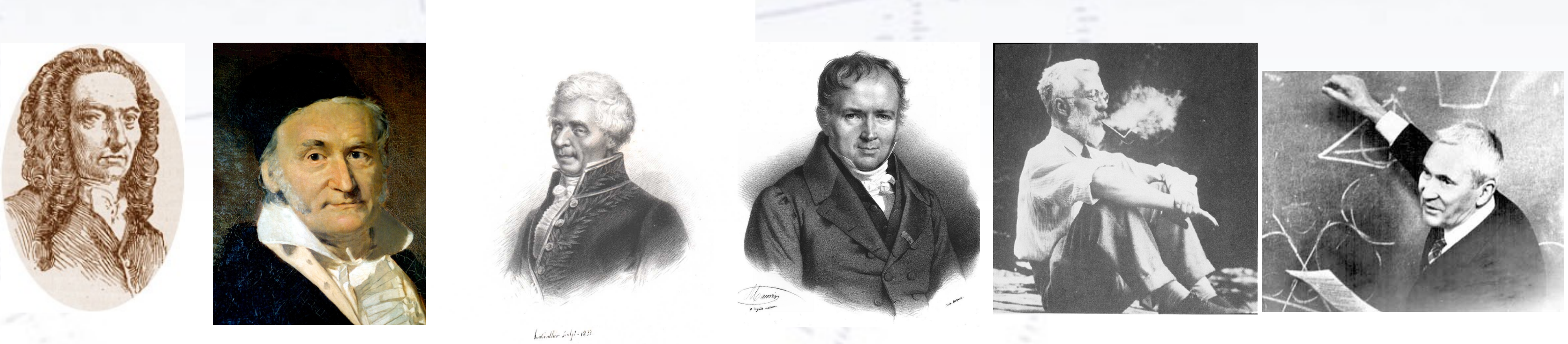
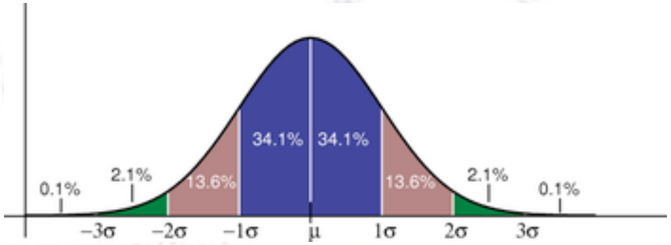
Applied ML

(Kernel) Principle Component Analysis



Troels C. Petersen (NBI)

