Multivariate data analysis

Christophe Bécavin

PhD course

January 2018

Institut Pasteur C3BI

Unsupervised/Exploratory vs Supervised Analysis

- In the supervised methods, we observe both a set of features/variables (e.g. gene expression) for each object, as well as a response or outcome variable (e.g. metastasis information or survival information of the patients).

- The goal was then to predict the response using the variables (e.g. which genes predict best or are associated with the survival of the patients).

- Here we instead focus on exploratory analysis, we are not interested (yet) in prediction.

Goal of exploratory analysis

- Discover interesting things about the measurements.

Is there an informative way to visualize the data?

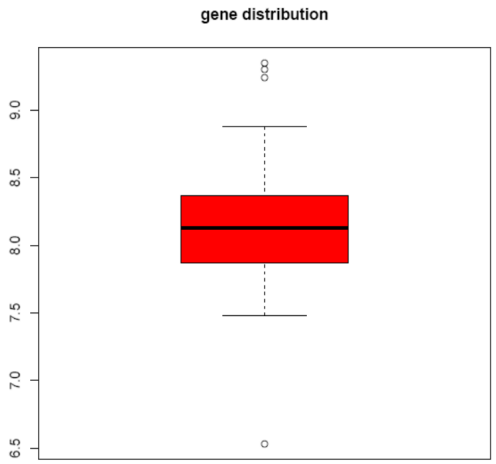
Can we discover subgroups among the variables or among the observations?

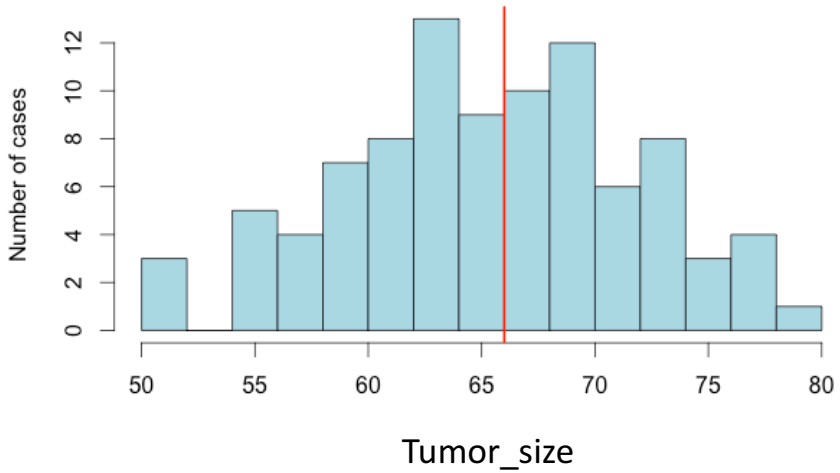
Can we highlight information not necessarily linked with the phenomenon under investigation?

- We will study two methods:
  - *principal component analysis*, a tool used for data visualization or data pre-processing before supervised techniques are applied;
  - *clustering*, a broad class of methods for discovering unknown subgroups in data.

Exploratory analysis: multivariate vs univariate analysis.

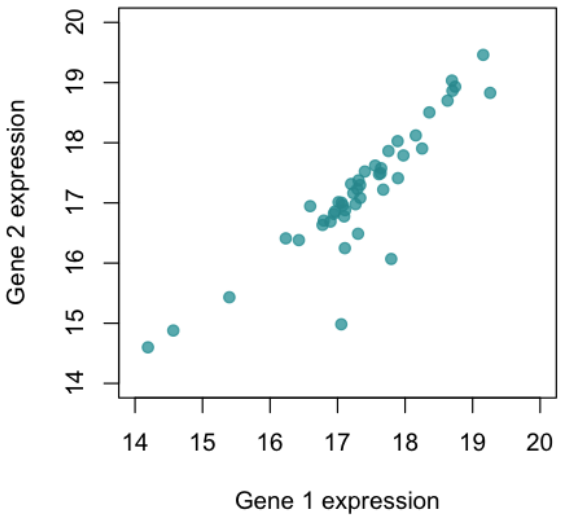
- Univariate: describe each variable separately.

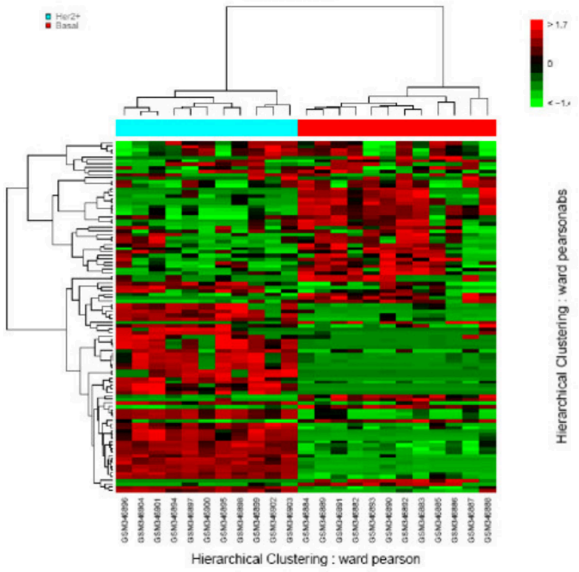


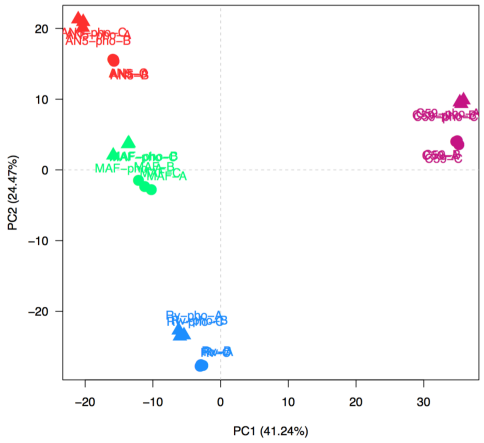


Exploratory analysis: multivariate vs univariate analysis.

- Multivariate: Simultaneous description of more than one variable.







Institut Pasteur | C3BI

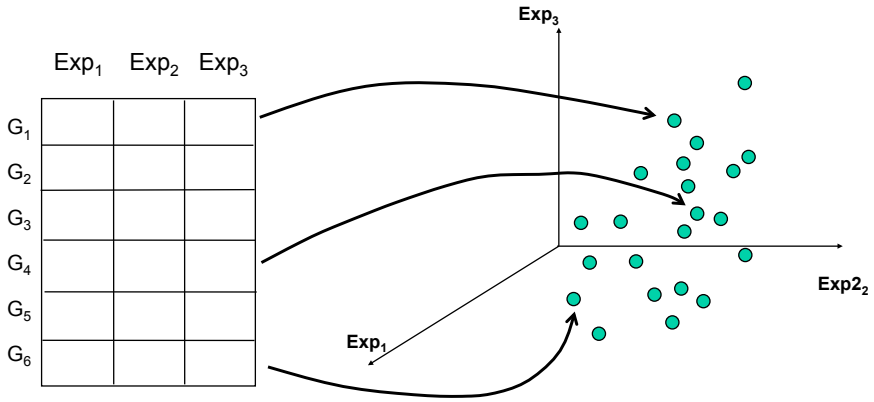
Principal Component Analysis

PCA: Introduction

- PCA is a descriptive method.
- Input: quantitative data table of *p* variables and *n* individuals.
- PCA reduces the dimensionality of the data while retaining most of the variation in the data set.
- Allows to investigate the structure of the data: Are there outliers ? Are there groups of samples? Which variables are correlated between each other?

Dimensionality reduction

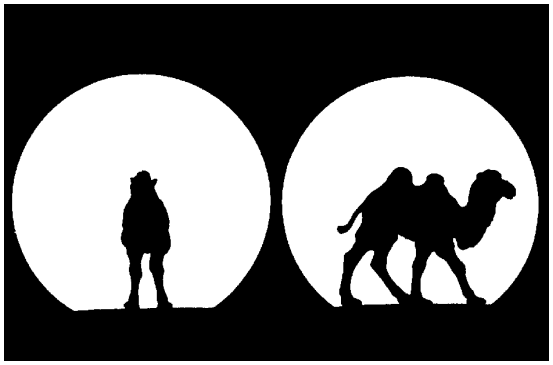
- If an individual is described by only two or three dimensions, easy to represent.



- Beyond 3 dimensions, impossible to represent or get a mental representation of the data.
- In life sciences: number of variables >> number of samples (thousands of genes, millions of SNPs,…)
- Such high-dimensionality makes visualization of samples difficult and limits simple exploration of the data.

Dimensionality reduction

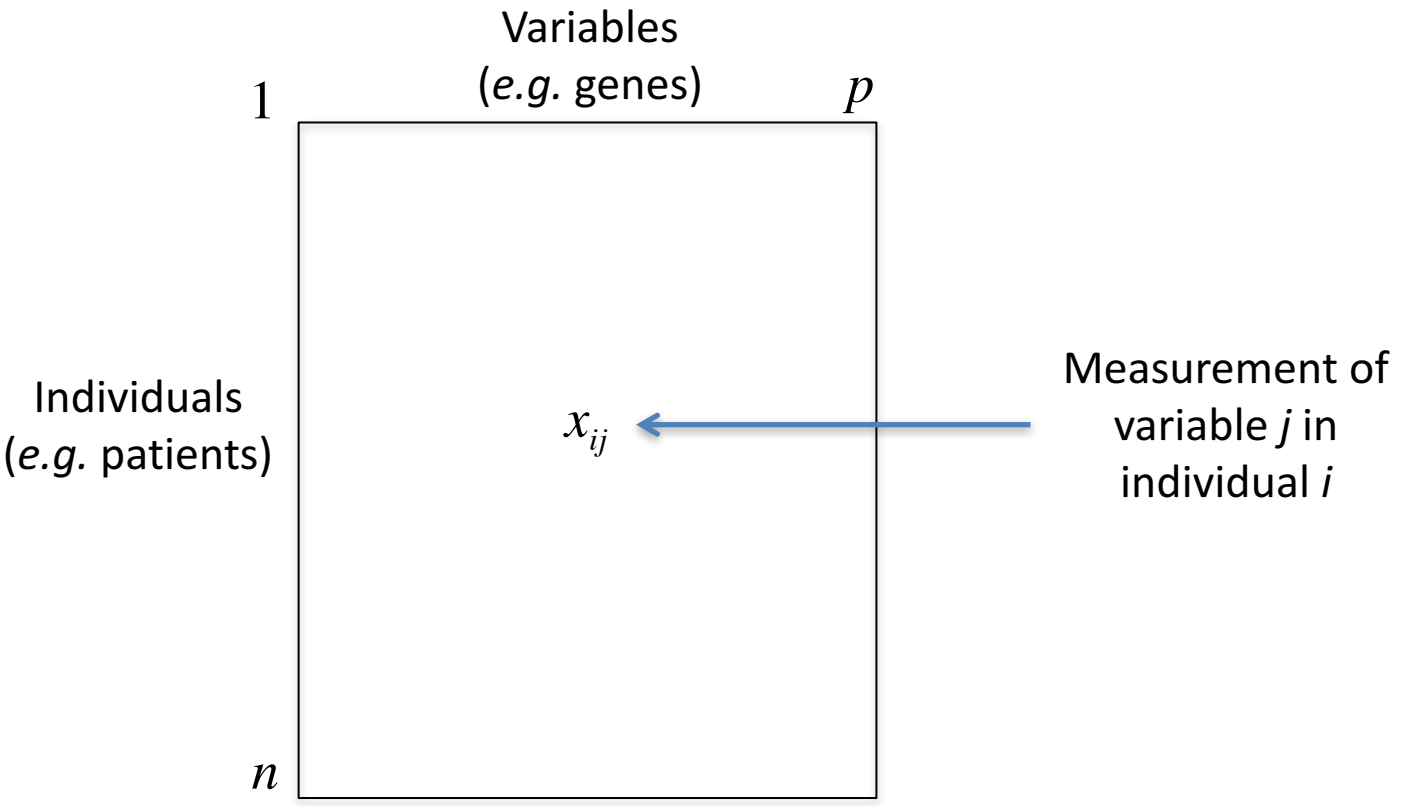
- We naturally project a 3-dimensional space onto a 2-dimensional space



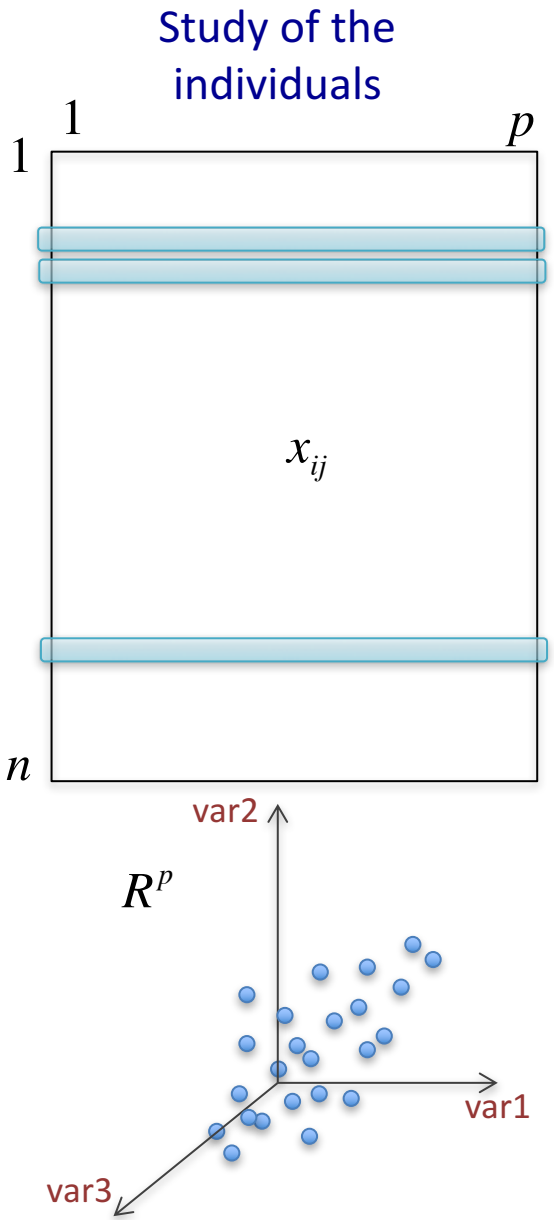
Camel vs dromedary?

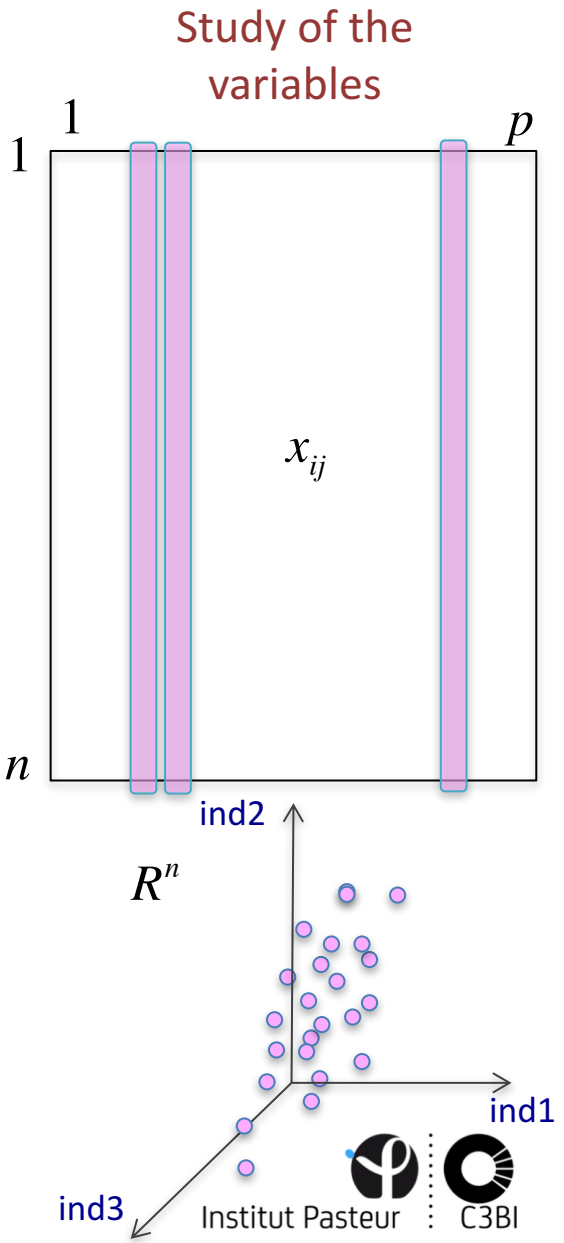
- Depending on the viewpoint, the retained information won't be the same.
- PCA offers one viewpoint on the data.

