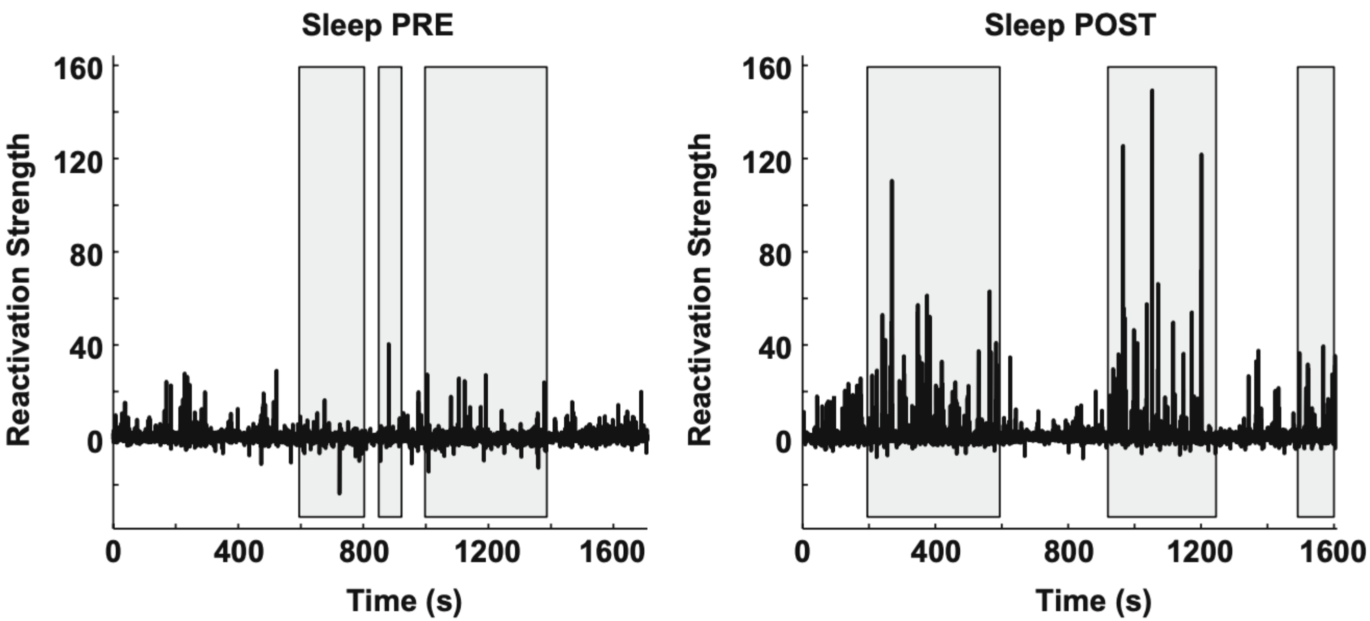
Activation of the first component during the task



Principal component reactivate at the beginning of the maze

Third componen reactivate t at the end of the maze

Reactivation of first component during sleep



Squared Projection of the activity along the first component minus diagonal contributions