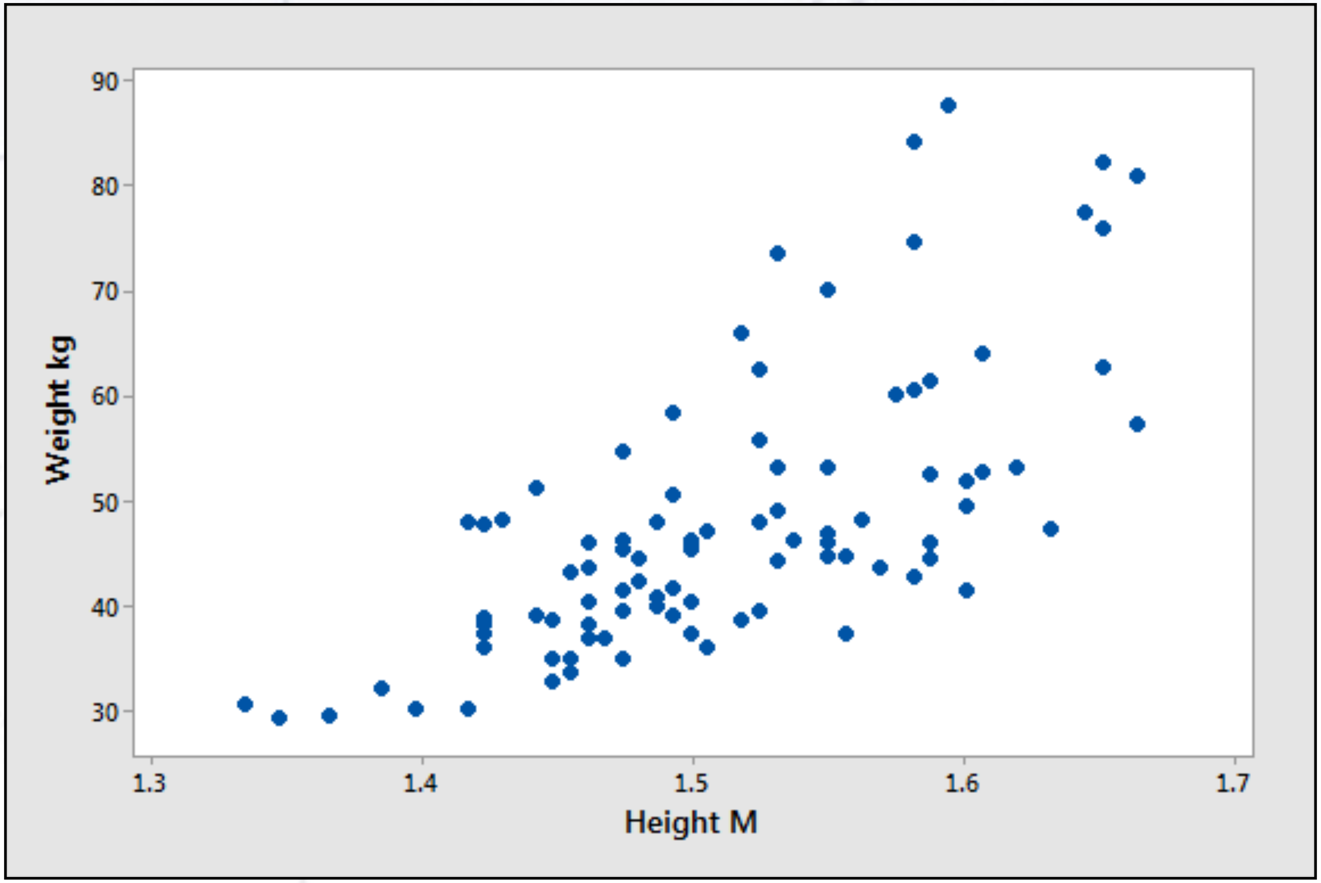


"Statistics is merely a quantisation of common sense - Machine Learning is a sharpening of it!"

PCA overview

Consider data which have correlations, here in 2D (for visualisation), but potentially in (very) high dimension.

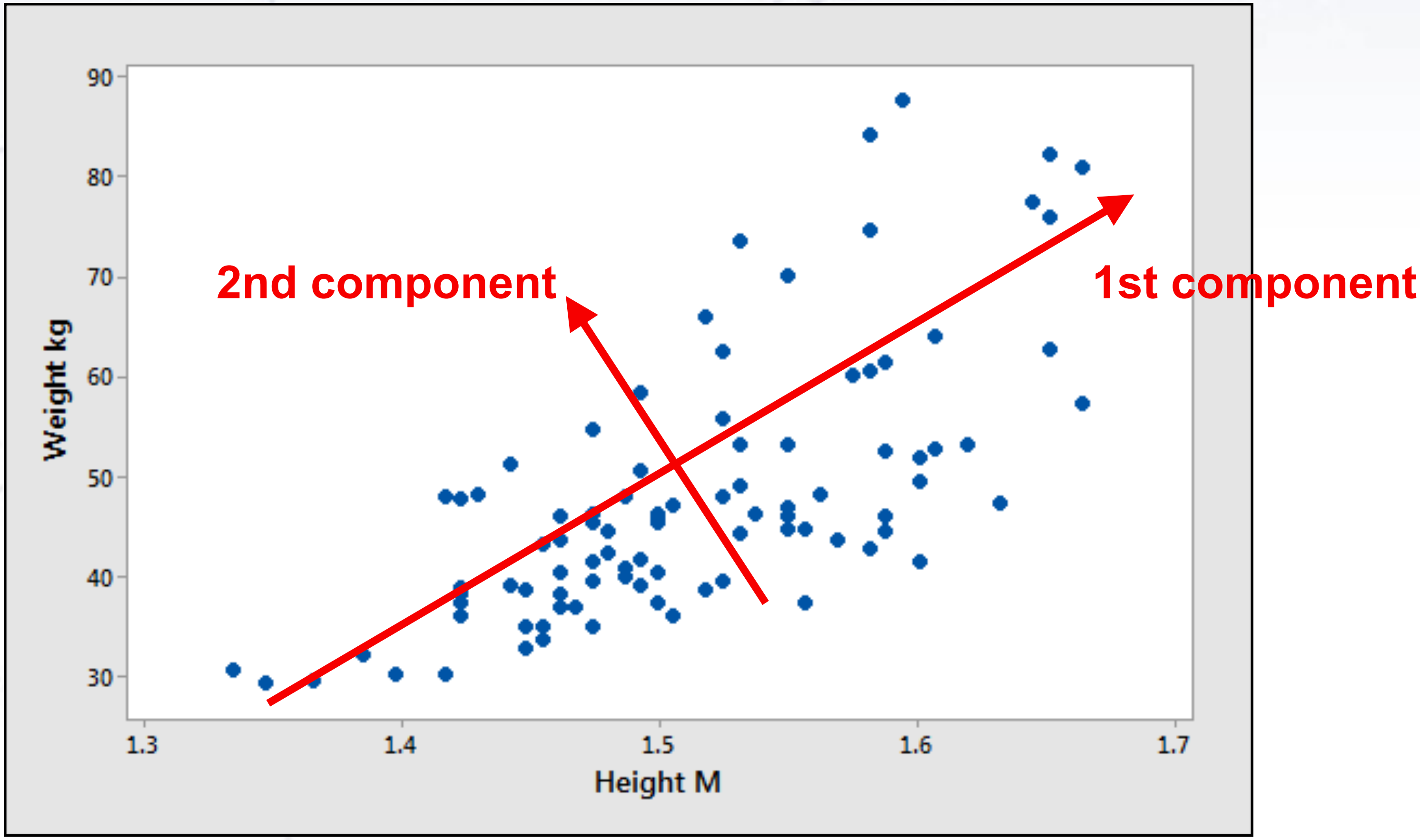
We want to apply a PCA to this data, to reduce dimensionality!