Data table


Variables
(*e.g.* genes)
1
p
Individuals
(*e.g.* patients)
x_{ij}
Measurement of variable *j* in individual *i*
n

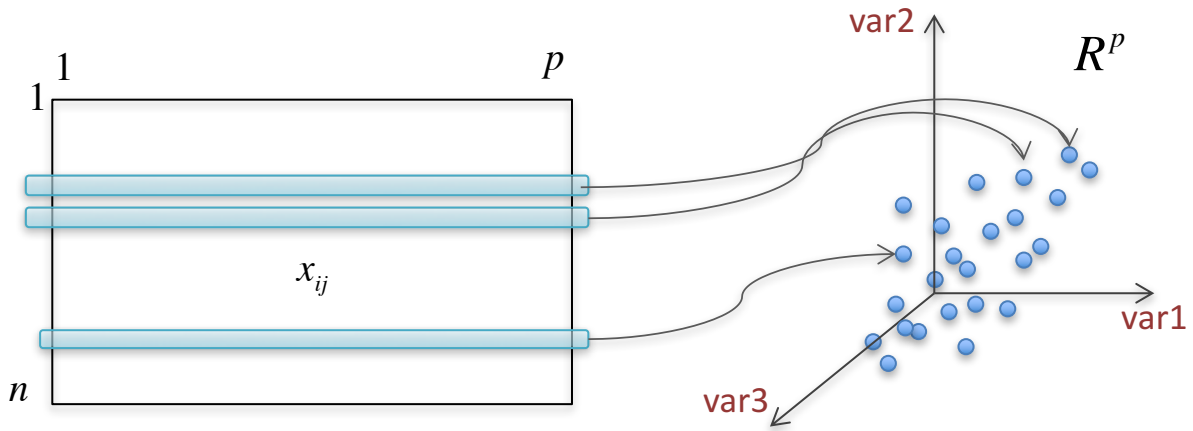
Data table





Institut Pasteur C3BI

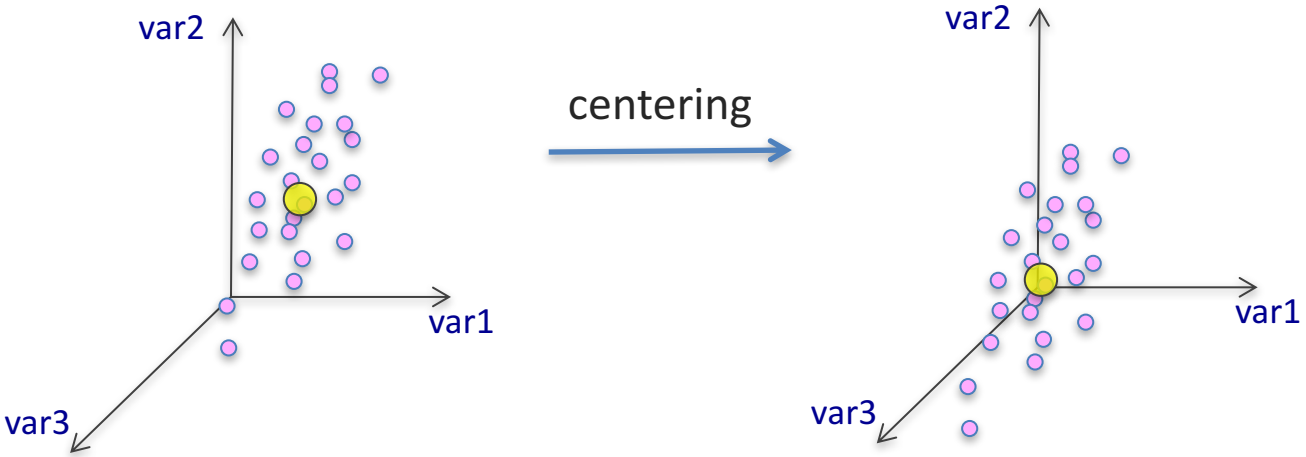
Study of the individuals



Study of the individuals = study of the shape of the Cloud of data points

Preprocessing before PCA

Mean-centering



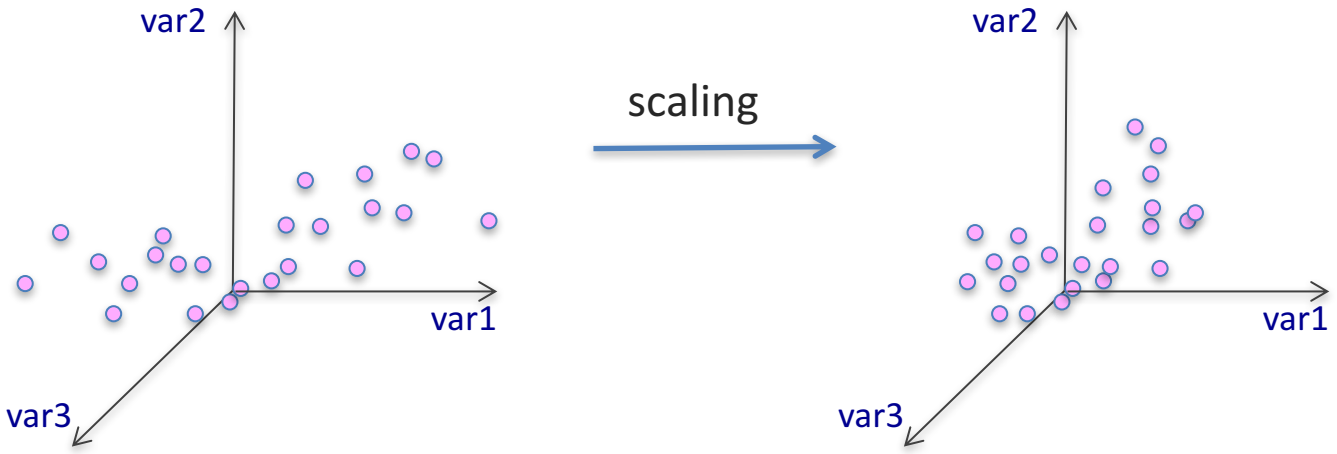
The vector of variable averages (e.g gene expression averages) is interpretable as a point (yellow here) in space. The mean-centering procedure corresponds to moving the origin of the coordinate system to coincide with the average point.

Mean-centering does not modify the shape of the data cloud.

Preprocessing before PCA

Scaling / Standardization

- Variables often have different numerical ranges.
- A variable with a larger range has a larger variance than a variable with a smaller range and is more likely to influence the PCA than a variable with a small variance.
- Unit variance scaling: each variable vector is divided by its standard deviation,



Institut Pasteur C3BI

Preprocessing before PCA

Unit variance scaling and mean-centering

$$x_{ij} \rightarrow \frac{x_{ij} - \overline{x}_i}{s_i}$$

Substract the mean, and divide by the standard deviation

With:

$$\overline{x}_i = \frac{1}{n}\sum_{j=1}^{n} x_{ij}$$

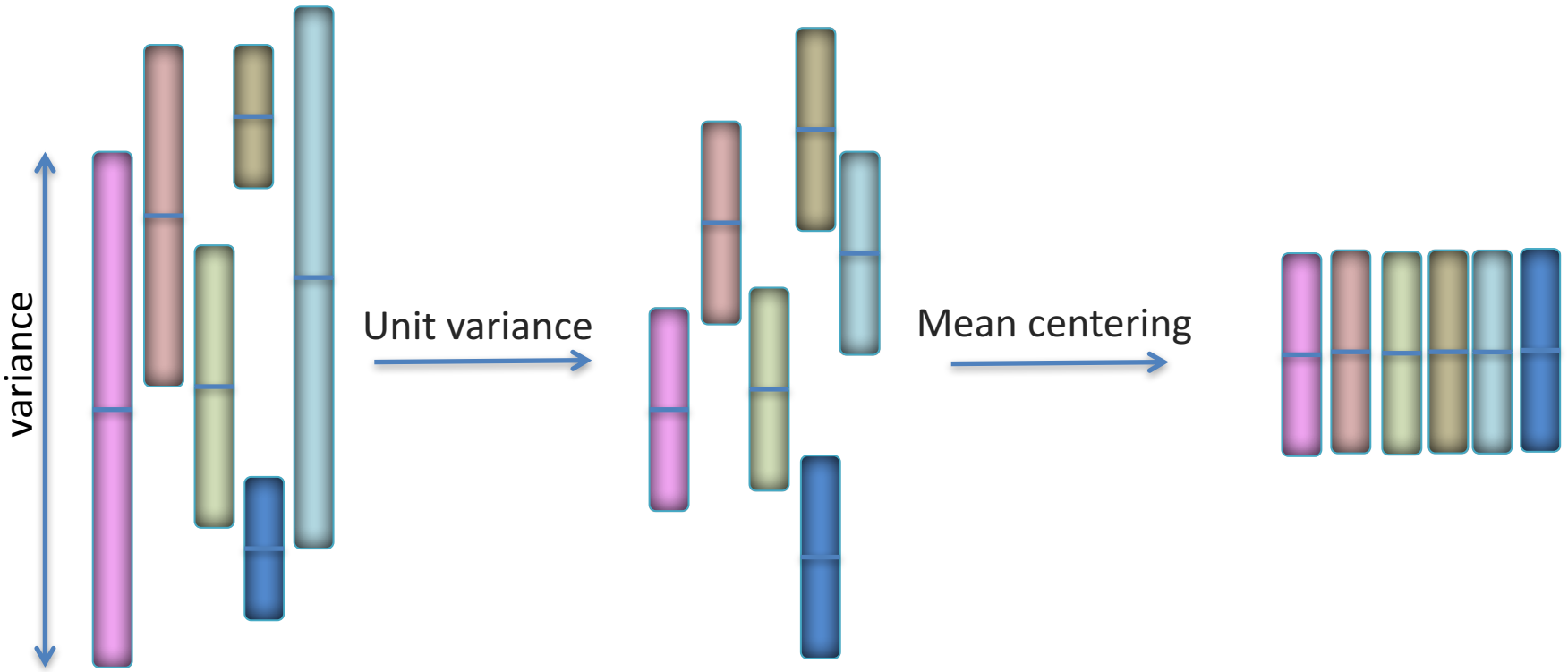
is the mean of the variables for individual *i* (e.g. mean of the vector of gene expression profile for sample *i*)

$$s_i = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(x_{ij} - \overline{x}_i\right)^2}$$

is the standard deviation of the variables for individual *i* (e.g. standard deviation of the vector of gene expression profile for sample *i*)

Preprocessing before PCA

Unit variance scaling and mean-centering



Principal Component Analysis: details

- Goal: Find a sequence of linear combinations of the variables that have maximal variance, and are mutually uncorrelated.
- The first principal component of a set of variables is the normalized linear combination of the p variables that has the largest variance.
- The first component is:

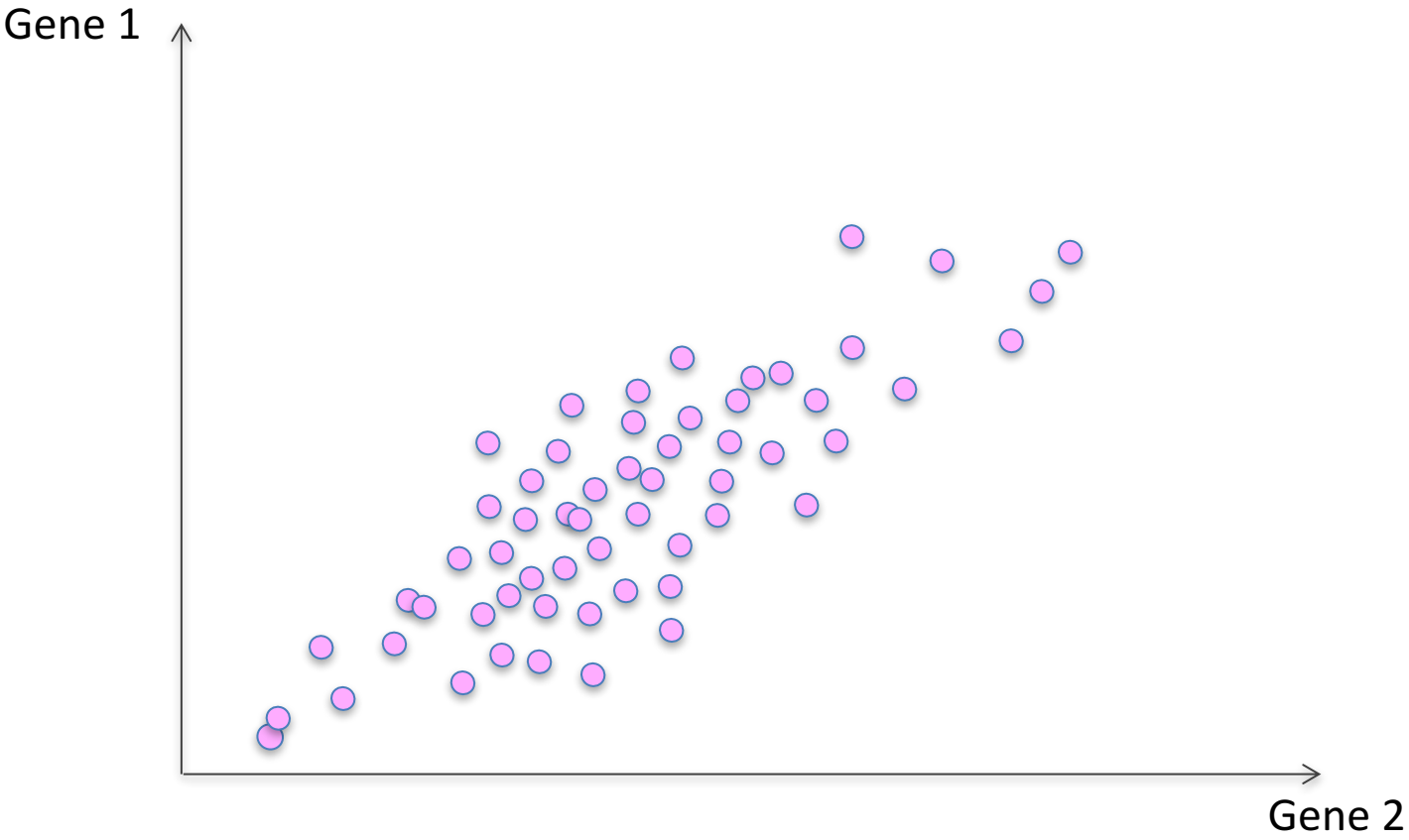
$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \ldots + \phi_{p1} X_p$$

Loadings of the PC (vector of length *p* the number of variables)

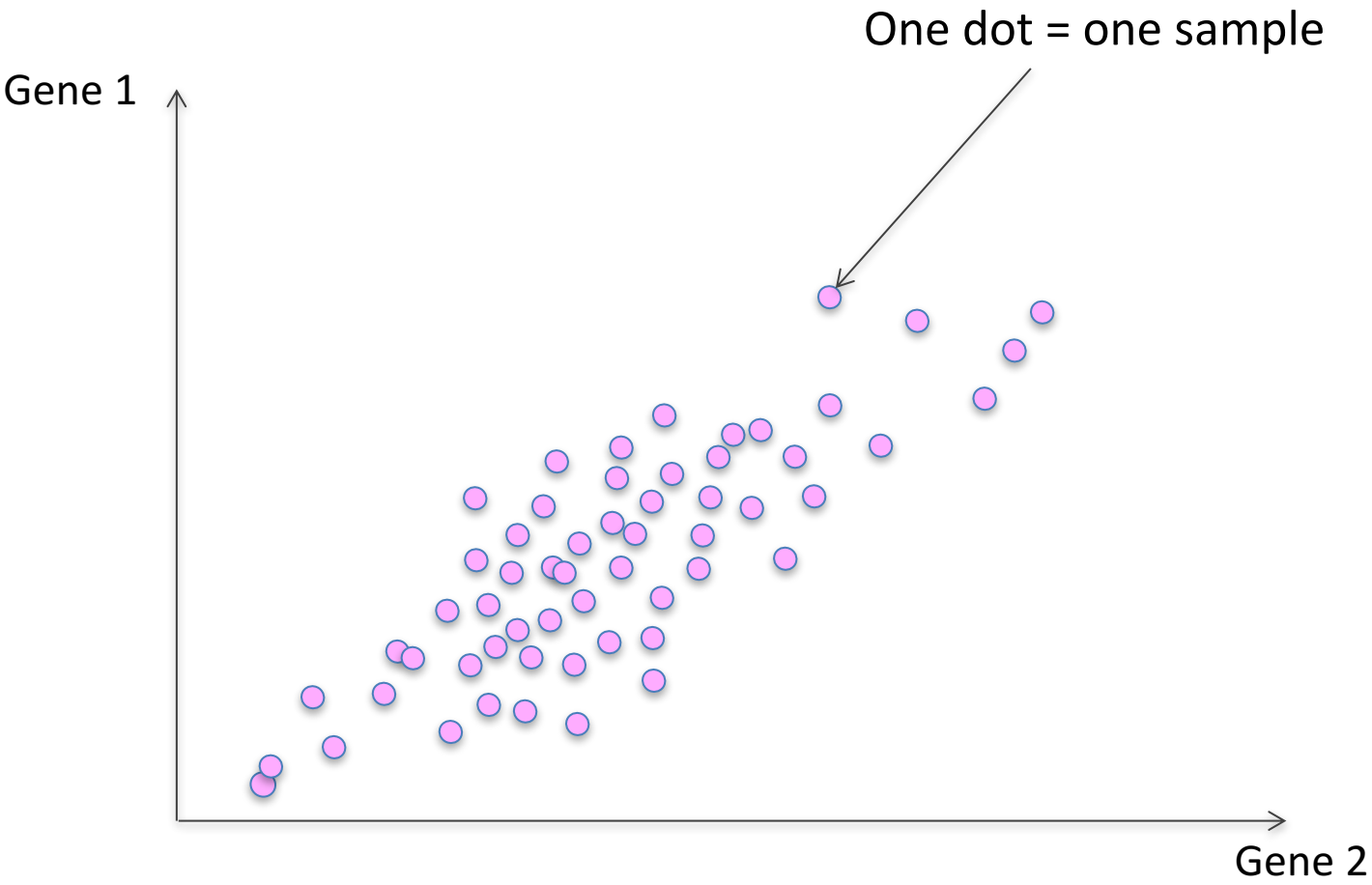
coordinates of the PC in the feature space

- There are at the most $\min(n-1, p)$ PCs.
- Constraint on the loadings: $\sum_{j=1}^{p} \phi_{j1}^2 = 1$ to avoid arbitrarily large variance.
- The loading vector $\phi_1 = (\phi_{11}, \phi_{21}, \ldots, \phi_{p1})^T$ defines a direction in the feature space.