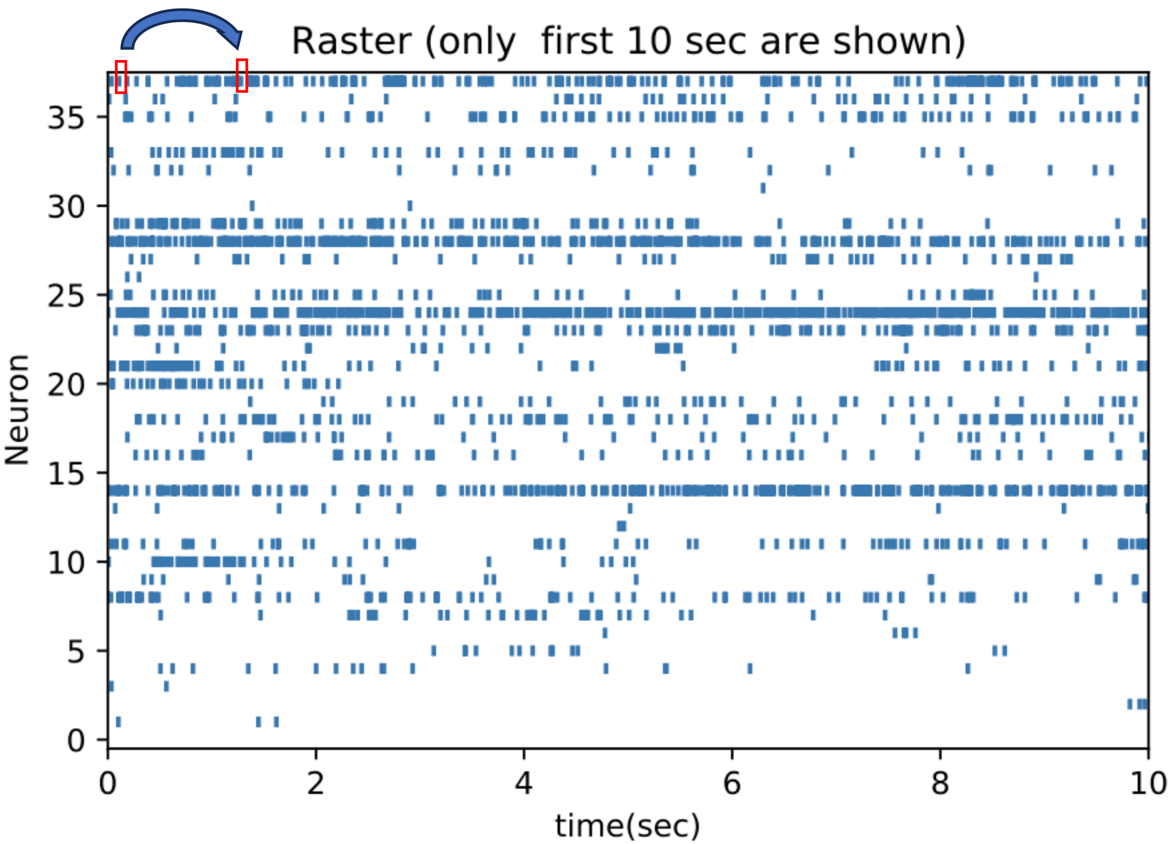
$$R_k^{match}(t_b) = \frac{1}{2}\left[\left(\sum_i y_{bi}^{match}\ \mathbf{e}_i^k\right)^2 - \sum_i (y_{bi}^{match}\ \mathbf{e}_i^k)^2\right]$$

Peyrache et al JC 2009

How many Principal Components to take?

Method 1: Generate uncorrelated variable from data

Destroy correlations by randomly reshuffling spike times for each neuron independently from the others



How many Principal Components to take?

Method 2: use Marcenko Pastur Spectrum for Random Covariance Matrices

Consider a multivariate distribution on uncorrelated random variables $<y_i>=0$, $<y_i y_j>=0$, $<y_i y_i>=1$

$$\widehat{C} = \begin{array}{c|c} 1 & 0 \\ \hline 0 & 1 \end{array}$$

Take M= measures

$$C_{ij} = \frac{1}{M} \sum_{b=1}^{M} y_{bi} y_{bj} \; .$$

$$\rho_{MP}(\lambda) = \frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{2\pi \, r \, \lambda} \, ,$$