PCA overview

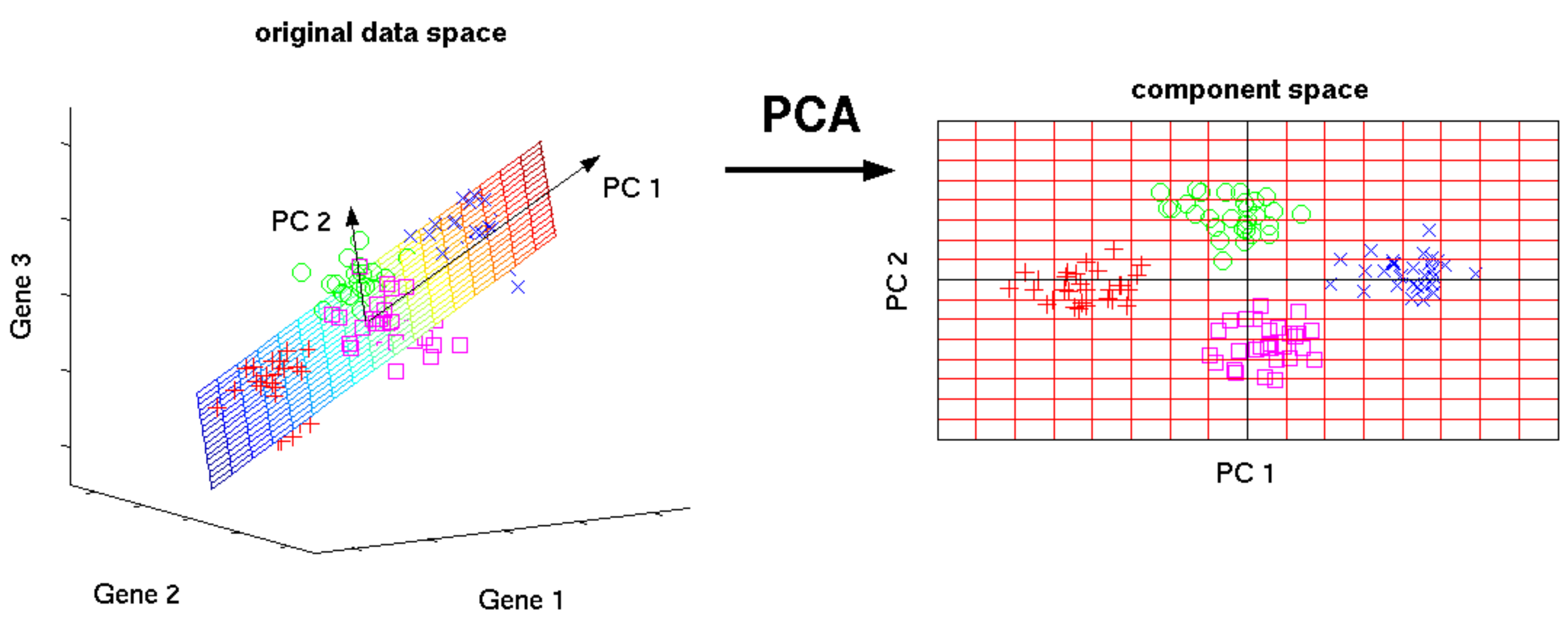
Find the direction, which has the maximum variance, i.e. "best along the direction of the data".