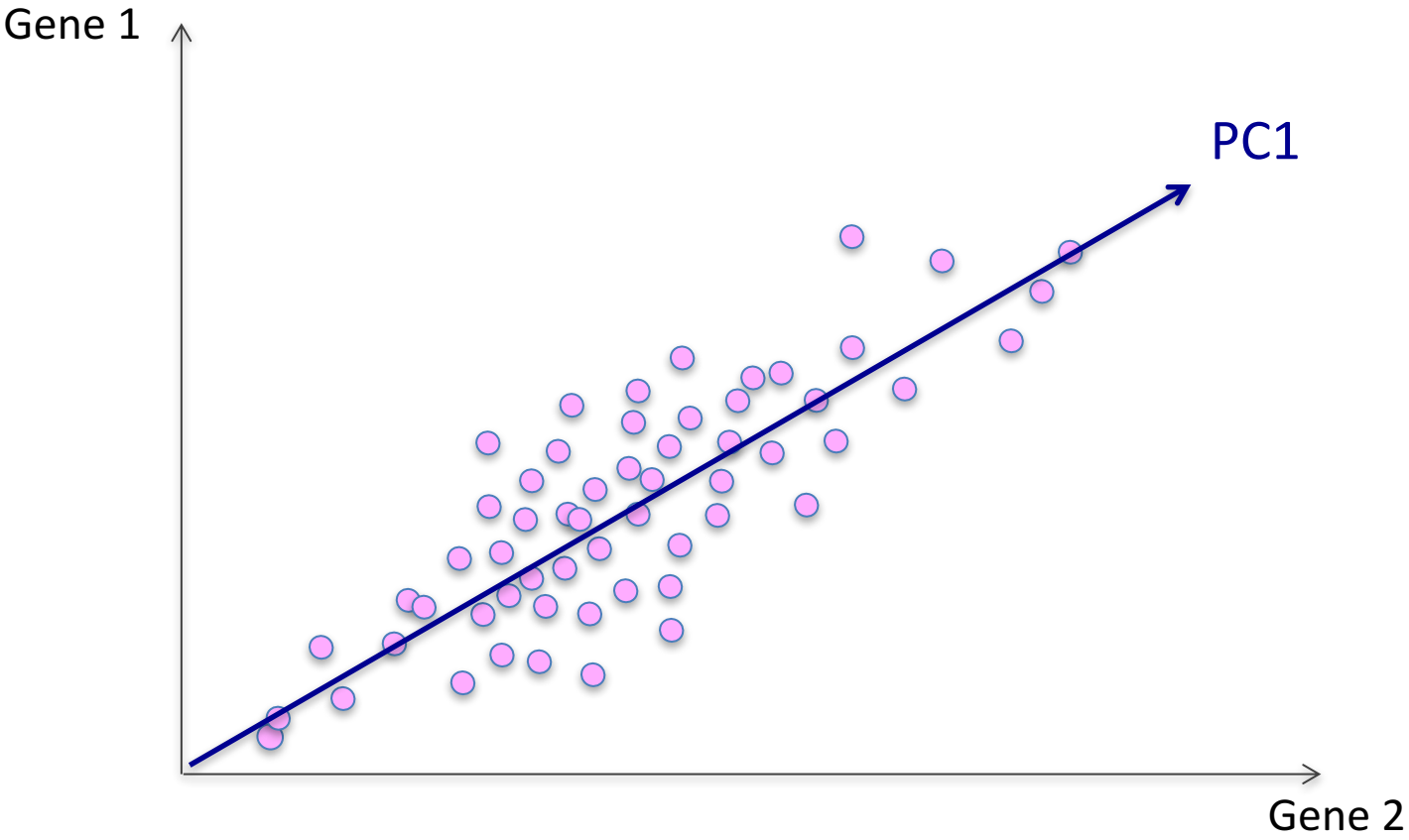
PCA: example in two dimensions



PCA: example in two dimensions



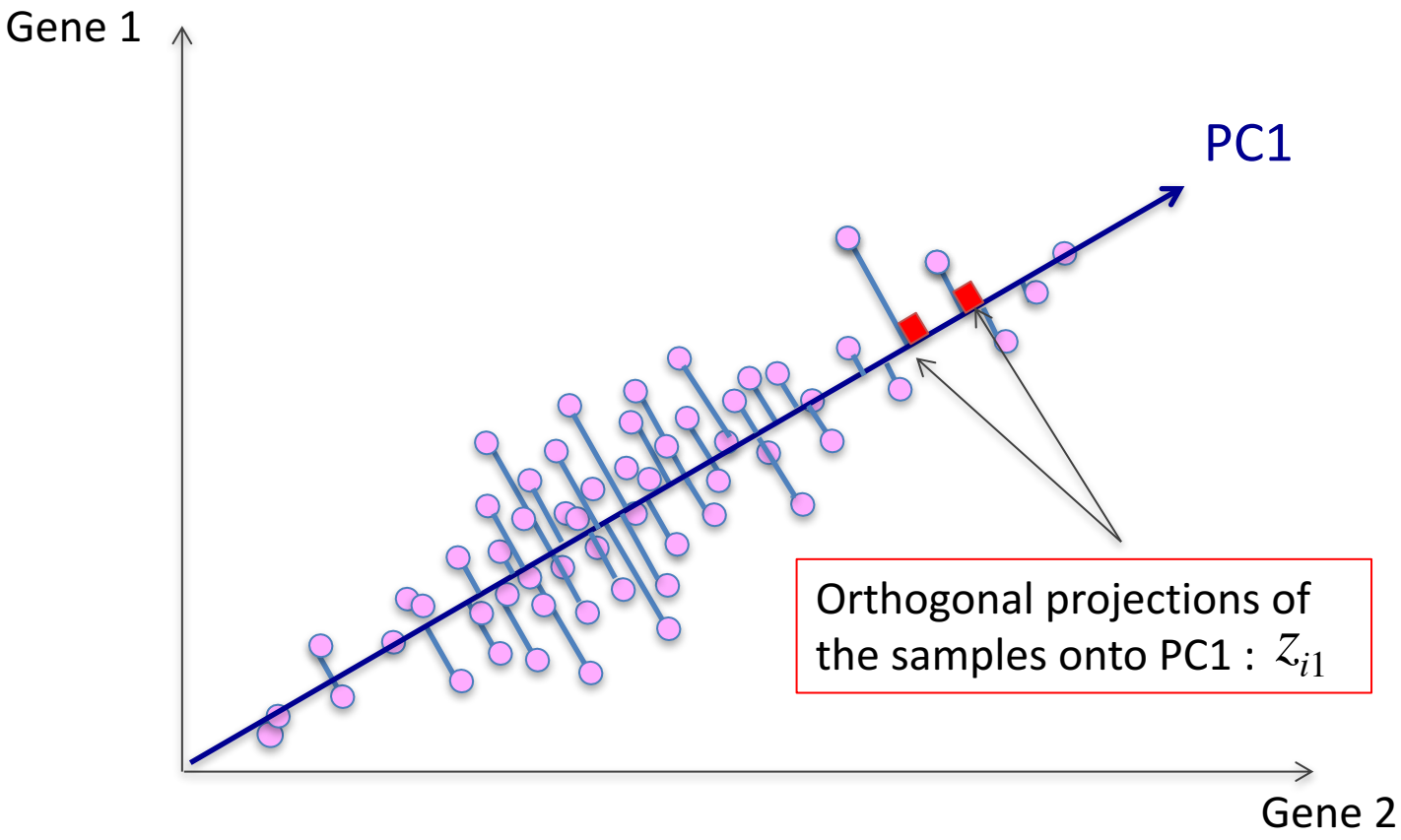
PCA: example in two dimensions



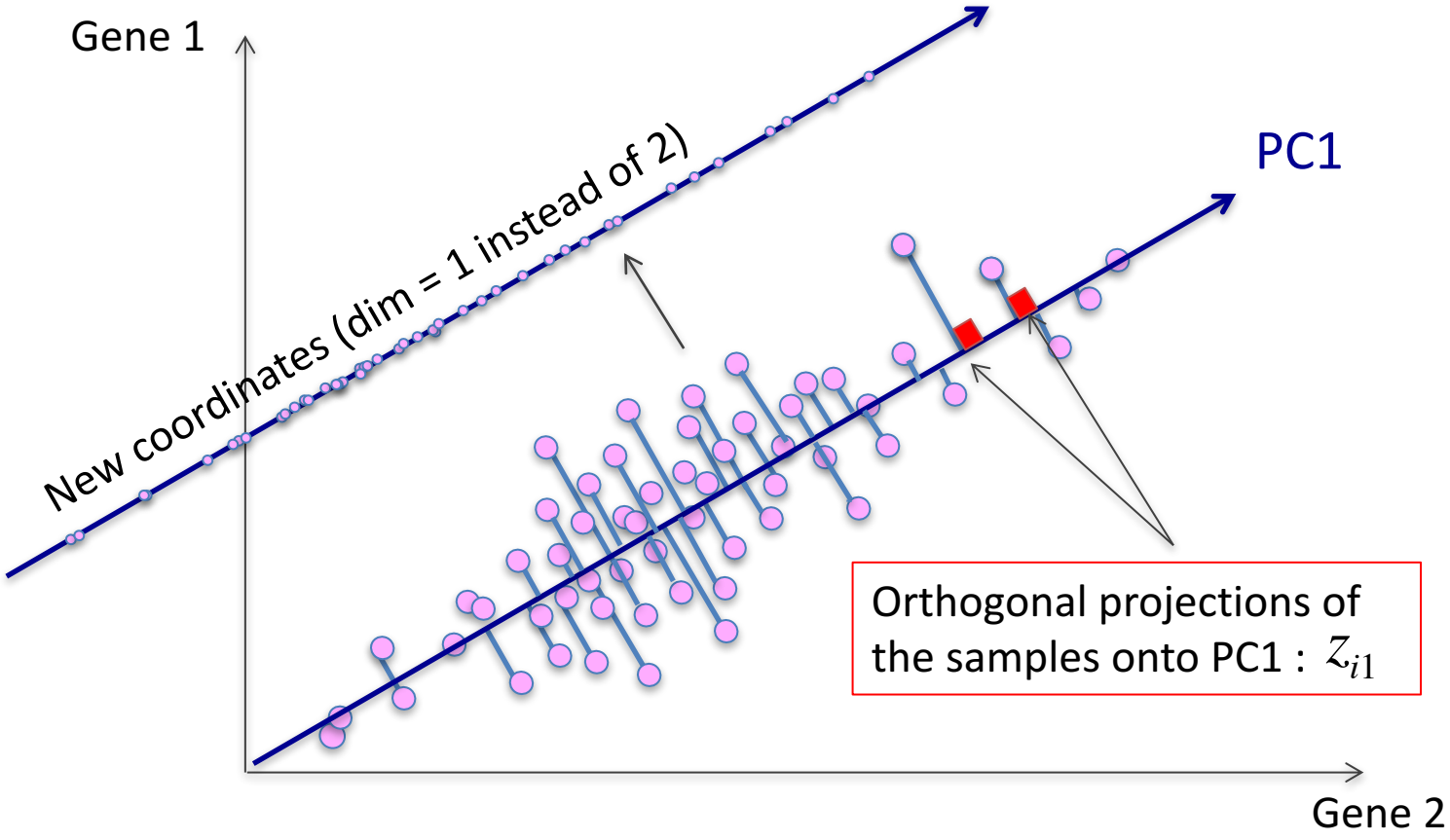
Find the direction(s)/space(s) with the property that the squared distance of the points to their orthogonal projection onto the space is minimized.
« The best line that fits the data»

Institut Pasteur C3BI

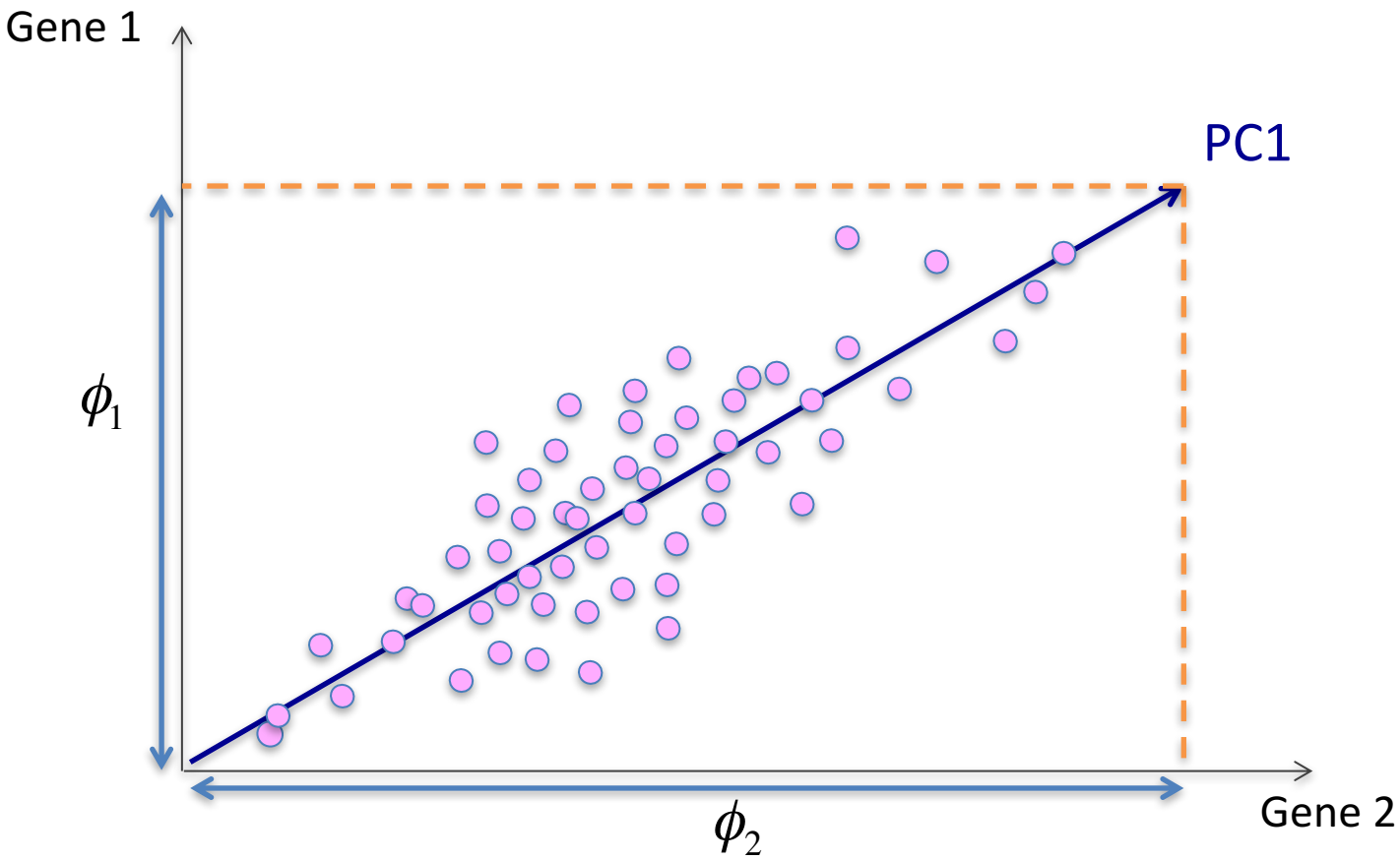
PCA: example in two dimensions



PCA: example in two dimensions

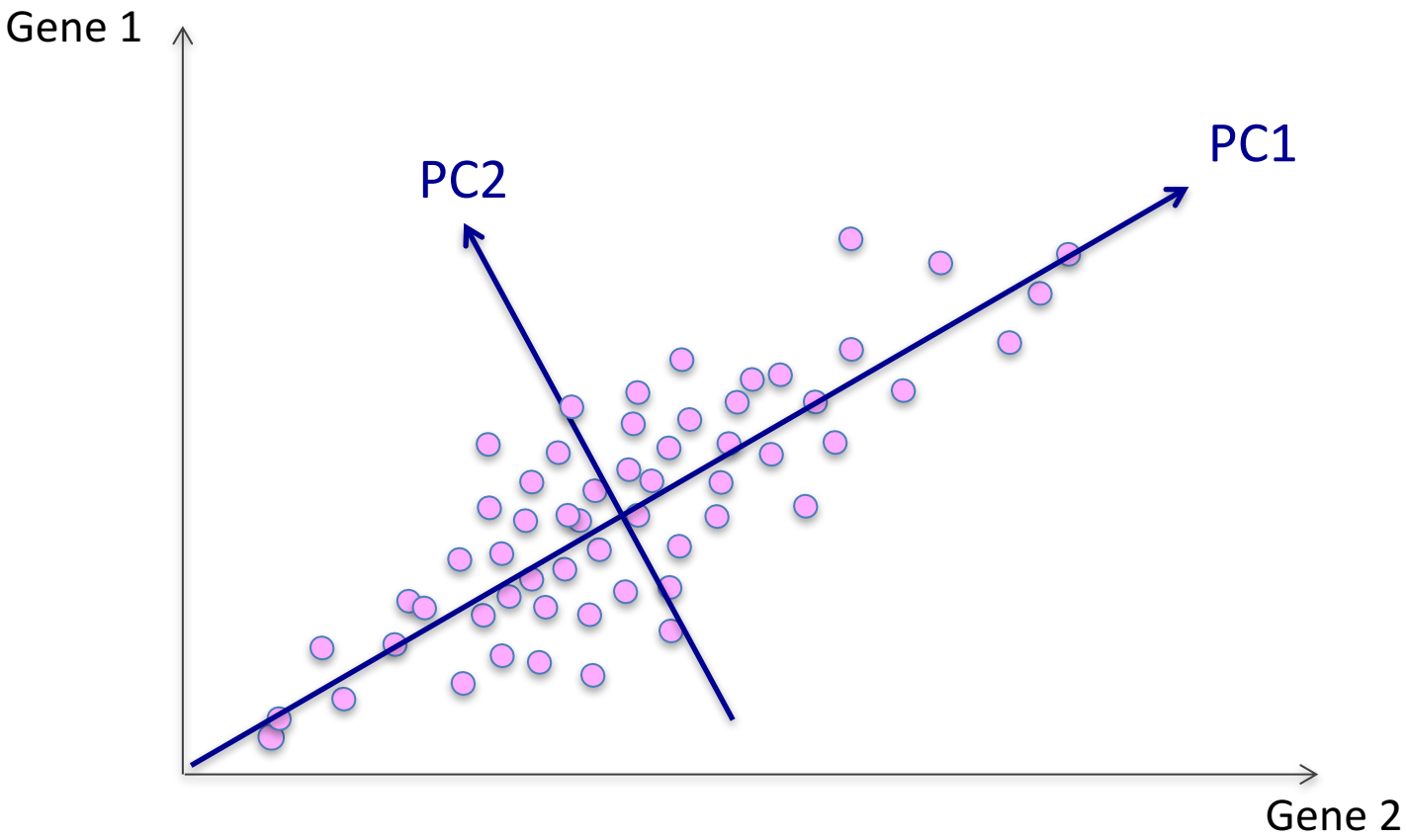


PCA: example in two dimensions



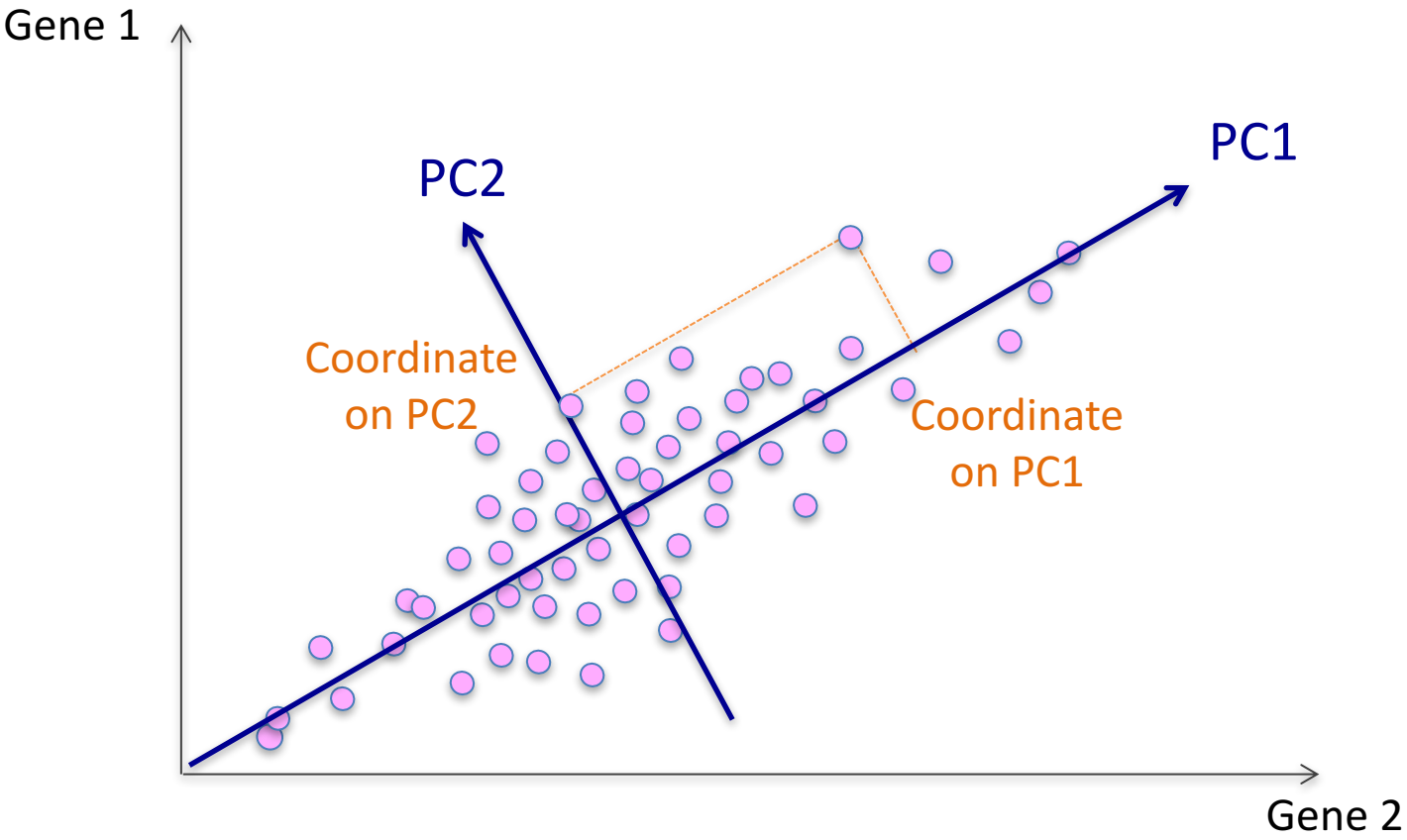
(PC1) $Z_1 = \phi_1 Gene_1 + \phi_2 Gene_2$

PCA: example in two dimensions



The second principal component is the linear combination of the variables that has maximal variance among all linear combinations that are orthogonal (uncorrelated) with PC1.