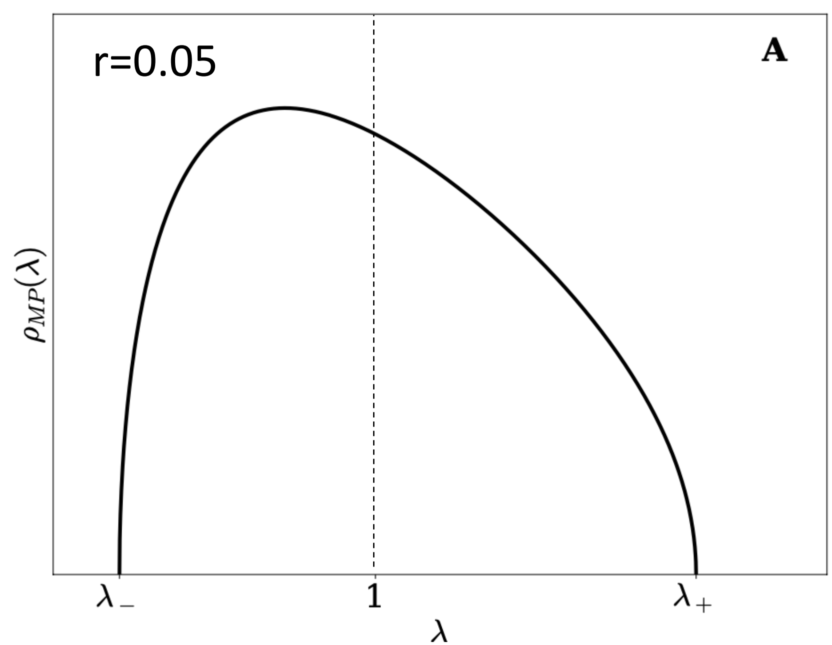
$r = \frac{N}{M}$

In our case

M=1.4 10^4
N=37
r= N/M= 2.6 10^{-3}

$\lambda_-=0.9$, $\lambda_+=1.1$,

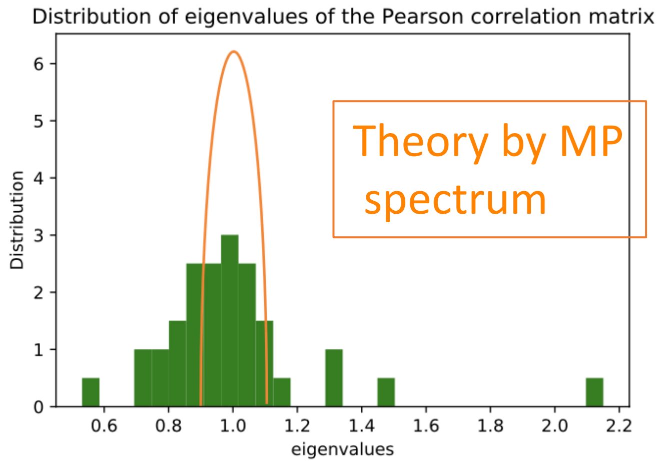
True case



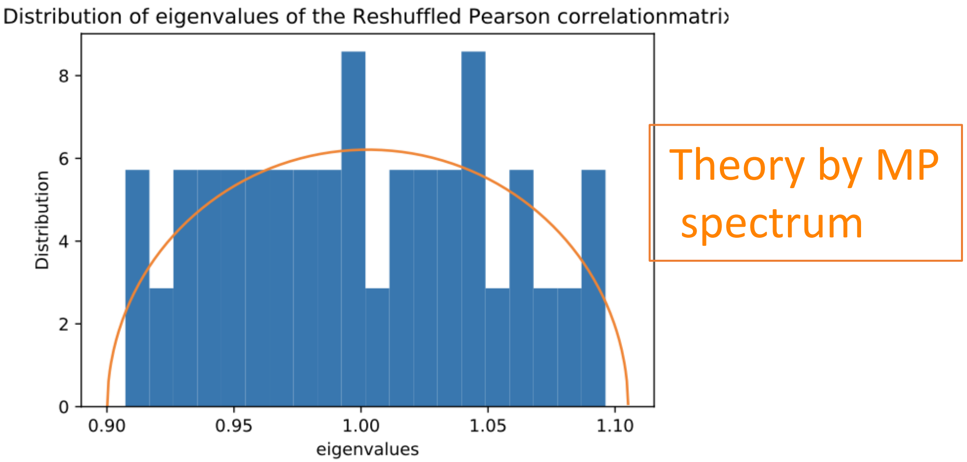
$$\lambda_- = (1 - \sqrt{r})^2 \qquad \lambda_+ = (1 + \sqrt{r})^2$$

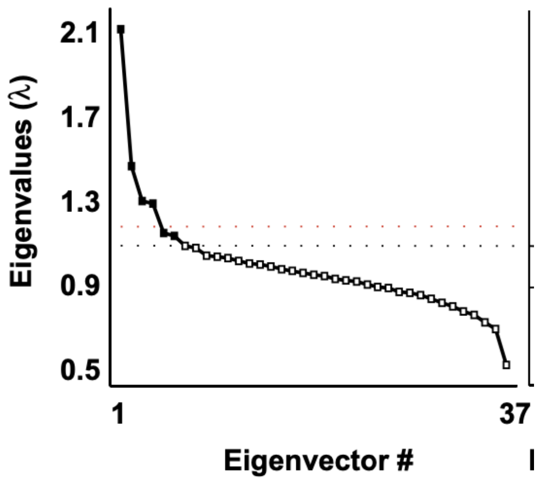
How many Principal Components to take?