The effective way to do this, is to find the eigenvectors and eigenvalues, and rank the eigenvectors (i.e. directions) according to eigenvalues.



PCA overview II

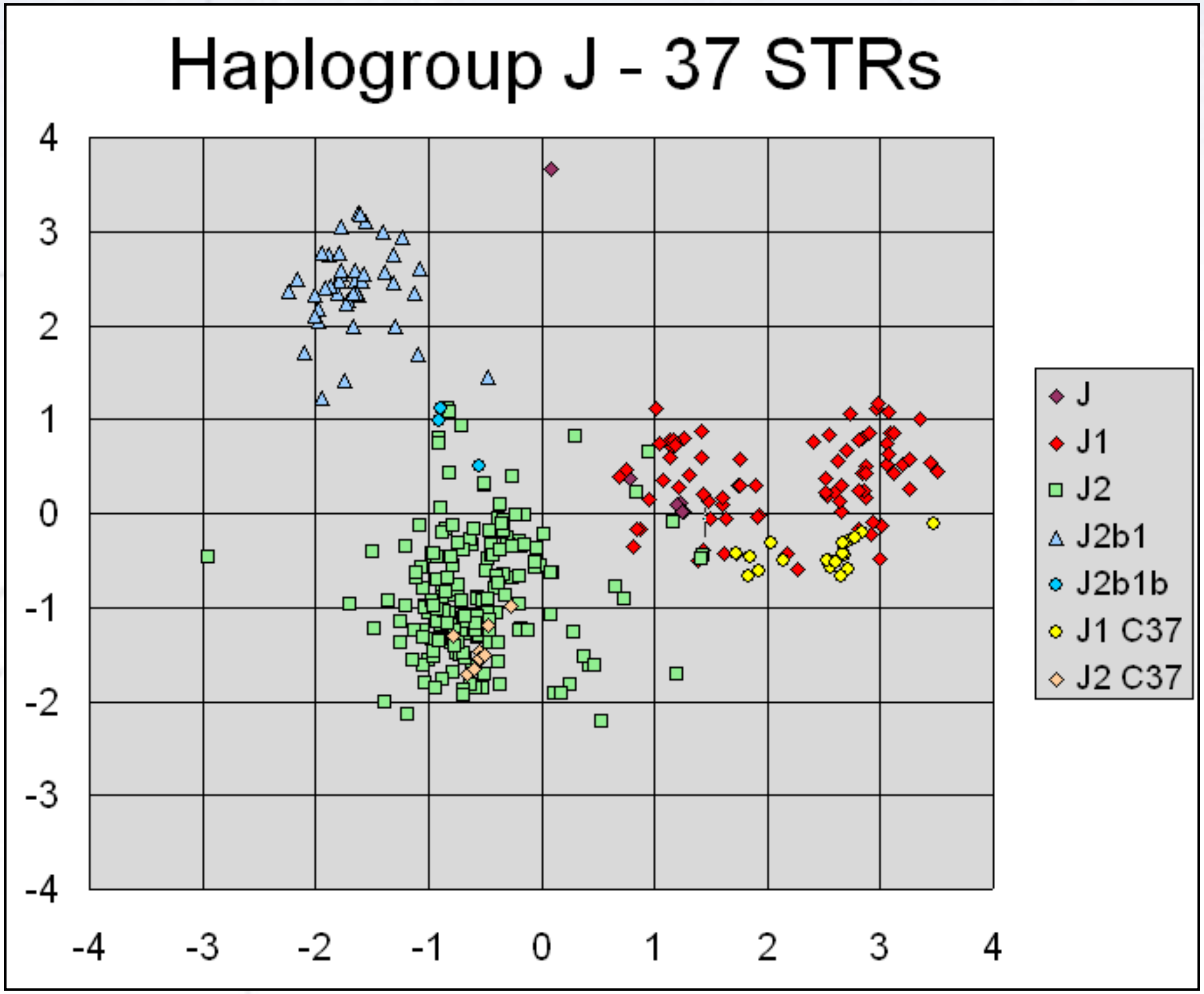
It is hard to illustrate the high dimensional cases, but here is an attempt at seeing 3D points reduced to 2D points by PCA.