PCA: example in two dimensions



PCA in *R* with package *FactoMineR*

Load data and package

```
## Load example dataset
dt <- read.table(file="Gene_Expression.txt",header=TRUE,
row.names=1,sep="\t")

## One data.frame object has been loaded:
## RNASeq read count per gene, data restricted to 50 genes
head(dt)
dim(dt)


## PCA will be computed on the log2 read counts
dt[,3:ncol(dt)] <- log2(dt[,3:ncol(dt)])


## load the factoextra and FactoMineR packages
## (install first if necessary)
library(factoextra)
library(FactoMineR)
```

PCA in *R* with package *FactoMineR*

Perform PCA

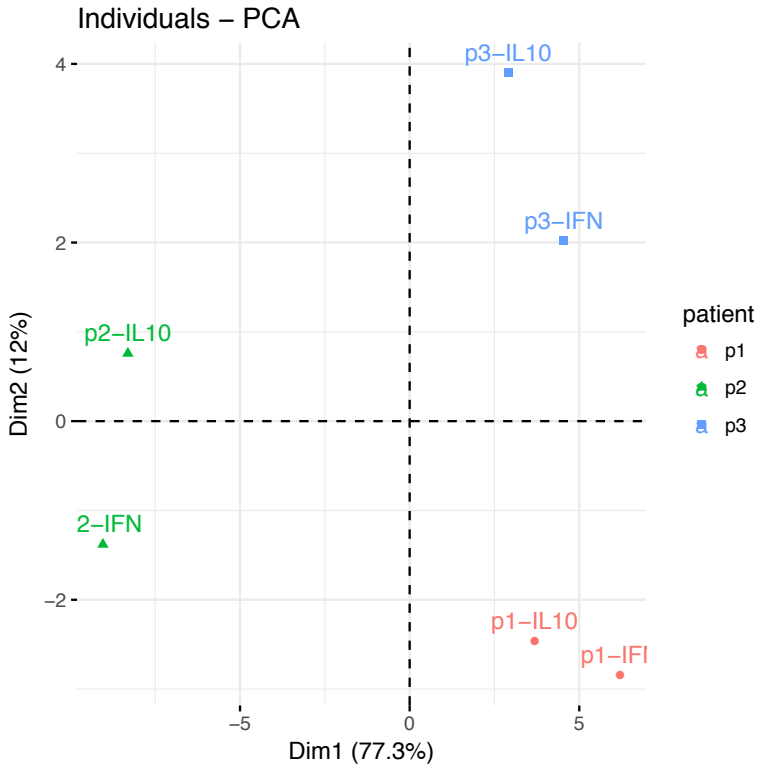
```
## Run the PCA
respca = PCA(dt, scale.unit=TRUE, graph=F, ncp=5, quali.sup=1:2)

# X: the data set used. Rows are individuals and columns are
    numeric variables.
# scale.unit: if TRUE, the data are scaled to unit variance
     before the analysis.
# ncp: number of PCs to keep.
# graph: if TRUE, the graphs are displayed.
```

PCA in *R* with package *FactoMineR*

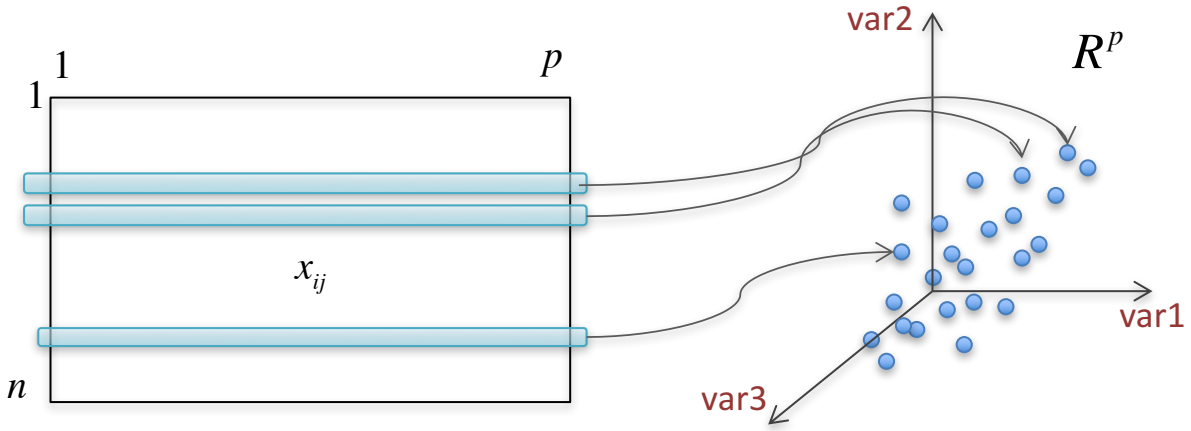
PCA plot of the subjects

```
## Plot of the samples onto the space spanned by the first and the second PCs.
## Color the data points according to the patient

fviz_pca_ind(respca,
             axes = c(1,2),
             habillage = 'patient',
             invisible = 'quali')
```

```
# Equivalent with the function plot.PCA (run faster for bigger datasets)
plot.PCA(respca, axes = c(1,2), choix = 'ind', habillage = 1,  invisible = 'quali')
```



PCA with p > 2 dimensions



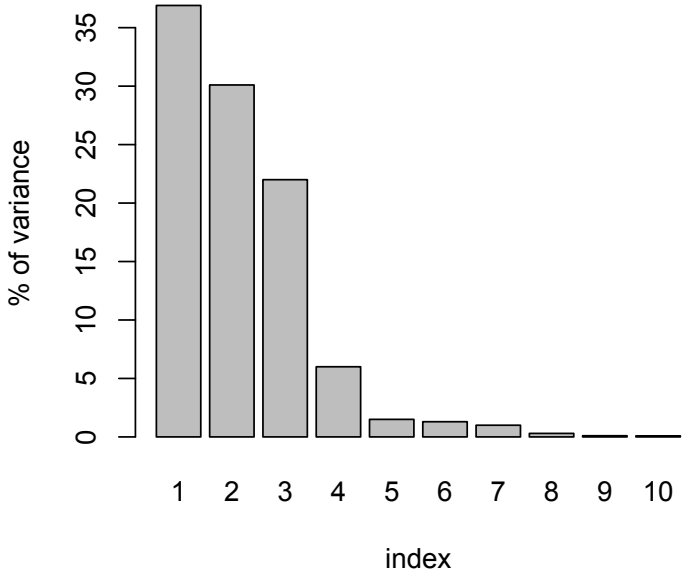
P dimensions = p-1 principal components (PCs)

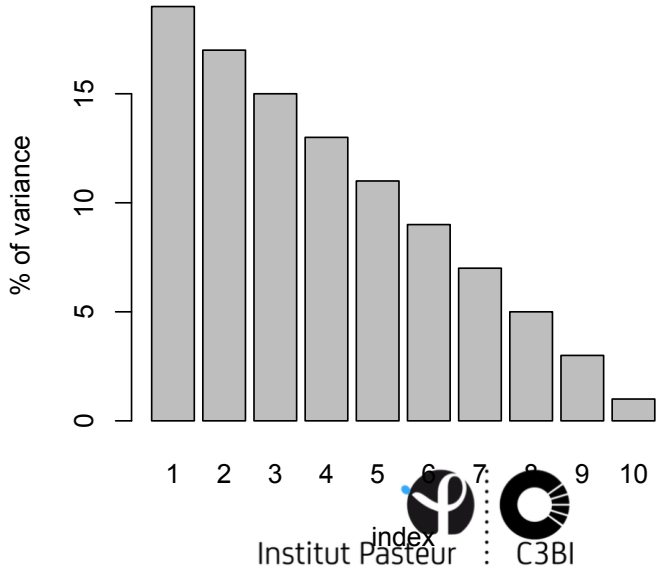
How many PCs?

- There are a couple ways of deciding how many principal components to use but **no correct answer**.
- Consider what is your reason of using PCA in the first place:
  - *How many do you need?* Sometimes PCA is used in a classification framework.
  - *How much variance do you want?* We could decide that we need a certain fraction of the total variance, and choose the number of PCs allowing to achieve it.
  - *How many eigenvalues are big?* i.e. *how many PCs capture large variance?*
- All these methods are arbitrary.

How many PCs?
Examples of methods

- Always useful to have a look at the plot of the fraction of total variance explained by each component.
- *Scree* graph (Cattell, 1966) : Look at the plot of the percentage of variance l_k against *k* and decide which value of k the defines an *'elbow'* in the graph (subjective). In practice, rarely easy to choose.





PCA in *R* with package *FactoMineR*

PCA output

```
## PCA output
respca
     **Results for the Principal Component Analysis (PCA)**
     The analysis was performed on 6 individuals, described by 50 variables
     *The results are available in the following objects:
      name                 description
     "$eig"               "eigenvalues"
     "$var"               "results for the variables"
     "$var$coord"         "coord. for the variables"
     "$var$cor"           "correlations variables - dimensions"
     "$var$cos2"          "cos2 for the variables"
     "$var$contrib"       "contributions of the variables"
     "$ind"               "results for the individuals"
     "$ind$coord"         "coord. for the individuals"
     "$ind$cos2"          "cos2 for the individuals"
     "$ind$contrib"       "contributions of the individuals"
     "$call"              "summary statistics"
     "$call$centre"       "mean of the variables"
     "$call$ecart.type" "standard error of the variables"
     "$call$row.w"        "weights for the individuals"
     "$call$col.w"        "weights for the variables"
```

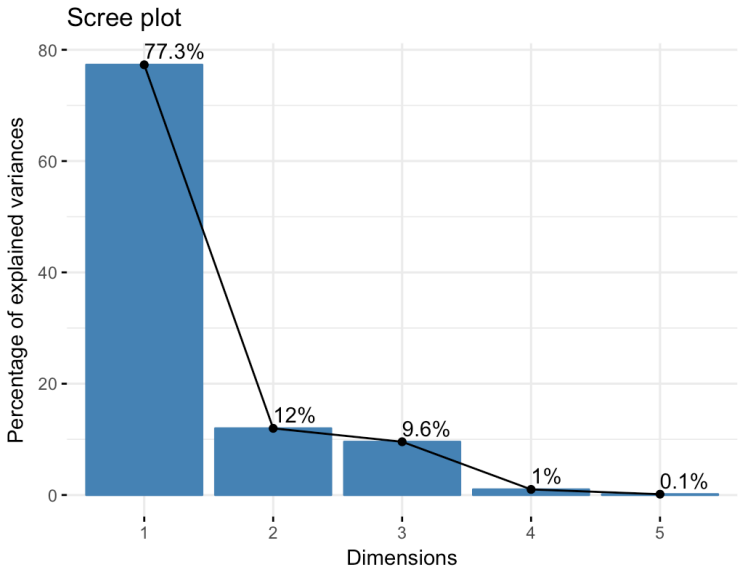
PCA in *R* with package *FactoMineR*

Study variances

```
## The attribute eig contains information relative to the variance
explained in each component.
eig = get_eig(respca)
```

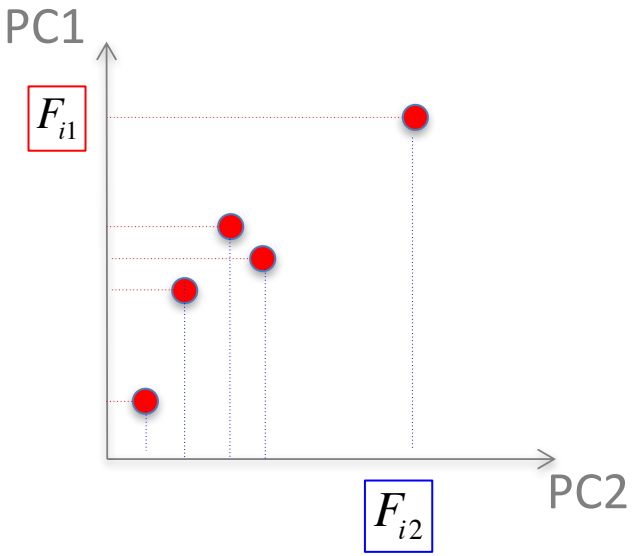
```
       eigenvalue variance.percent cumulative.variance.percent
Dim.1 38.65276031       77.3055206                    77.30552
Dim.2  5.98938312       11.9787662                    89.28429
Dim.3  4.78358701        9.5671740                    98.85146
Dim.4  0.50177984        1.0035597                    99.85502
Dim.5  0.07248971        0.1449794                   100.00000
```

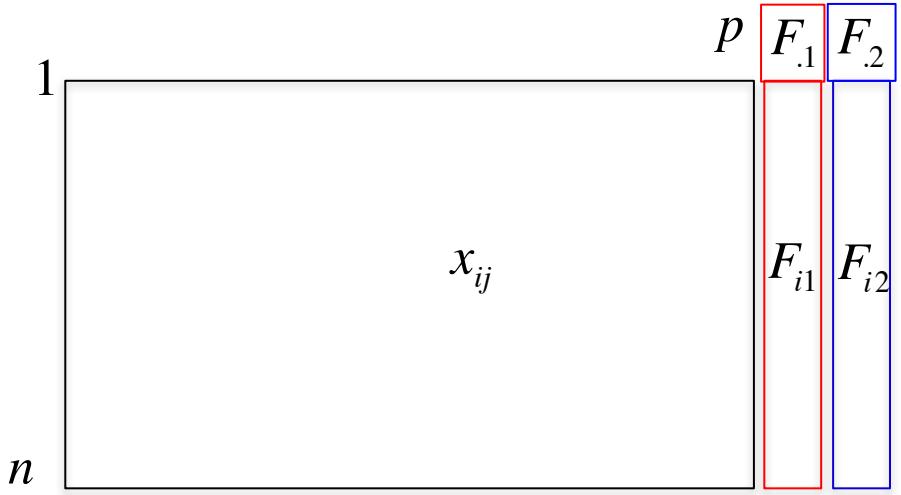
```
## Plot the % of variances explained by each PC
fviz_screeplot(respca, addlabels = TRUE)
```



Interpretation of the PCA plot of the individuals using the variables

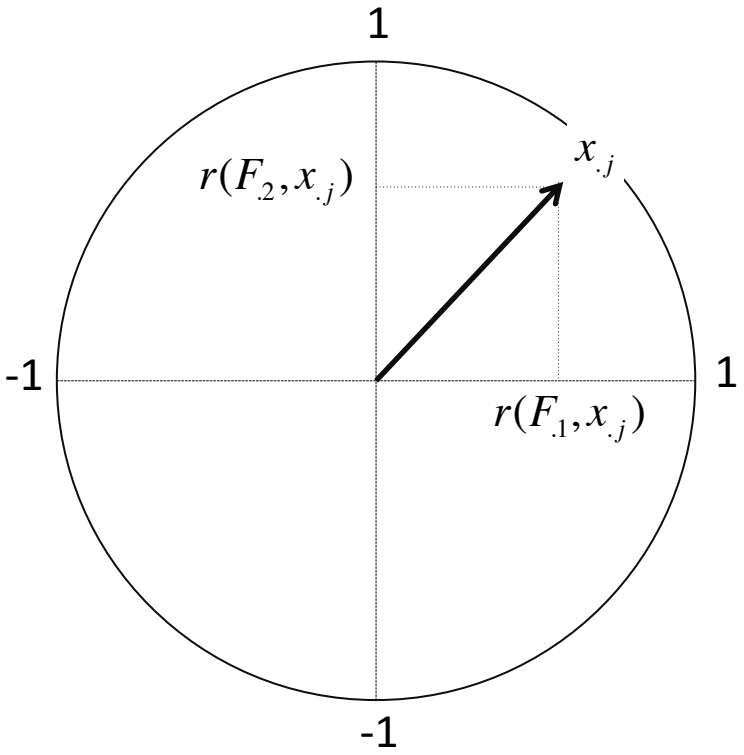
Let's consider the coordinates of the individuals on the PCs as variables.





Interpretation of the PCA plot of the individuals using the variables

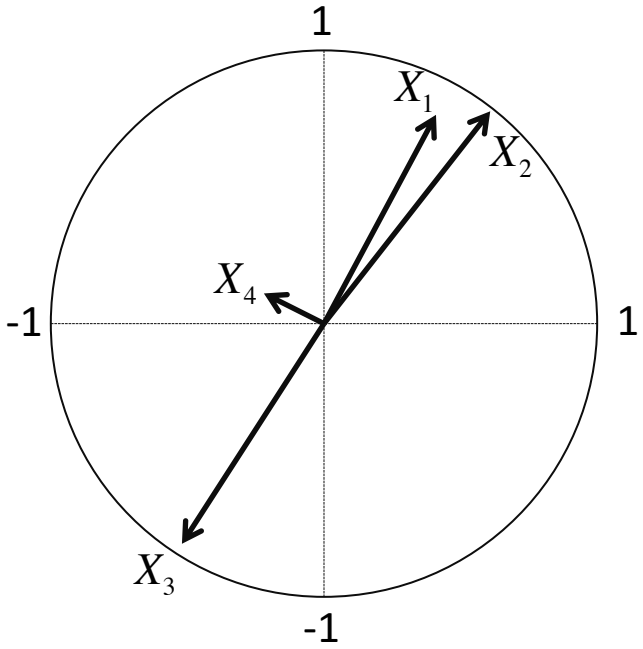
- We can study the correlation between the initial variables and the PCs.
- The correlation between a PC and a variable estimates the information they share.
- **Correlation circle:** The variables can be plotted as points in the component space using their correlations with the coordinates of the individuals on the PCs.