From Data



From Reshuffled data





Book: From Statistical Physics to data Driven Modeling
S.C, R.Monasson,F.Zamponi, Oxford University Press 2022

https://github.com/StatPhys2DataDrivenModel/DDM_Book_Tutorials

Principal Components Analysis & Information theory

$$P(\mathbf{y}|\hat{T}) \propto e^{-\frac{1}{2}\sum_{i,j} y_i T_{ij} y_j}$$

Suppose we do not have access to **y** directly but we can only measure its projection along one direction **v** (with $\mathbf{v}^2=1$) :

$$x = \boldsymbol{v}^T \cdot \boldsymbol{y} + \xi$$

Where ξ is a measurement noise, which we assune to be Gaussian with zero mean and variance σ_ξ^2

How can we choose the direction of projection **v** so that the scalar variable x is maximally informative about the L-dimensional random variable **y** ?

Principal Components Analysis & Information theory

$$P(\mathbf{y}|\hat{T}) \propto e^{-\frac{1}{2}\sum_{i,j} y_i T_{ij} y_j}$$

Suppose we do not have access to **y** directly but we can only measure its projection along some direction **v** (with **v**^{2}=1) :

$$x = \boldsymbol{v}^T \cdot \boldsymbol{y} + \xi$$

Where ξ is a measurement noise, which we assune to be Gaussian with zero mean and variance σ_ξ^2

How can we choose the direction of projection **v** so that the scalar variable x is maximally informative about the L-dimensional random variable **y** ?

One can compute the **Mutual Information** between x and **y** (doable as x is a linear combination of gaussian variables and then itself a gaussian), and show that it is maximal along the Principal eigenvector of $\widehat{C}$

Application of PCA to facial recognition: EigenFace

Image is decomposed on pixels (128 X 128: 2^{14} dimensions)
and gray levels (2^8) at each position: $y(\mathbf{x})$ $x=[x_1, x_2]$ (or also by a fourier coefficient at each pixel)
M=115 Pictues

Average face:

[Sirovich & Kirby 1986 J. OptcSoc Am]

Application of PCA to facial recognition: EigenFace

Image is decomposed on pixels (128 X128: 2^{14} dimensions)
and gray levels (2^8) at each position: y(**x**). $x=[x_1, x_2]$ (or also by a fourier coefficient at each pixel)
M=115 Pictues

Extract the principal component analysis of the covaraince of the ensemble of faces

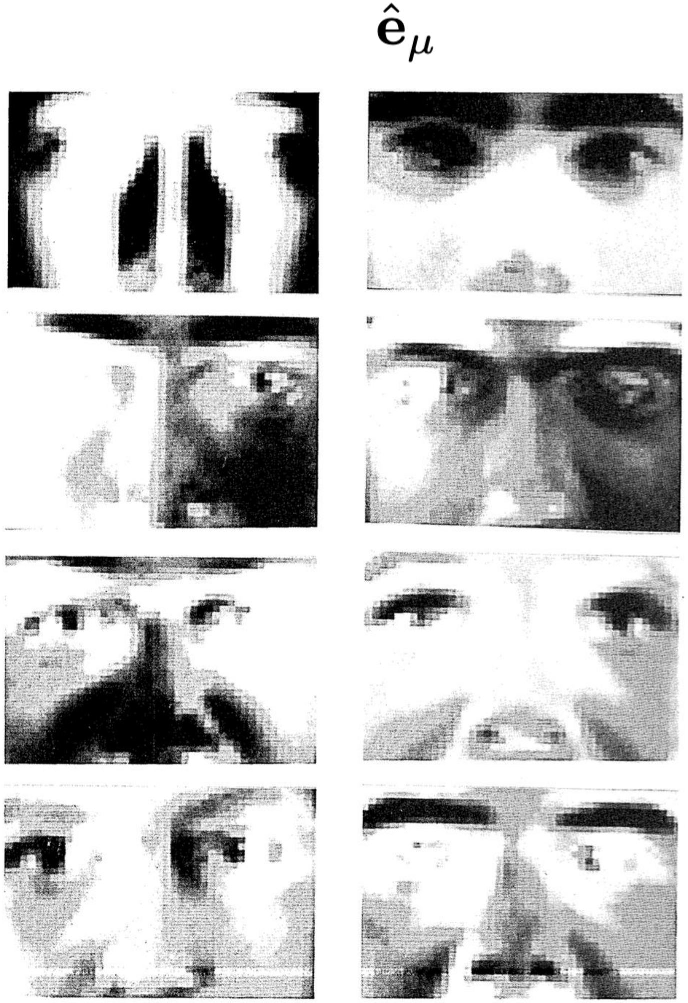


Fig. 4. First eight eigenpictures starting at upper left, moving to the right, and ending at lower right.