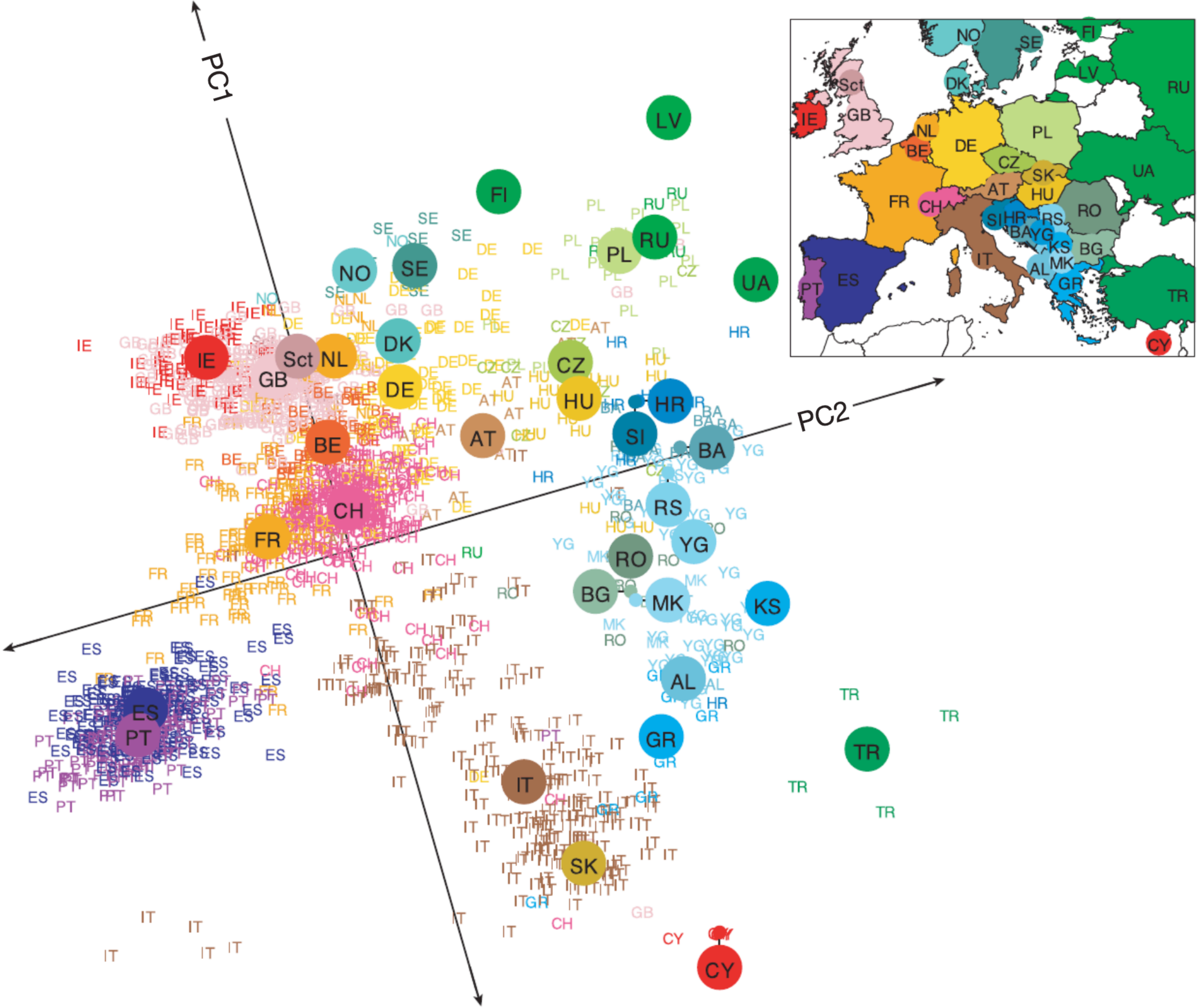
Essentially, one finds the two (orthogonal) directions, which approximates the data best, and "throw away" all the other dimensions in this new space.