Institut Pasteur C3BI

Interpretation of the PCA plot of the individuals using the variables

- A point close to the circle indicates a variable that is well characterized by the represented PCs (X_1, X_2, X_3).
- A point close to the center indicates a variable whose properties are not captured/described by the represented PCs (X_4).
- Two points close to the circle and to each other indicate a positive correlation between the two variables (X_1, X_2).
- Two points close to the circle and in opposite direction indicates a negative correlation between the two variables (X_1, X_3 or X_2, X_3).



Institut Pasteur C3BI

PCA in *R* with package *FactoMineR*

Study of the subjects in the PCA space

```
##  Extract the data associated with the observations from the pca object
ind = get_pca_ind(respca)
ind
```

```
Principal Component Analysis Results for individuals
 ===================================================
  Name       Description
1 "$coord"   "Coordinates for the individuals"
2 "$cos2"    "Cos2 for the individuals"
3 "$contrib" "contributions of the individuals"
```

PCA in *R* with package *FactoMineR*

Study of the subjects in the PCA space

```
##  Extract the data associated with the observations from the pca object
ind = get_pca_ind(respca)
ind
```

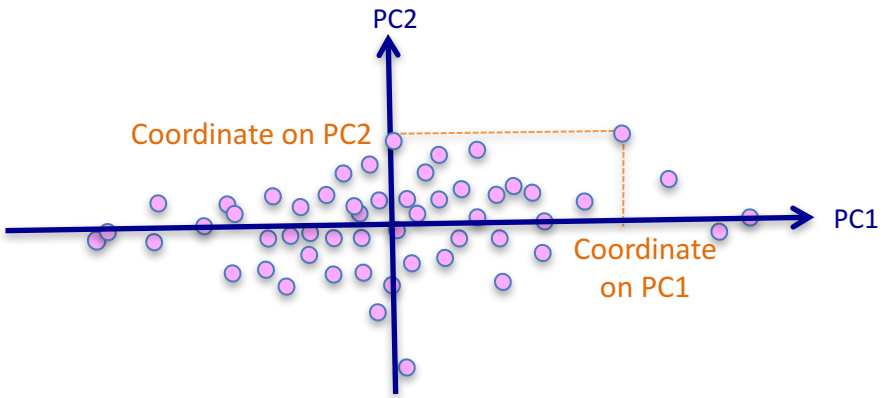
```
Principal Component Analysis Results for individuals
 ===================================================
  Name       Description
1 "$coord"   "Coordinates for the individuals"
2 "$cos2"    "Cos2 for the individuals"
3 "$contrib" "contributions of the individuals"
```

PCA in *R* with package *FactoMineR*

Coordinates of the subjects in the PCA space

```
## coordinates of the individuals in the PCA space
ind$coord
```

```
          Dim.1      Dim.2      Dim.3      Dim.4         Dim.5
p1-IFN   6.204816 -2.8459130  0.7827891 -1.1252954 -0.0005134927
p1-IL10  3.687225 -2.4569552 -2.6855672  1.0344472  0.0025792190
p2-IFN  -9.041060 -1.3802662  2.1466311  0.2488657 -0.3275923186
p2-IL10 -8.304380  0.7585994 -1.7753633 -0.4400246  0.3878338844
p3-IFN   4.545581  2.0215806  3.2095013  0.5763130  0.2648686444
p3-IL10  2.907818  3.9029544 -1.6779910 -0.2943059 -0.3271759365
```



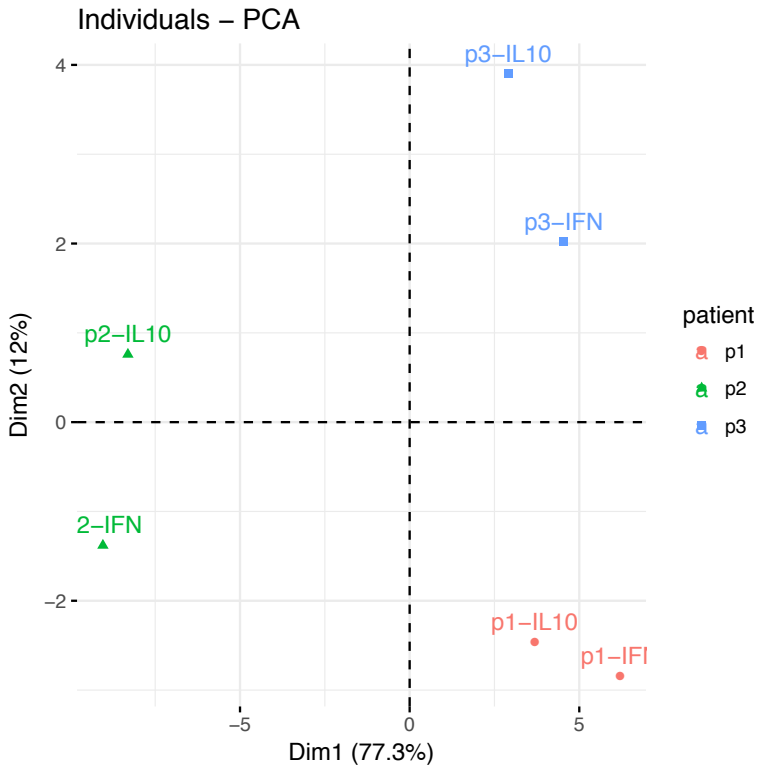
Institut Pasteur C3BI

PCA in *R* with package *FactoMineR*

PCA plot of the subjects

```
## Plot of the samples onto the space spanned by the first and the second PCs.
## Color the data points according to the patient

fviz_pca_ind(respca,
             axes = c(1,2),
             habillage = 'patient',
             invisible = 'quali')
```

```
# Equivalent with the function plot.PCA (run faster for bigger datasets)
plot.PCA(respca, axes = c(1,2), choix = 'ind', habillage = 1,  invisible = 'quali')
```

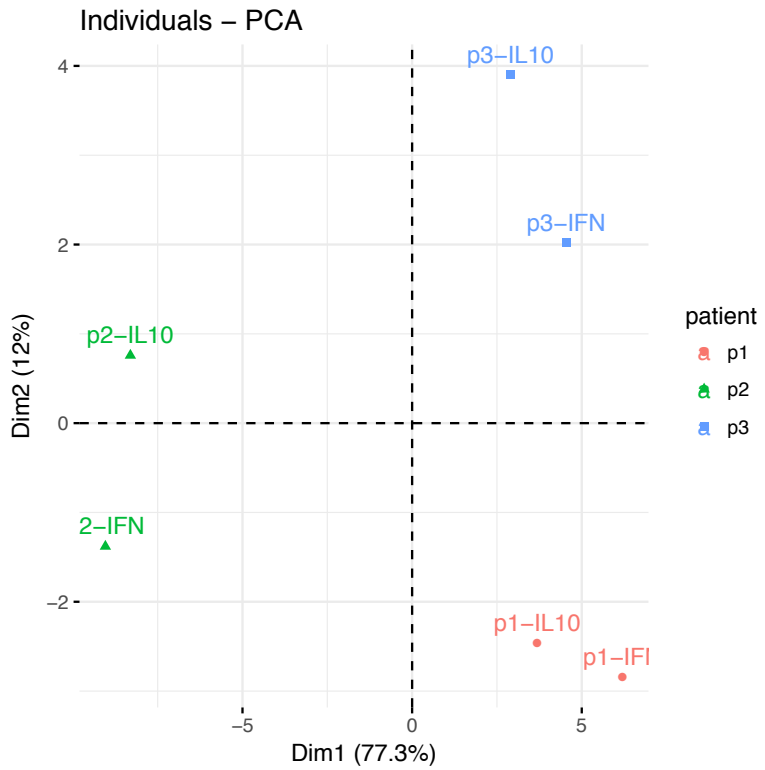


Institut Pasteur | C3BI

PCA in *R* with package *FactoMineR*

PCA plot of the subjects

```
## Plot of the samples onto the space spanned by the first and the second PCs.
## Color the data points according to the patient

fviz_pca_ind(respca,
             axes = c(1,2),
             habillage = 'patient',
             invisible = 'quali')
```



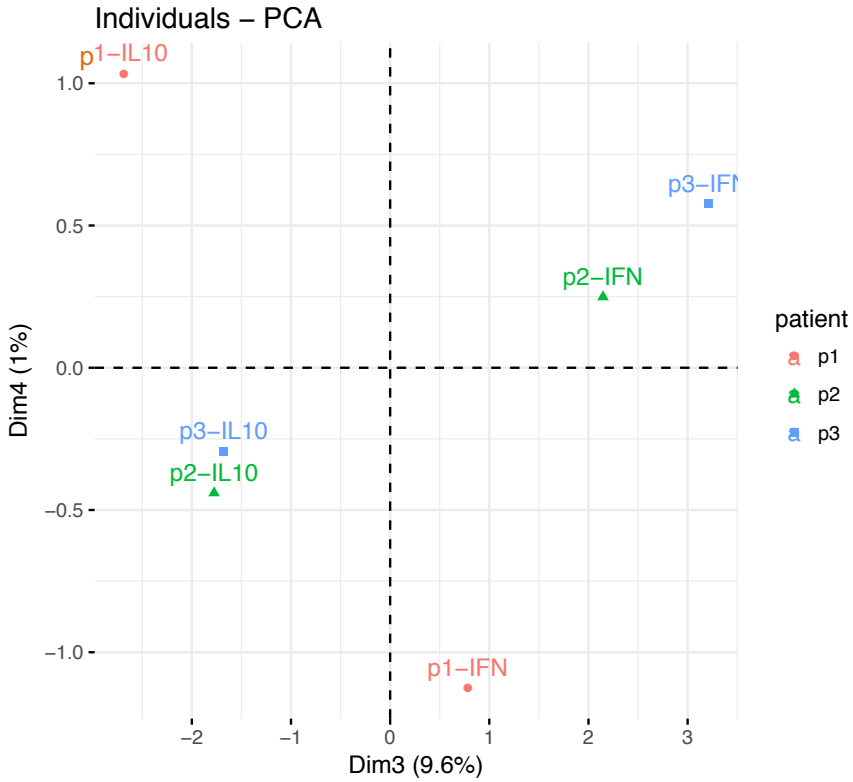
PC1 separates p2 from p1 and p3 (batch?).
PC2 separates the three patients.

PCA in *R* with package *FactoMineR*

PCA plot of the subjects

```
## Plot of the samples onto the space spanned by the third and the
fourth PCs.
## Color the data points according to the patient

fviz_pca_ind(respca,
             axes = c(3,4),
             habillage = 'patient',
             invisible = 'quali')
```

```
# Equivalent with the function plot.PCA (run
faster for bigger datasets)
plot.PCA(respca, axes = c(3,4), choix = 'ind',
habillage = 1,  invisible = 'quali')
```

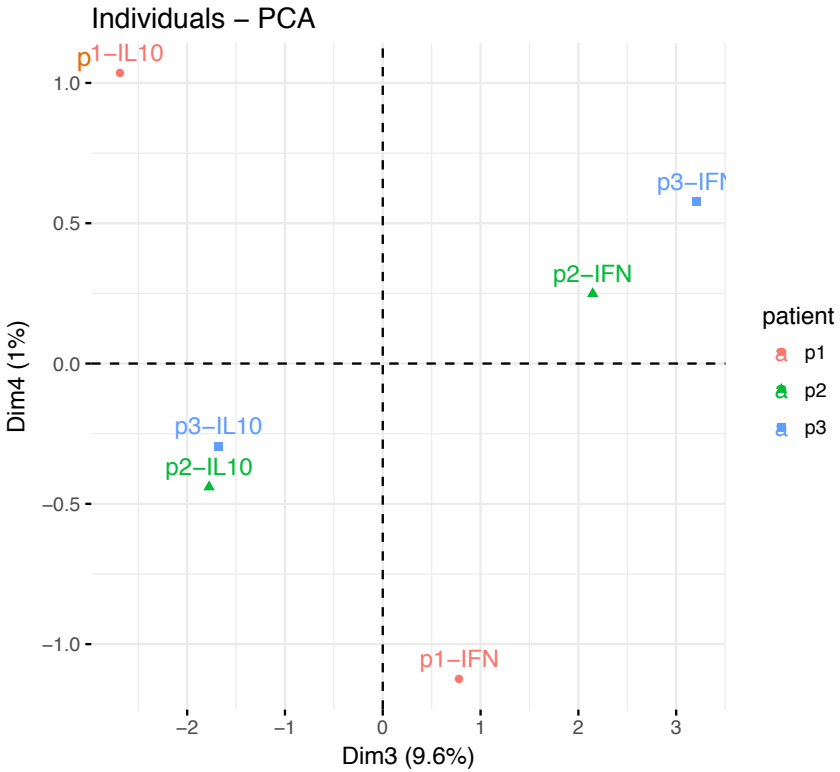


Institut Pasteur | C3BI

PCA in *R* with package *FactoMineR*

PCA plot of the subjects

```
## Plot of the samples onto the space spanned by the third and the
fourth PCs.
## Color the data points according to the patient

fviz_pca_ind(respca,
             axes = c(3,4),
             habillage = 'patient',
             invisible = 'quali')
```



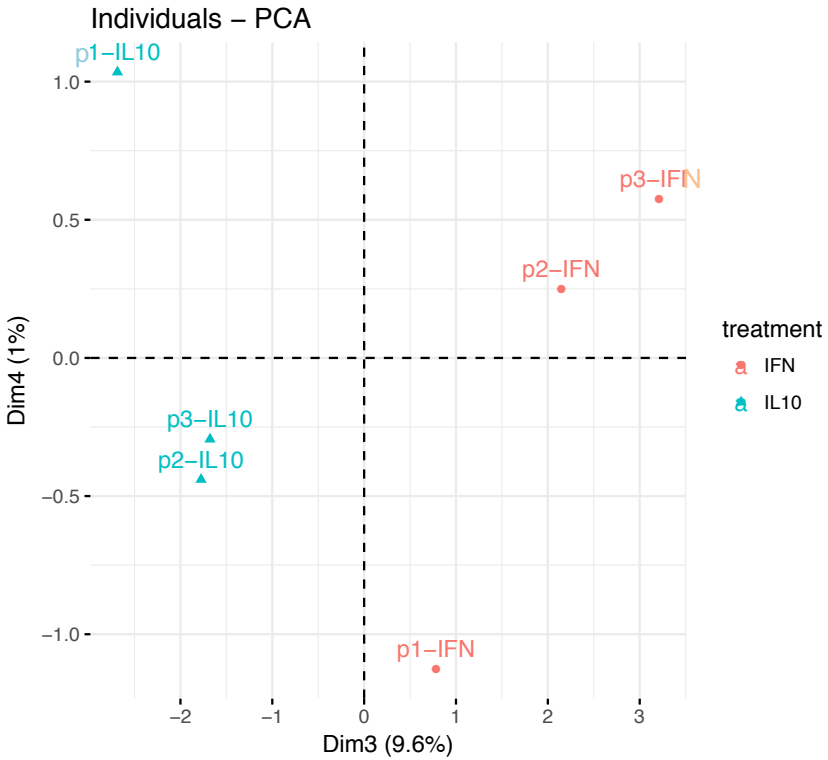
The samples coming from the same patients are not clustered close to each other in the space spanned by PC3 and PC4.

PCA in *R* with package *FactoMineR*

PCA plot of the subjects

```
## Plot of the samples onto the space spanned by the third and the
fourth PCs.
## Color the data points according to the treatment
fviz_pca_ind(respca,
             axes = c(3,4),
             habillage = 'treatment',
             invisible = 'quali')
```

```
# Equivalent with the function plot.PCA (run
faster for bigger datasets)
plot.PCA(respca, axes = c(3,4), choix = 'ind',
habillage = 2,  invisible = 'quali')
```

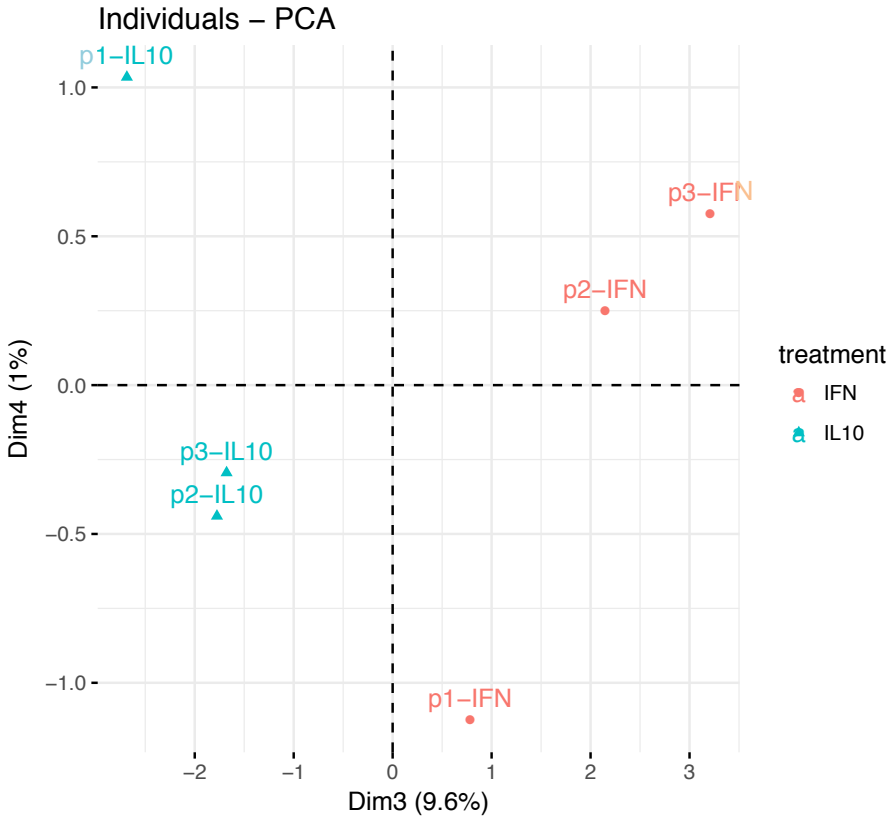


Institut Pasteur C3BI

PCA in *R* with package *FactoMineR*

PCA plot of the subjects

```
## Plot of the samples onto the space spanned by the third and the
fourth PCs.
## Color the data points according to the treatment

fviz_pca_ind(respca,
             axes = c(3,4),
             habillage = 'treatment',
             invisible = 'quali')
```



PC3 separates the samples according to the treatment they received.

