Classification methods

The goal of classification methods is associating an object (sample) to a specific class, based on the values of a certain number of independent variables (descriptors)

The following requisites have to be fulfilled:

1) classes must be defined preliminarily
2) a training set of objects (samples) must be available
3) each object of the training set can be assigned to one of the predefined classes.

The preliminary definition of classes can occur according to one of the following criteria:

1) classes are known *a priori*, based on theoretical considerations
2) classes are searched for through methods related to Cluster Analysis
3) classes can be defined through a categorical variable (e.g., the type of catalyst adopted for a chemical reaction)
4) classes are defined through the categorization of a quantitative variable.

An example of procedure 4 is shown in the following figure:

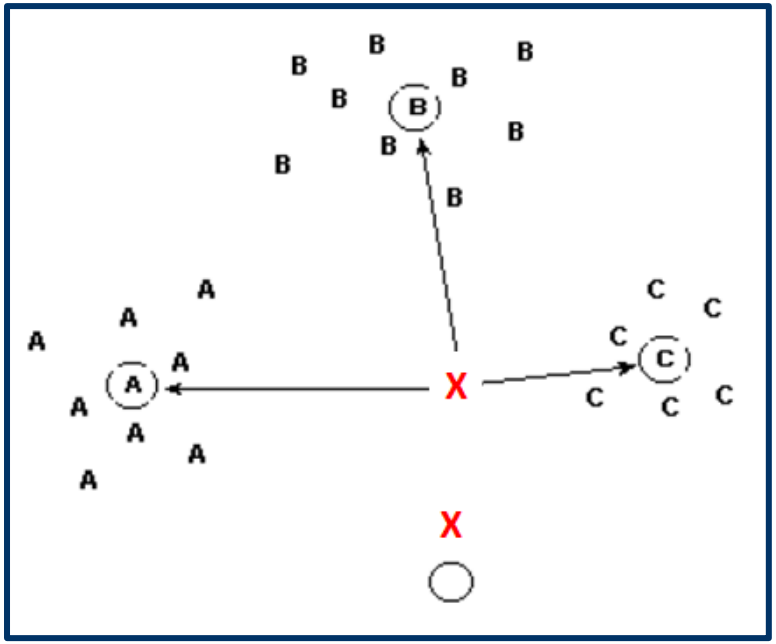
class 1	class 2	class 3	
< 2.20	2.20 - 3.30	> 3.30	x_1

Objects/samples are thus assigned to classes 1, 2 or 3 based on the value of variable x_1. Once this is used to define classes, it is removed from the classification model.

The classification model is subsequently developed, starting from other variables related to objects, which must be independent on classes.

After the model has been constructed, an unknown object, not belonging to the training set, can be assigned to one of the classes.

The most natural criterion for classification consists in assigning this object (X) to the class whose centroid is closest to the obiect, as shown for a bidimensional space in the figure on the right, where class centroids are indicated by circles.

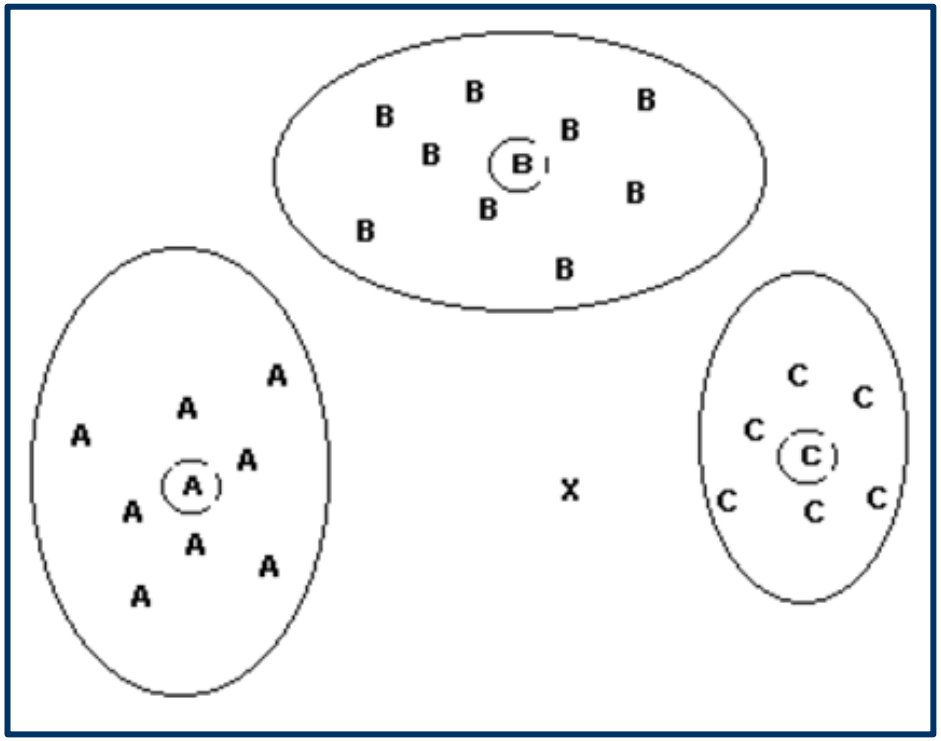


Classification methods can be distinguished in non-modelling and modelling ones.