PCA example

Using PCA on genetic data is quite common, as there is no “labels” and the dimensionality is very high (millions), here boiled down to two dimensions:

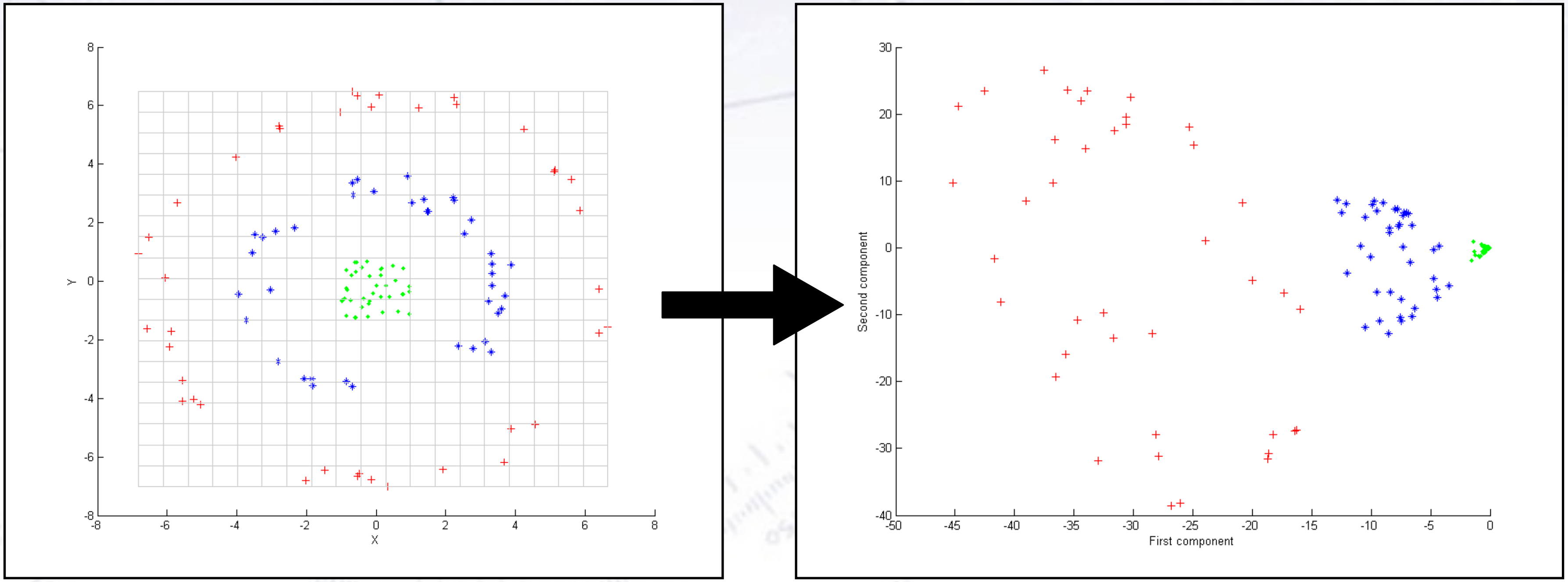


PCA example


PC1
PC2
LV
FI
RU
PL
UA
NO
SE
DK
IE
Sct
NL
GB
DE
CZ
HU
HR
SI
BA
AT
BE
CH
FR
RS
YG
RO
BG
MK
KS
AL
GR
TR
ES
PT
IT
SK
CY
NO
SE
FI
DK
Sct
LV
RU
IE
GB
NL
BE
DE
PL
UA
CZ
SK
AT
HU
FR
CH
SI
HR
RS
RO
BA
YG
KS
BG
IT
AL
MK
ES
PT
GR
TR
CY

Kernel PCA

For non-linear problems, the kernel PCA might be the solution. Here, a (non-linear) kernel is applied before the PCA transformation. This is computationally heavy, but often works well as shown below:



There are other non-linear unsupervised methods, in particular t-SNE and UMAP have gained popularity from their performance.