PCA in *R* with package *FactoMineR*

Study of the subjects in the PCA space

```
##  Extract the data associated with the observations from the pca object
ind = get_pca_ind(respca)
ind
```

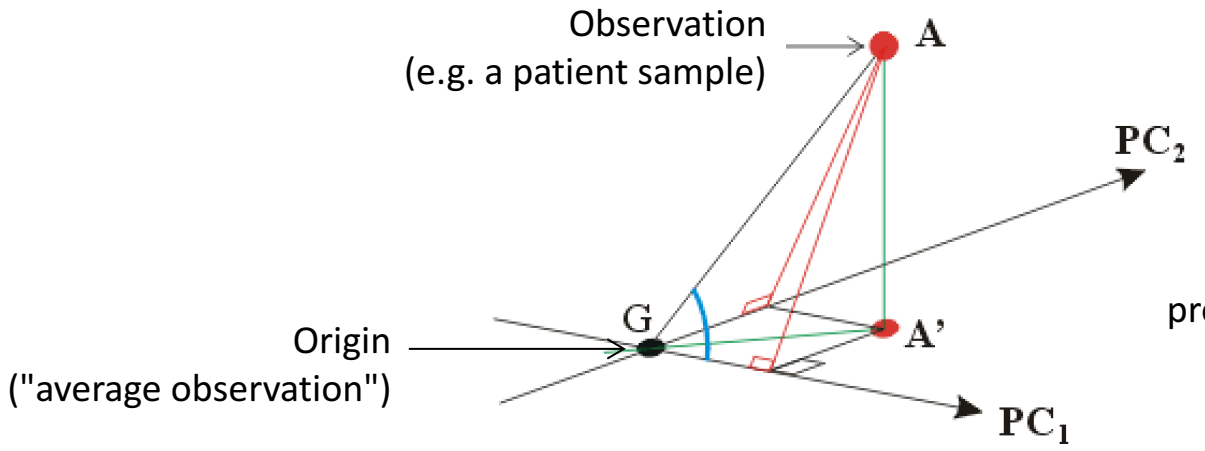
```
Principal Component Analysis Results for individuals
 ===================================================
  Name       Description
1 "$coord"   "Coordinates for the individuals"
2 "$cos2"    "Cos2 for the individuals"
3 "$contrib" "contributions of the individuals"
```

PCA in *R* with package *FactoMineR*

Squared cosine of the subjects with the components

- Is the distance between the origin and the projection of a point in the principal plane a good approximation of the true distance of the observation to the origin ?

- Rather than giving the distance to the principal plane, softwares often output the ratio of the projected distance to the true distance. This quantity corresponds to the square of the cosine of the angle from the right triangle made with the origin, the observation, and its projection on the component.



Cos2(blue angle) =
projected distance (PC1) / true distance

PCA in *R* with package *FactoMineR*

Squared cosine of the subjects with the components

- The squared cosines indicate the "quality of representation" of an observation, observations with a high cos2 (close to 1) are well represented on the principal plane.
- cos2 is convenient to use because in the [0,1] range

```
## squared cosines of the observations
ind$cos2
```

```
             Dim.1       Dim.2      Dim.3        Dim.4        Dim.5
p1-IFN  0.7941691 0.167070009 0.01263993 0.0261209120 5.439059e-09
p1-IL10 0.4870433 0.216253354 0.25836902 0.0383340769 2.383114e-07
p2-IFN  0.9244269 0.021545649 0.05211332 0.0007004288 1.213672e-03
p2-IL10 0.9442531 0.007879506 0.04315673 0.0026511112 2.059518e-03
p3-IFN  0.5828202 0.115275715 0.29055665 0.0093685536 1.978868e-03
p3-IL10 0.3167082 0.570573971 0.10546400 0.0032443138 4.009477e-03
```

- The "quality of representation" of an individual in the subspace spanned by the first k components is equal to the sum of the cos2 associated with these k components.

PCA in *R* with package *FactoMineR*

Study of the subjects in the PCA space

```
##  Extract the data associated with the observations from the pca object
ind = get_pca_ind(respca)
ind
```

```
Principal Component Analysis Results for individuals
 ===================================================
  Name       Description
1 "$coord"   "Coordinates for the individuals"
2 "$cos2"    "Cos2 for the individuals"
3 "$contrib" "contributions of the individuals"
```

PCA in *R* with package *FactoMineR*

Contributions of the subjects to the components

- How much influence has an observation on a principal component ?
- The "**contribution**" measures how much a subject contributes to the variance captured by a principal component.

```
## contributions
ind$contrib
```

```
           Dim.1     Dim.2     Dim.3     Dim.4        Dim.5
p1-IFN  16.600684 22.537716  2.134935 42.059938 6.062348e-05
p1-IL10  5.862292 16.798137 25.128532 35.542843 1.529498e-03
p2-IFN  35.245764  5.301422 16.054985  2.057148 2.467401e+01
p2-IL10 29.736005  1.601370 10.981699  6.431162 3.458310e+01
p3-IFN   8.909373 11.372312 35.889729 11.031952 1.612997e+01
p3-IL10  3.645882 42.389044  9.810120  2.876957 2.461133e+01
```

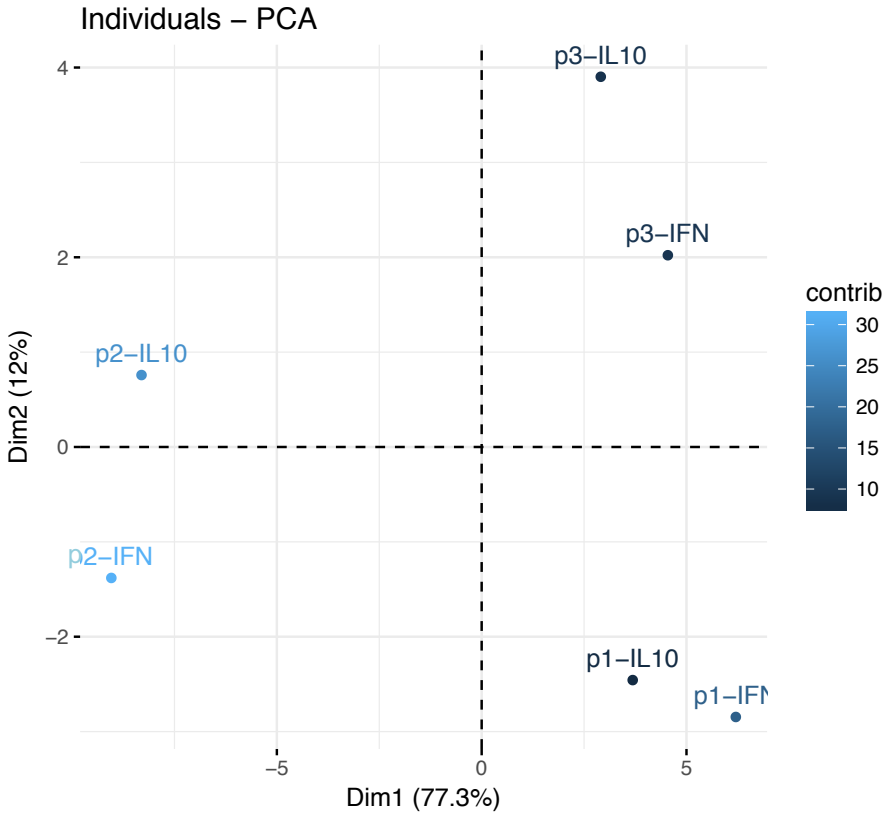
- For a given component, the sum of all contributions equals 1.

```
colSums(ind$contrib)
```

PCA in *R* with package *FactoMineR*

PCA plot of the subjects colored by contribution value

```
## Plot of the samples onto the space spanned by the first and the second PCs.
## Color the data points according to their contributions to PC1

fviz_pca_ind(respca, axes = c(1,2), invisible = 'quali', col.ind = 'contrib')
```




```
ind$contrib[,1, drop = F]
```

	Dim.1
p1-IFN	16.600684
p1-IL10	5.862292
p2-IFN	35.245764
p2-IL10	29.736005
p3-IFN	8.909373
p3-IL10	3.645882

Institut Pasteur | C3BI

PCA in *R* with package *FactoMineR*

Study the variables

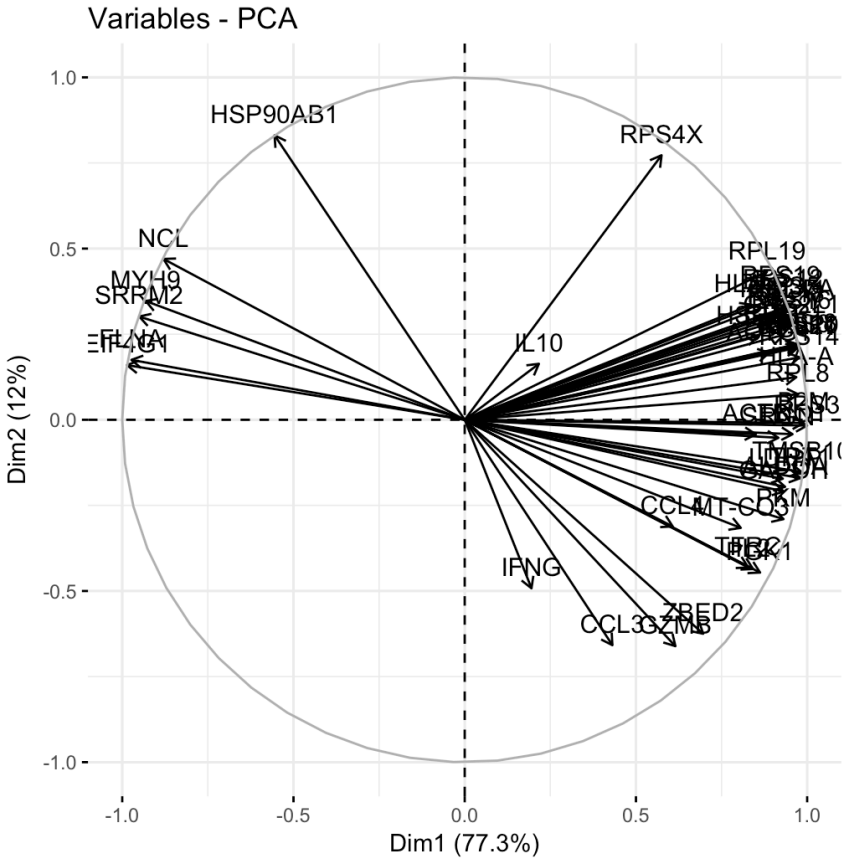
```
## Extract the data associated with the variables from the pca object
vars = get_pca_var(respca)
vars
```

```
Principal Component Analysis Results for variables
 ===================================================
  Name       Description
1 "$coord"   "Coordinates for the variables"
2 "$cor"     "Correlations between variables and dimensions"
3 "$cos2"    "Cos2 for the variables"
4 "$contrib" "contributions of the variables"
```

PCA in *R* with package *FactoMineR*

Plot variable contributions to PCs

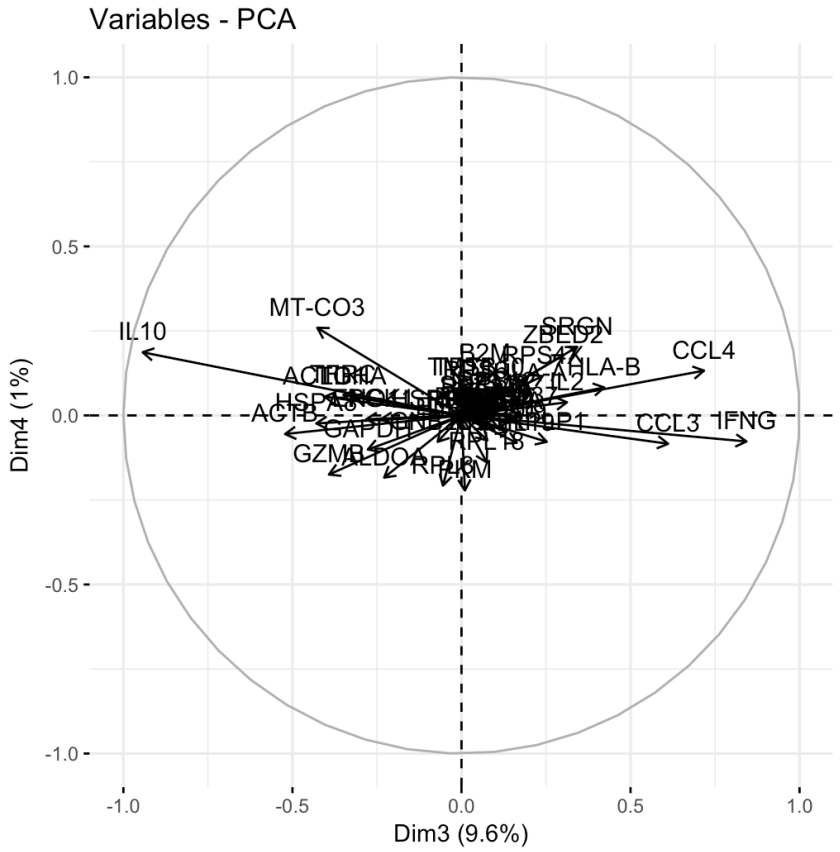
```
## Plot of the correlation of the variables with PC1 and PC2.
# plot.PCA(respca, axes = c(1,2), choix = 'var')
fviz_pca_var(respca, axes = c(1,2), choix = 'var')
```



PCA in *R* with package *FactoMineR*

Plot variable contributions to PCs

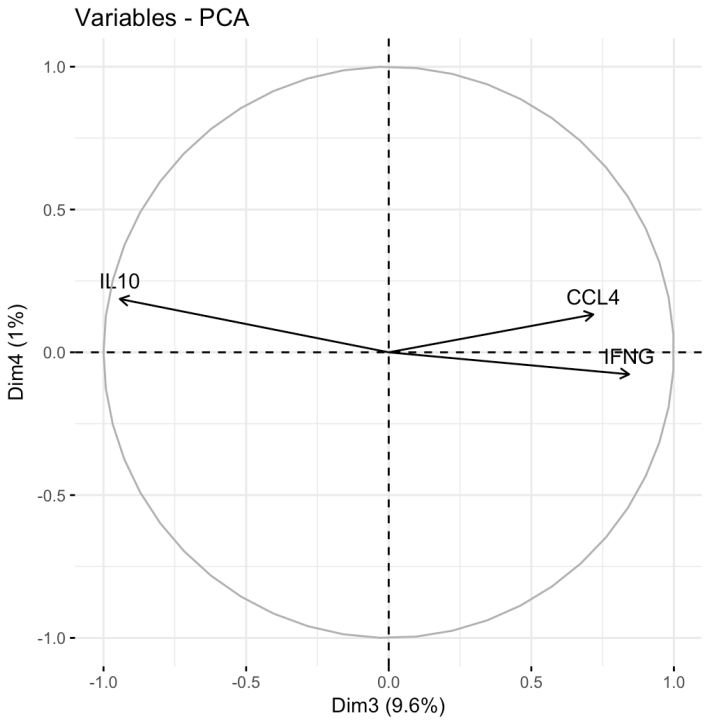
```
## Plot of the correlation of the variables with PC3 and PC4.

fviz_pca_var(respca, axes = c(3,4), choix = 'var')
```



PCA in *R* with package *FactoMineR*

Plot variable contributions to PCs

```
## Plot of the correlation of the variables with the PCs
## use parameter lim.cos2.var to only show variables with a cos2 value larger than 0.5

fviz_pca_var(respca, axes = c(3,4), choix = 'var', select.var = list(cos2 = 0.5))
```

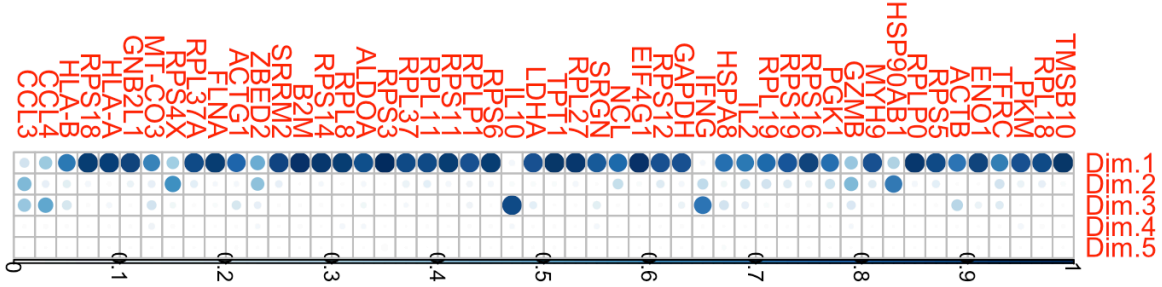


IFN and IL10 are correlated with PC3 which separates the samples according to the treatment (IFN or IL10) they received.