Modelling methods produce a model able to define the borders of each class, i.e., the dimensions of a space enclosing all objects belonging to that class.

They are drawn as ellipses in the figure on the right.

Note that object X in the figure cannot be assigned to any of the three classes, since it is external to their borders.



Evaluation of a classification method: confusion matrix

The so-called confusion matrix enables the evaluation of a classification method, based on its ability to assign objects correctly to classes.

In the matrix, actual classes, those adopted in the training set, are represented by rows, whereas assigned classes are represented by columns:

assigned classes

actual classes	A'	B'	C'	n_g
A	9	1	0	10
B	2	8	2	12
C	1	2	5	8
$n_{g'}$	12	11	7	$n = 30$

In the example shown in the figure, 30 objects are distributed among classes A (10), B (12) and C (8).

Numbers reported along the main diagonal of the matrix represent objects classified correctly, thus 9 (out of 10) for class A, 8 (out of 12) for class B and 5 (out of 8) for class C.

Numbers located outside the main diagonal represent objects that, although belonging to a certain class, are erroneously assigned to another class.

	assigned classes			
actual classes	classes	A'	B'	C'	n_g
	A	9	1	0	10
	B	2	8	2	12
	C	1	2	5	8
	$n_{g'}$	12	11	7	$n = 30$

Consequently:

1 object of class A has been assigned to class B;
2 objects of class B have been assigned to class A and 2 objects to class C
1 object of class C has been assigned to class A and 2 objects to class B

The final number of each row, n_g, corresponds to the number of objects originally present in a specific class.
The final number of each column, $n_{g'}$, corresponds to the number of objects assigned to a specific class based on the calculated model.

A parameter that can summarize in a simple way the result of a classification procedure is the correct classifications percentage, or, non-error rate, NER%, defined as follows:

$$NER\% = \frac{\sum_g NER\%_g}{G} \times 100$$

where $NER\%_g$ represent the non-error rates for the different classes and G is the number of classes.

In the specific example:

NER% (A) = 9/10 = 90.0% **NER% (B) = 8/12 = 66.7%** **NER% (C) = 5/8 = 62.5%**

thus:

NER%= [(9/10) + (8/12) + (5/8)]/3 x 100 = 73.05%

A parameter complementary to NER% is the error rate, ER%, defined as 100-NER%. In the specific example ER% = 26.95%.

A priori probability, sensitivity and specificity of a class

If specific indications are not available, two equations can be adopted to assign *a priori* probabilities to classes, P_g:

$$P_g = \frac{1}{G} \qquad\qquad P_g = \frac{n_g}{n}$$

where n is the total number of objects.

In the first case the same probability is assigned to each class, without considering the corresponding number of objects.

In the second case the probability corresponds to the ratio between the number of objects in a specific class and the total number of objects.
This definition obviously leads to low probabilities for classes including a few objects.

The sensitivity of a class is defined as the percentual ratio between objects correctly assigned to a certain class, c_{gg}, and the total number of objects actually belonging to that class, n_g:

$$Sn_g = \frac{c_{gg}}{n_g} \times 100$$

The specificity of a class measures the capacity of isolating objects of a certain class from those of other classes; indeed, it corresponds to the percentual ratio between the number of objects correctly assigned to a specific class, c_{gg}, and the total number of objects assigned to that class, n_g':

$$Sp_g = \frac{c_{gg}}{n_{g'}} \times 100$$

In the following table, values of Sn_g% and Sp_g%, calculated from data shown before, are reported:

Classes		*Sn%*	*Sp%*	
A	(9/10)	90.0	75.0	(9/12)
B	(8/12)	66.7	72.7	(8/11)
C	(5/8)	62.5	71.4	(5/7)

It can be easily verified that, if no incorrect assignment is made, sensitivity and specificity are equal to 100% for all classes.