[Sirovich & Kirby 1986 J. OptcSoc Am]

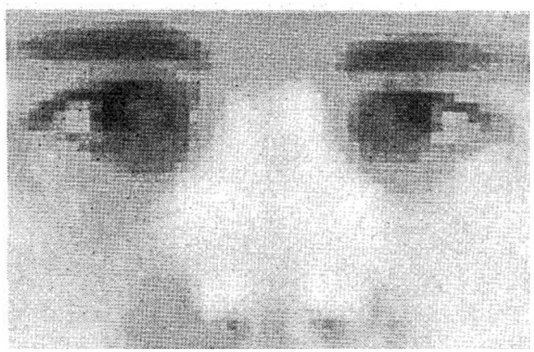
Application of PCA to facial recognition: EigenFace

Image is decomposed on pixels (128 X128: 2^{14} (=16384) dimensions)
and gray levels (2^8) at each position: y(**x**). x=[x_1 ,x_2] (or also by a fourier coefficient at each pixel)
M=115 Pictues

$$\boldsymbol{y} = \sum_{\mu} \sqrt{\lambda_{\mu}}\, z_{\mu} \mathbf{e}_{\mu}$$

Charecterize a face through its projections on the principal components.

Real image



Reconstructed images

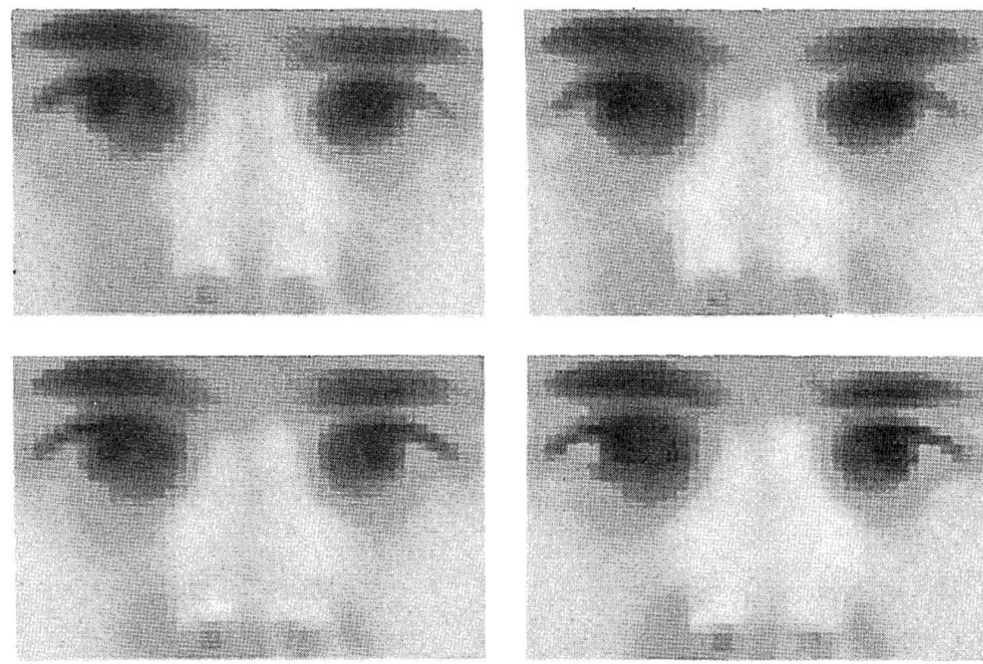


Fig. 5. Approximation to the exact picture (middle panel of Fig. 3) using 10, 20, 30, and 40 eigenpictures.

[Sirovich & Kirby 1986 J. OptcSoc Am]
[Turk & Pentland 1991 J. Cognitive Neuroscience]

From 16000 to 10-40 dimensions

Take Home Message

- It is difficult to interpret data in an high dimensional space

- PCA Reduce the dimensionality of the data, by projecting them in the directions of their maximal variability

- PCA is a powerfull method in many fields of cognitive science

- Models from theory of random matrices to decide how many component to take .

Limitations of Principal Components Analysis

1. It is not the optimal way to find cell assembly: an arbitrary threshold to select Neurons participating in it.
2. One neuron may participate to several cell assembly.

1. It can only perform linear transformations

Beyond PCA

1. Sparse PCA : approximation of the principal vector, with many component to zero

1. Independent Component Analysis, enforce separations of neurons in different Components

3. Kernel methods -> non linear transformation