PCA in *R* with package *FactoMineR*

Plot variable contributions to PCs

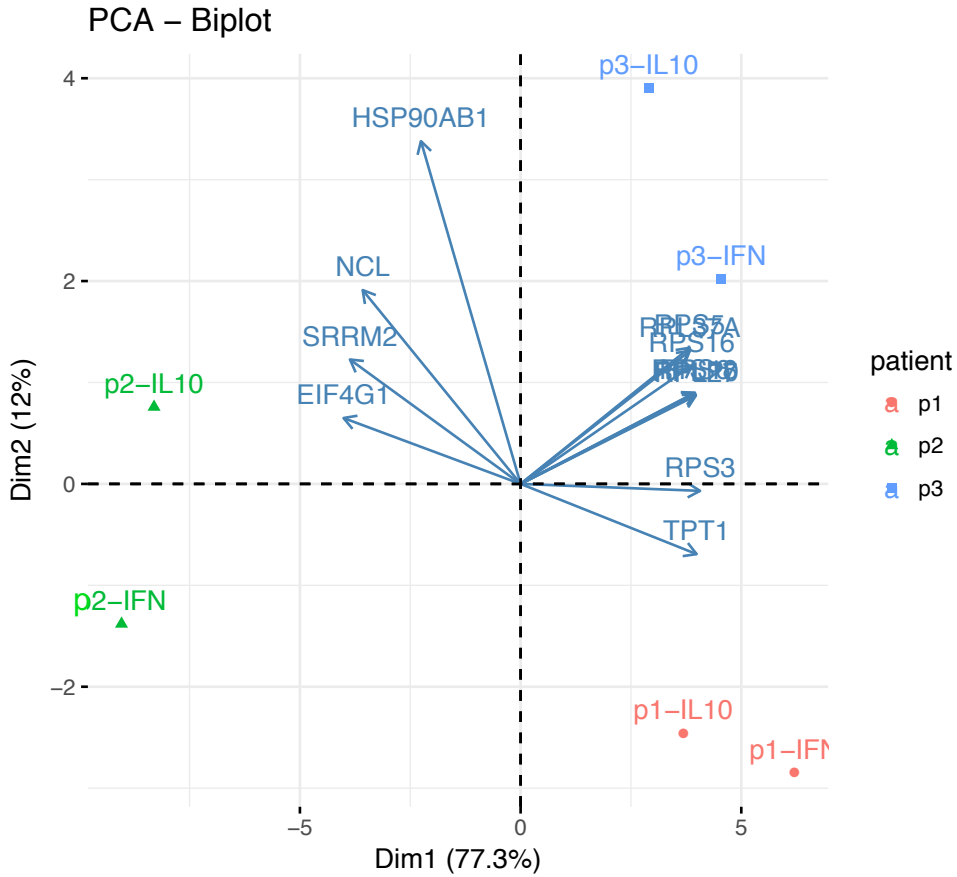
```
## You can visualize the cos2 of the variables on all the dimensions using the
corrplot package

library("corrplot")
corrplot(respca$var$cos2)
```



PCA in *R* with package *FactoMineR*

PCA plot of individual and variable contributions in a "biplot"

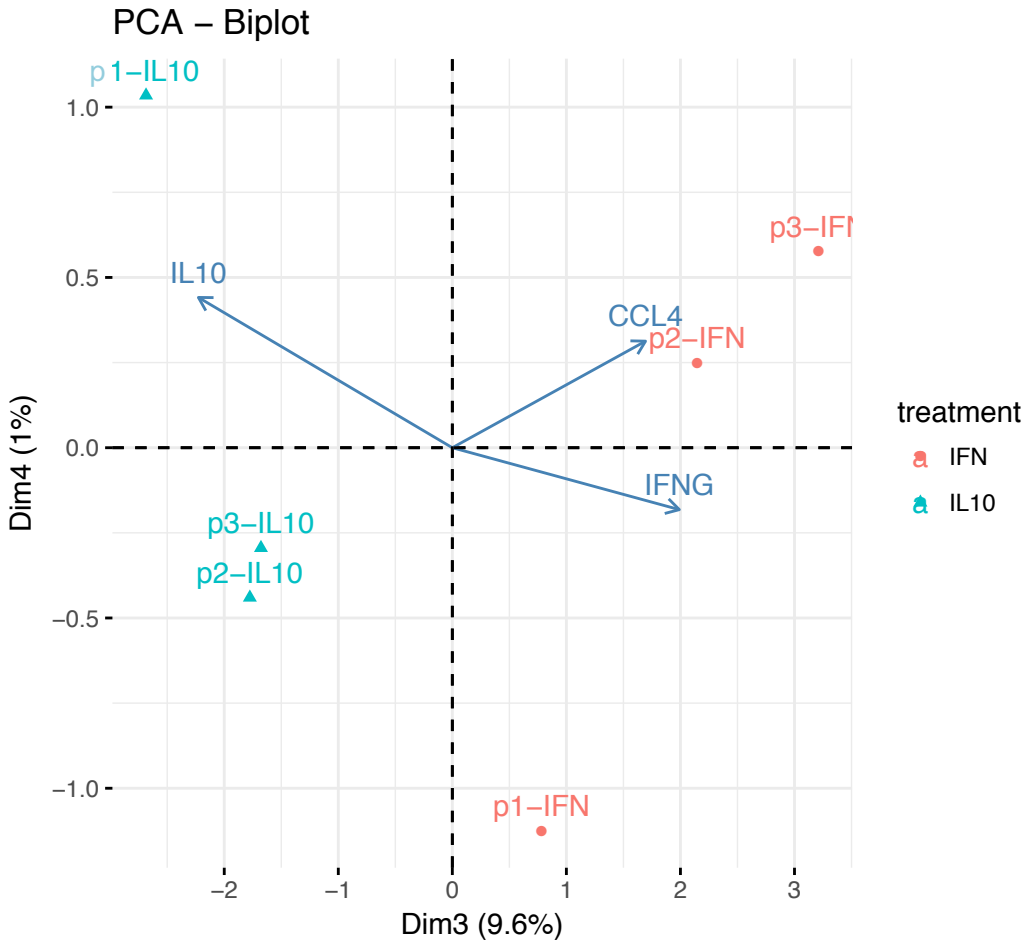
```
library(factoextra)

fviz_pca_biplot(respca, axes = c(1,2),
                habillage = 'patient',
                select.var = list(cos2 = 0.99),
                invisible = c("ind.sup", "quali"))
```



PCA in *R* with package *FactoMineR*

PCA plot of individual and variable contributions in a "biplot"

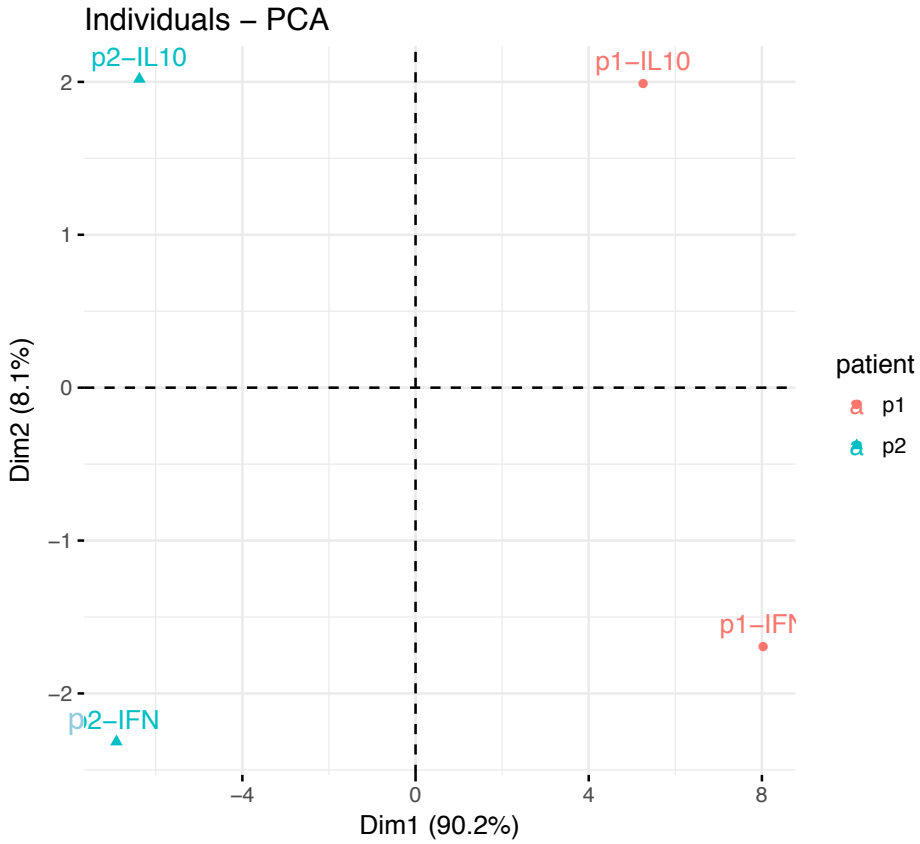
```
fviz_pca_biplot(respca, axes = c(3,4),
                habillage = 'treatment',
                select.var = list(cos2 = 0.5),
                invisible = c("quali"))
```



PCA in *R* with package *FactoMineR*

Discard samples from the PCA computation

```
## Declare the two samples from patient 3 as supplementary individuals.
respca = PCA(X = dt, scale.unit = TRUE, graph = F,
             ncp = 5, quali.sup = 1:2,
             ind.sup = which(dt$patient == 'p3'))

fviz_pca_ind(respca, axes = c(1,2),
             habillage = 'patient',
             invisible = c("ind.sup", "quali"))
```



PC1 separates patient 1 from patient 2

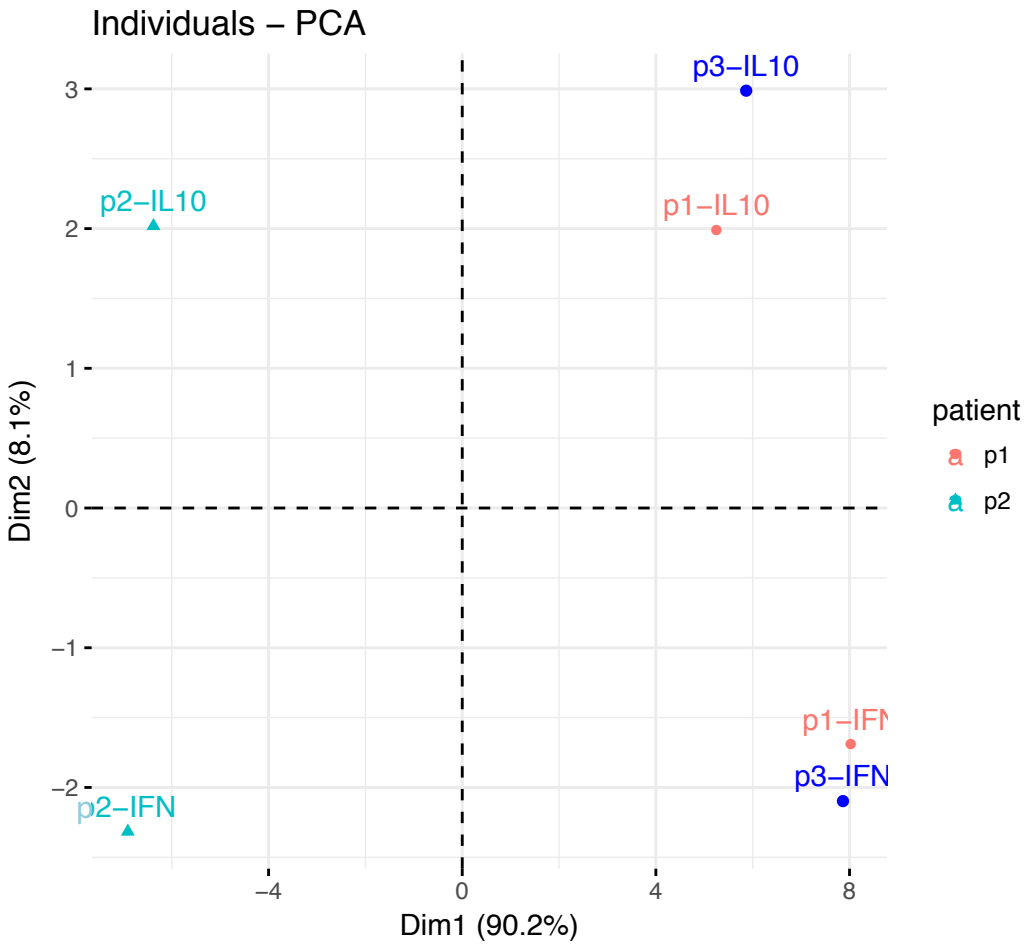
PC2 separates IFN from IL10

Institut Pasteur | C3BI

PCA in *R* with package *FactoMineR*

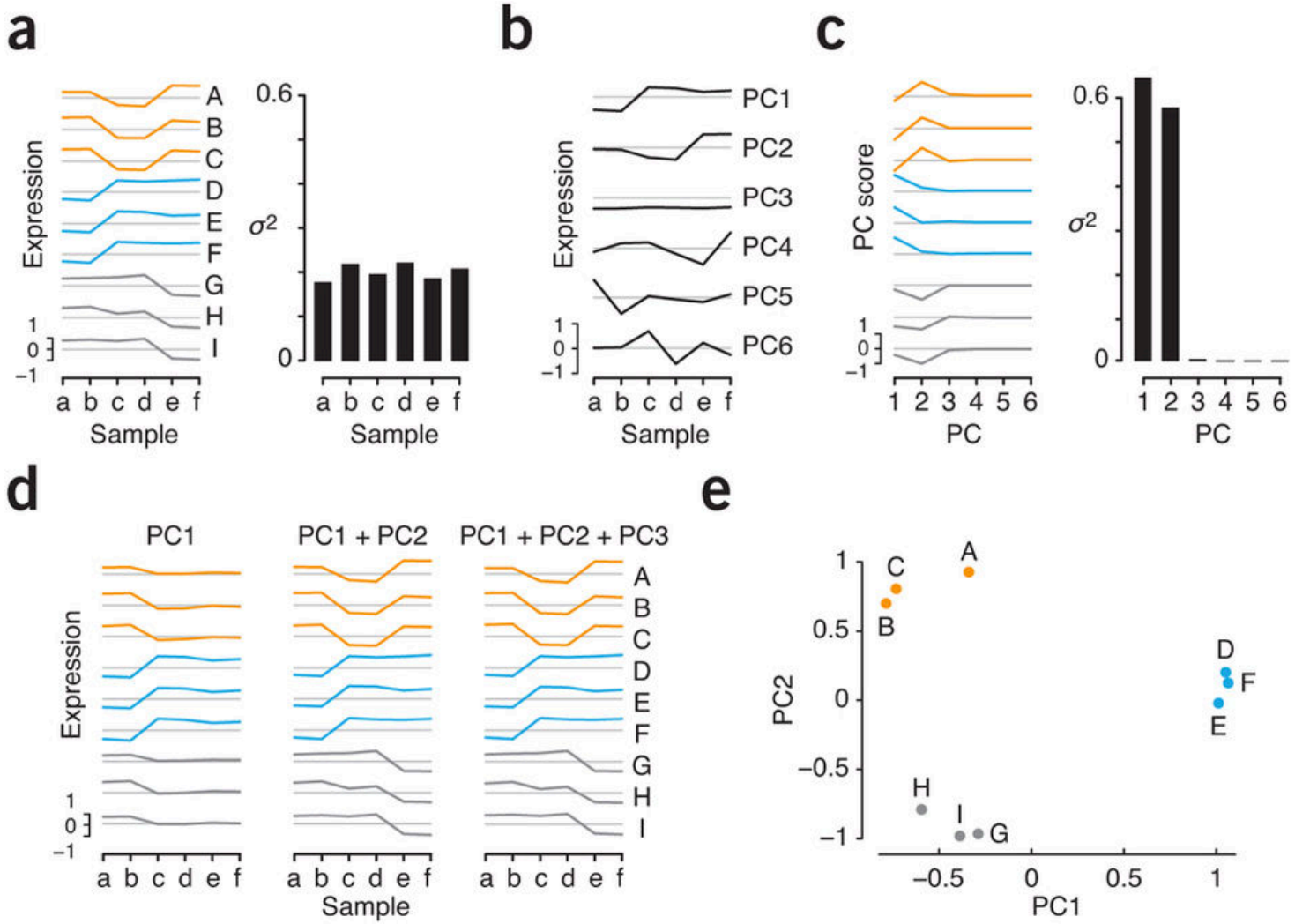
Discard samples from the PCA computation

```
## Project the patient 3 samples in the PC space defined using only patients 1 and 2
fviz_pca_ind(respca, axes = c(1,2),
             habillage = 'patient',
             invisible = c("quali"))
```



Patient 3 appears closer to patient 1 than patient 2.
If needed Patient 3 treatments could have been guessed by using their location in the PC space.

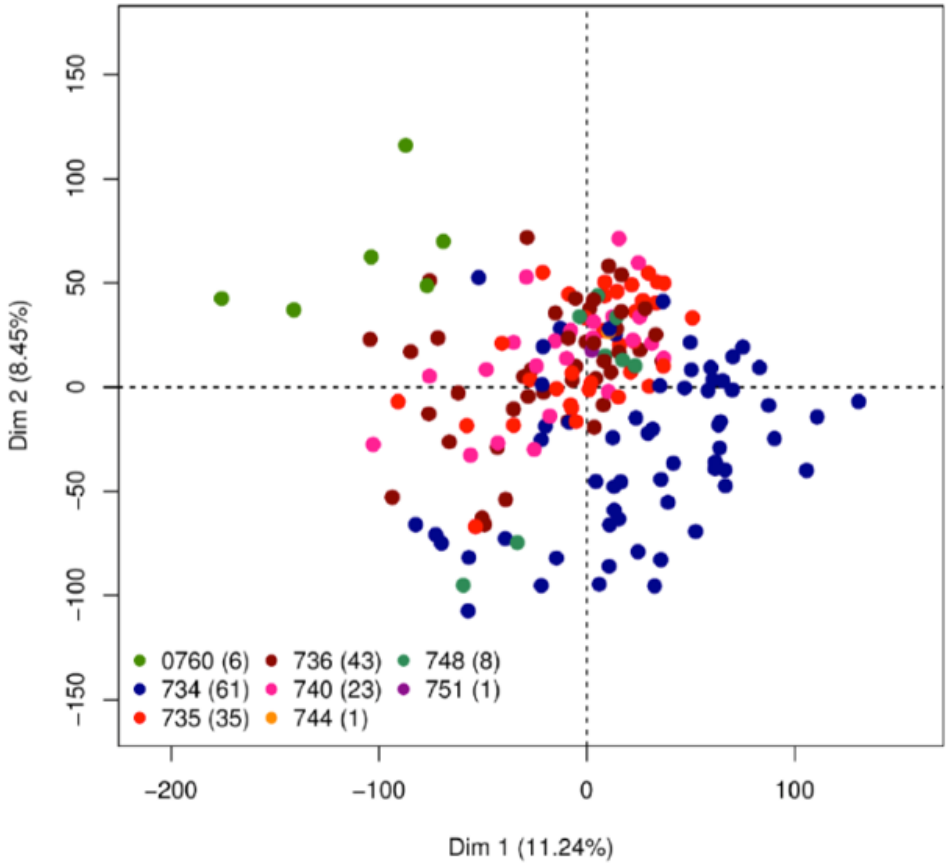
PCA on gene expression data, a summary



Points of Significance: Principal component analysis
Lever *et al.*, Nature Methods, 2017.

PCA, examples of applications

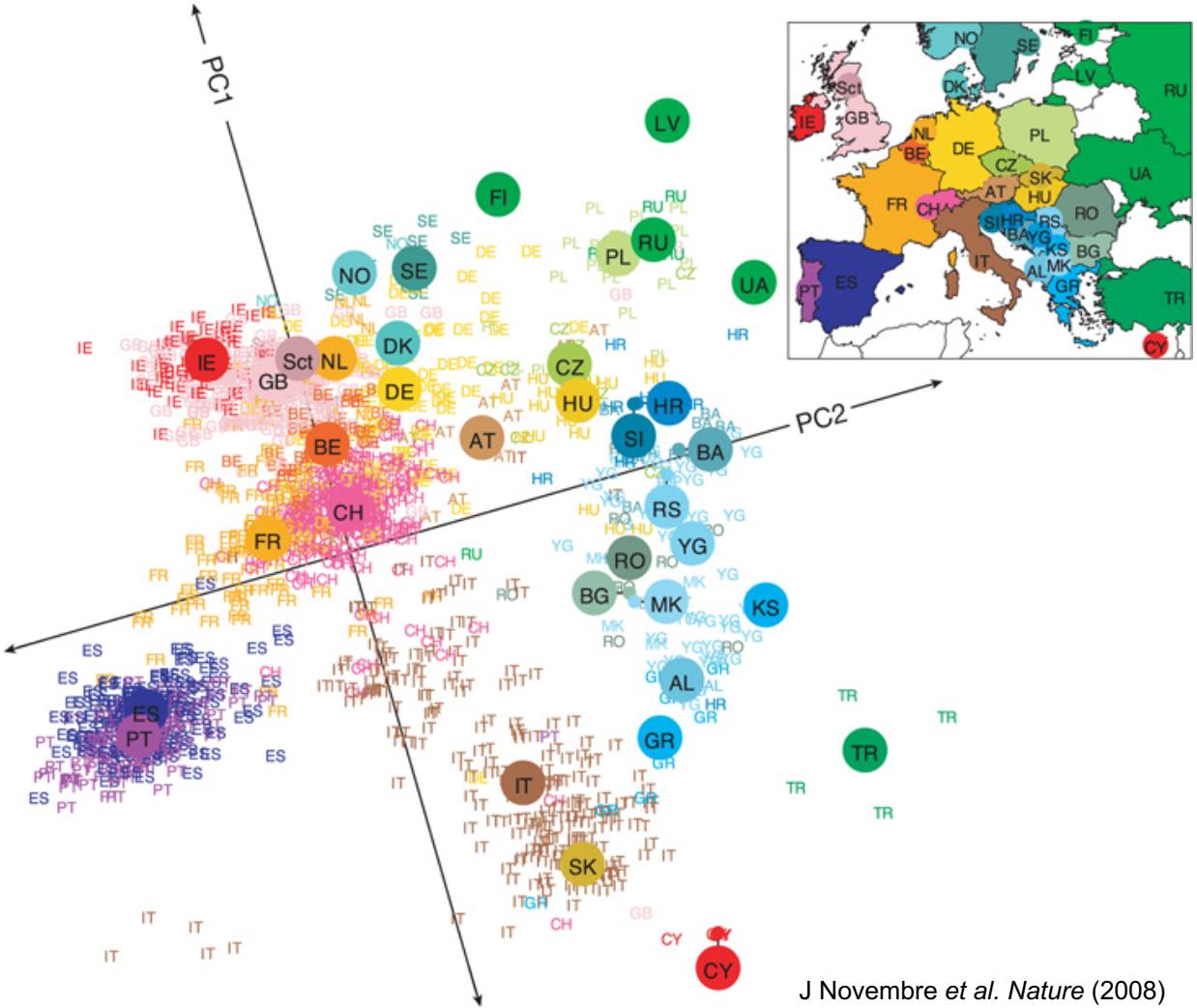
- TCGA gene expression data set: Projection of tumor samples colored by batch onto the space spanned by the first and the second PCs.



Institut Pasteur C3BI

PCA, examples of applications

A dataset of genotypes measured on a SNP array: the projection of individuals onto the space spanned by the first and the second PCs highlights population structure within Europe.

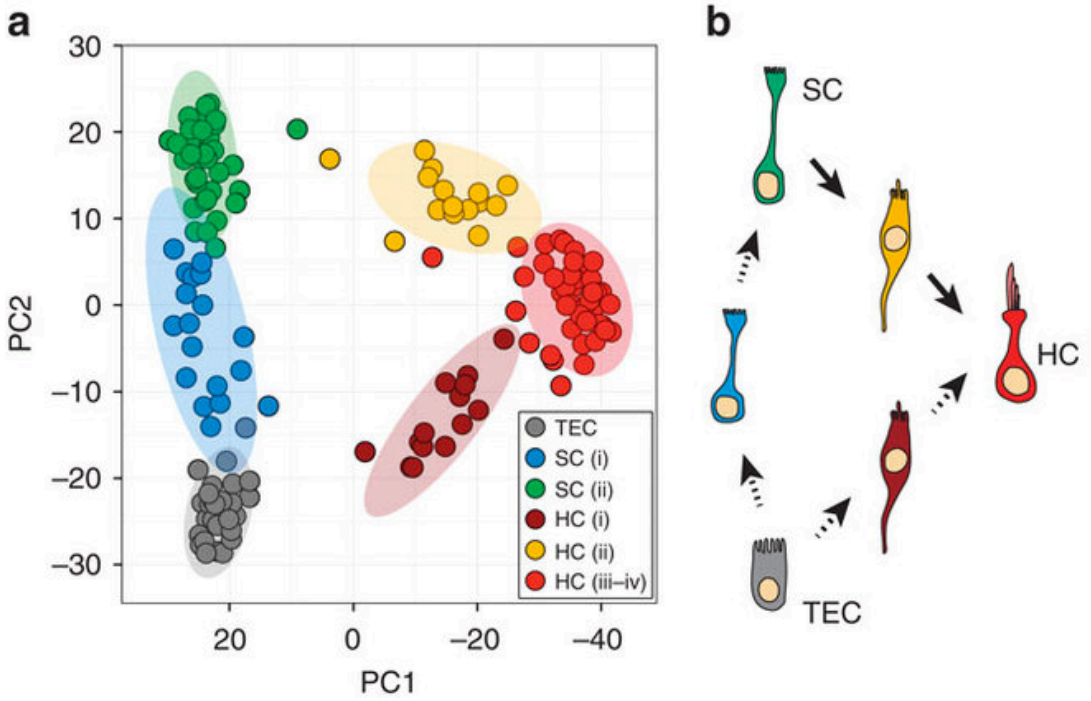


J Novembre *et al. Nature* (2008)

Institut Pasteur | C3BI

PCA, examples of applications

Example: RNA-Seq single cell data for single cells of the inner ear: the projection of individuals onto the space spanned by the first and the second PCs reflects cell types.



J Burns *et al. Nature* (2015)

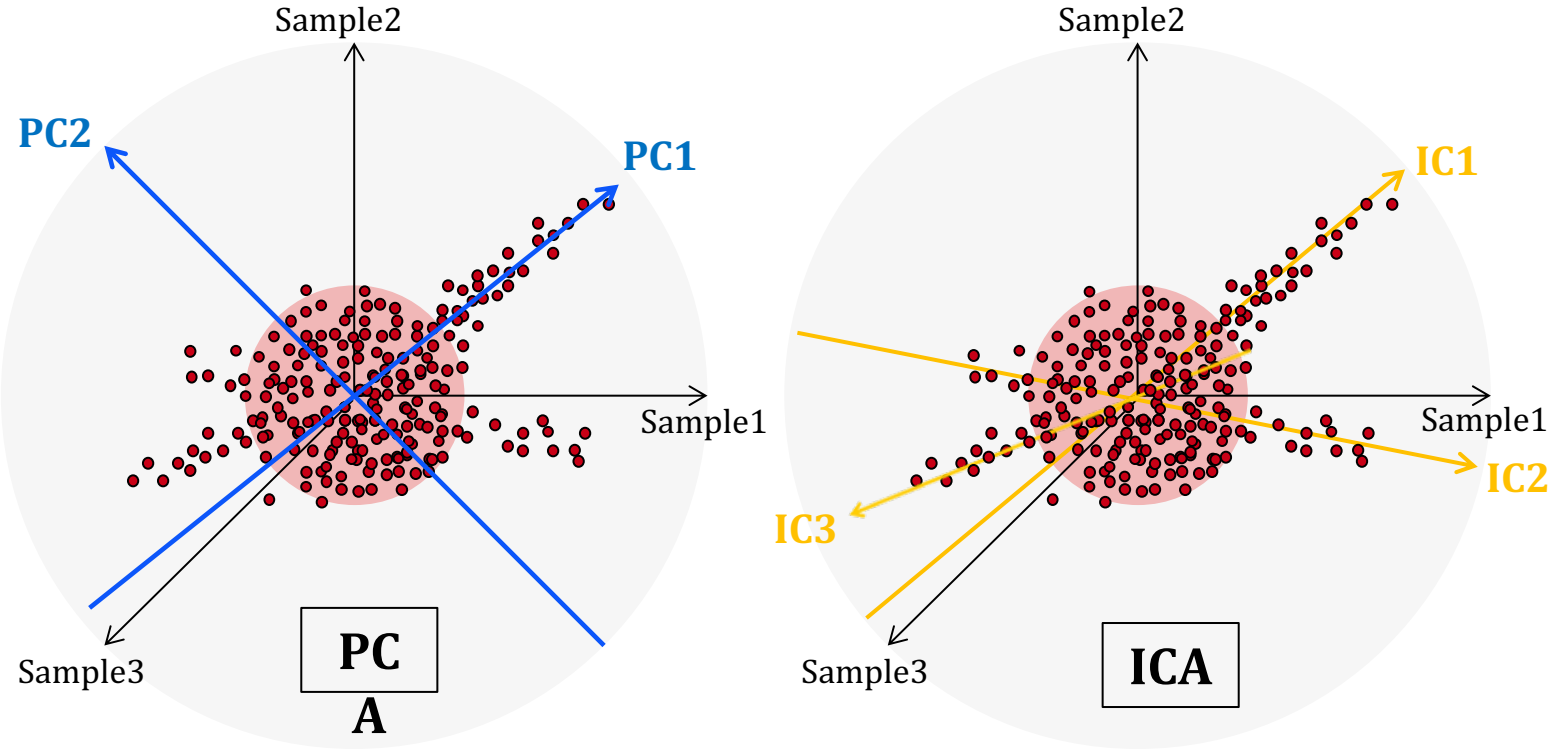
Hypotheses behind PCA and alternatives

- Applying PCA, we assume that the biological question is related to the highest variances in the dataset, not adequate for strongly non-Gaussian processes.

- Orthogonality constraint between components that is not always in line with biological reality.

- An alternative is to use for example Independent Component Analysis.

Comparison between PCA and ICA



	PCA	ICA
Constraint(s)	Orthogonality	Independence
Decomposition	Unique	Non unique
Ranking of the components	Yes	No

A word on Multidimensional Scaling (MDS)

- MDS represents the information contained in a distance matrix using a low-dimensional (2 or 3) configurations of points that preserve as well as possible the original distances.

- It allows the visualisation of the observations based on their level of dissimilarity.

- MDS input: a *n* x *n* matrix of pairwise distances (not necessarily Euclidean) between the observations and a dimension *K* (typically 2 or 3).

- MDS output: *n* vectors of coordinates in the Euclidean space $\mathbf{R}^K$ with pairwise distances "as close as possible" to the original distances.

A word on Multidimensional Scaling (MDS)

- MDS can be formulated as an optimization problem, where one seeks to minimize a loss function or stress function, i.e. how is defined "as close as possible".

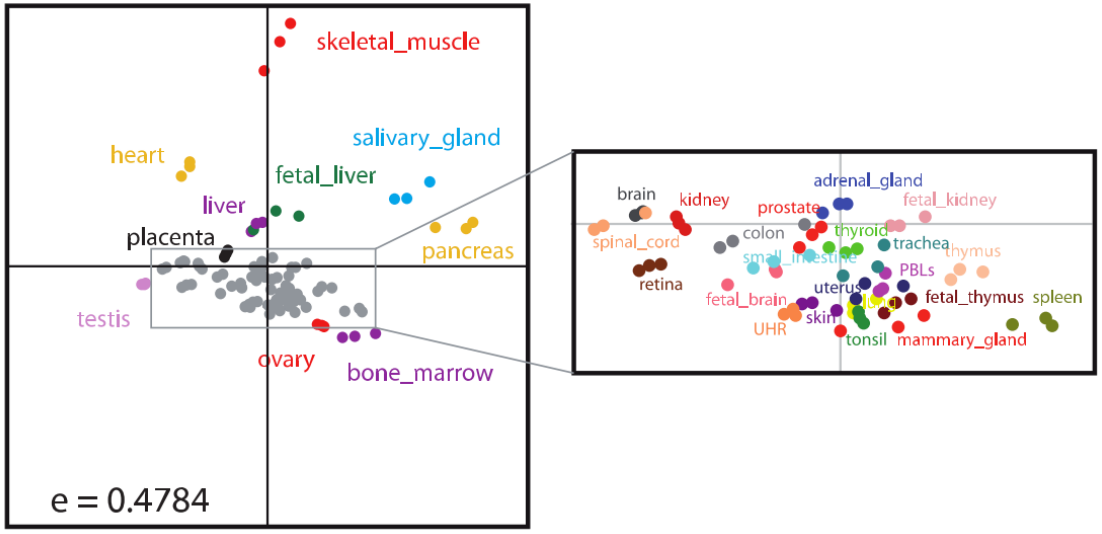
- MDS is a family of different algorithms, each designed to arrive at an optimal low-dimensional configuration. They differ by the distance they use in the new space and the loss/stress function they aim at minimizing.

- Classical MDS or metric MDS uses the Euclidean distance in the new space $\mathbf{R}^K$ and the sum of squared differences between the original and new distances as loss function. $\sum_{i<j}(D_{i,j} - \widehat{D}_{i,j})^2$

- In non-metric MDS, one seeks a configuration of points whose distance reflect as closely as possible the rank order of the original distances.

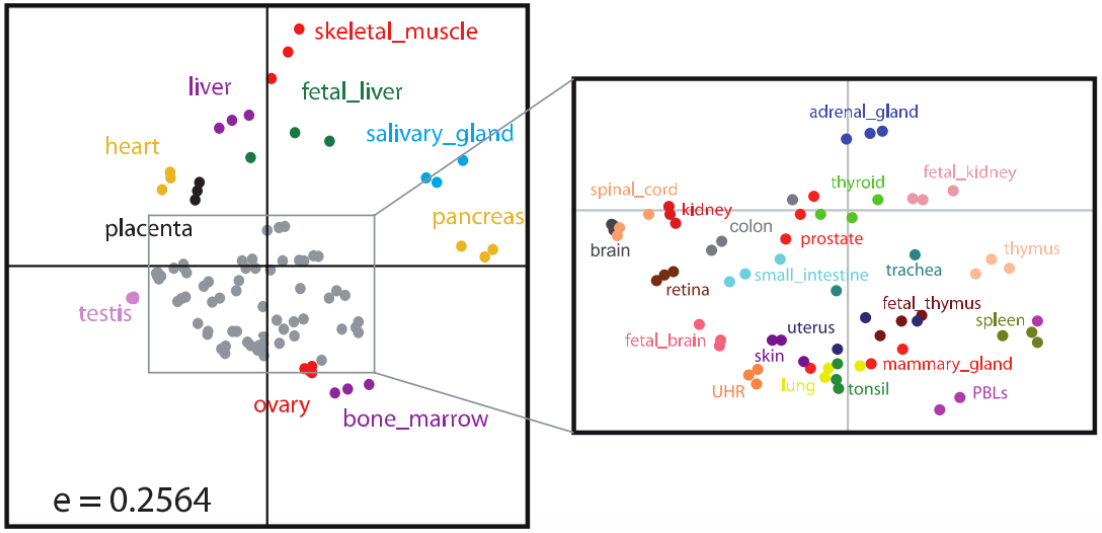
A word on Multidimensional Scaling (MDS)

96 transcriptome datasets of 32 human cells



PCA

Stress = 0.4784



MDS

Stress = 0.2564

Better representation of the overall distribution (local and global)

Institut Pasteur C3BI

Hypotheses behind PCA and alternatives

- PCA is adapted to continuous data only.
- For qualitative data, there are others methods dedicated to the descriptive study of the link between qualitative variables:

Correspondence factor analysis (for two variables) or Multiple Factor Analysis (for more than two variables).

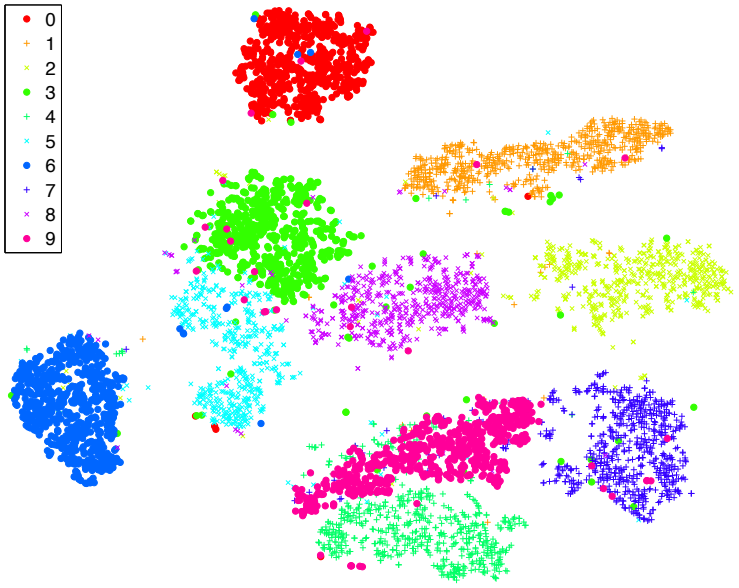
- All the "classical" dimensionality reduction approaches (PCA, MDS, ICA, …) only characterize *linear* subspaces in the data.
In the *nonlinear* case, other recent techniques have been developed (Kernel PCA, Isomap, locally linear embedding, tSNE, …).

A word on t-SNE

- t-Distributed Stochastic Neighbor Embedding (t-SNE) is another distance-based embedding method like MDS for non-linear cases.
- t-SNE is non-deterministic, i.e., the results can differ between different runs.
- t-SNE depends on a tuneable parameter (the "perplexity" allowing to balance between local and global aspects of the data), and other parameters related to the optimization process.
- Like MDS, it aims at preserving distances between points, but it places higher priority on preserving relationships between small distances.
- The “importance” of preserving distances falls off according to a t-distribution.

A word on t-SNE

- t-SNE has become very popular (especially in single cell RNA-Seq) because of its flexibility and it can often find structure where other dimensionality-reduction algorithms cannot.
- t-SNE is for visualization only, not as interpretable as PCA.
- Relative sizes of clusters are not conserved in a t-SNE plot.
- Distances between clusters might not mean anything.
- t-SNE tends to expand denser regions of data.



- Paper: L. van der Maaten and G. E. Hinton, "Visualizing data using t-SNE," Journal of Machine Learning Research, vol. 9, pp. 2579-2605, November 2008.
- R package: tsne

Institut Pasteur | C3BI

Exercise: Perform a PCA of the Fisher's iris dataset

- The R dataset iris provides measurements (in cm) of the variables sepal length and width and petal length and width, for 50 flowers from each of 3 species of iris (setosa, versicolor, and virginica).

```
# Load the iris dataset
data(iris)
# read the help
?iris
```

- **Your turn! Perform a PCA of the iris dataset and interpret.**

Clues:

- Select the numeric columns of the iris dataset to perform the PCA
- Extract and plot the percentage of variance captured by each PC
- Plot the observations in the PC spaces, color with respect to an adequate variable
- Plot the variable contributions to the components
- Interpret
- What are the characteristics of the samples contributing the most to PC1 ? Extract the information about a few of them.