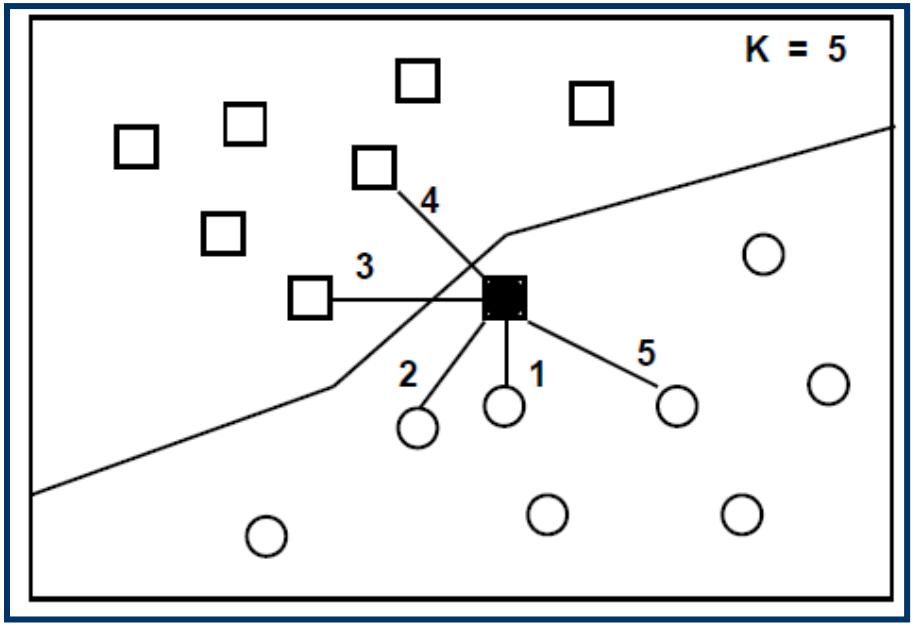
k-nearest neighbours (K-NN) classification method

The k-nearest neighbours classification method, first developed by American statisticians Evelyn Fix and Joseph Hodges in 1951, and then expanded by the American information theorist Thomas Cover in the Sixties, is a non parametric one, i.e., it does not require the knowledge of the distribution of variables.

In this case classification is based on the concept of analogy.
The method considers the distance (usually the euclidean distance) between objects and a selection of an integer number, k, of neighbour objects with respect to the one to be classified.

The algorithm of the K-NN method includes the following steps:

a. data scaling
b. choice of the type of distance to use
c. choice of the number of neighbours, k
d. calculation of the matrix of distances
e1. consideration of the k-nearest neighbour objects for a specific object
e2. assignment of the object to the most represented class in the k neighbours.



Usually, several values of k need to be tried before finding the optimal one, i.e., the one leading to the lowest number of classification errors in the training set.

When the same number of nearest neighbours belonging to different classes is found, the object to be classified is assigned to the class for which the sum of distances between that object and the nearest neighbours belonging to that class is minimum.

The K-NN model is not a mathematical model; it consists in the ensemble of the best k value determined, the type of measure adopted and all objects belonging to the training set.

The prediction of the class for a new object is performed by adding the object to the training set and then evaluating to which class the object is assigned, based on the criterion described before.

The method usually provides good results and is particularly efficient when the borders between classes are non-linear and particularly complex.

A numerical example

Let us consider the table shown on the right, in which the results obtained for four training samples, represented by a special paper tissue, are reported.

Two objective attributes (acid durability and strength) and a classification as bad or good, provided from a survey with customers, were obtained for each sample.

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	Y = Classification
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

In this case the classification of the four samples of the training set is based on the customers' evaluation (bad or good).

The classification problem is expressed as follows:

a new paper tissue whose objective attributes are X1 = 3 and X2 = 7 is produced; the K-NN method is adopted to evaluate if it would be classified as bad or good by customers.

Let us choose k = 3 as the first option.

Since the coordinates of the new object are (3,7), squared Euclidean distances from objects of the training set can be easily calculated and ranked, as shown in the following table:

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	Square Distance to query instance (3, 7)	Rank minimum distance	Is it included in 3-Nearest neighbors?
7	7	$(7-3)^2+(7-7)^2=16$	3	Yes
7	4	$(7-3)^2+(4-7)^2=25$	4	No
3	4	$(3-3)^2+(4-7)^2=9$	1	Yes
1	4	$(1-3)^2+(4-7)^2=13$	2	Yes

As shown in the last column of the table, two of the three nearest neighbours belong to class «good», whereas one belong to class «bad», so the new object is assigned to class «good».

Another example: classifying elements between metals and nonmetals, based on four periodic properties.

The following four periodic properties:

Atomic radius
Ionization energy
Electron affinity
Electronegativity

were considered for 38 elements, divided into two classes: metals (22) and nonmetals (16), according to some versions of the periodic table.

The KNN method was used to assign the elements to the two classes, changing the number of nearest neighbours from 1 to 10.

Table 1 Elements and the four periodic properties[a] selected to run *k*-NN method.

Element	Atomic radius (pm)	Ionization energy ($kJ\,mol^{-1}$)	Electron affinity[b] ($kJ\,mol^{-1}$)	Electronegativity[c]	Group
Metal					
Li	152	519	60	1.00	1
Na	154	494	53	0.93	1
K	227	418	48	0.82	1
Rb	248	402	47	0.82	1
Cs	265	376	46	0.79	1
Fr	270	400	44	0.70	1
Be	113	900	−66[d]	1.60	2
Mg	160	736	−67[d]	1.30	2
Ca	197	590	2	1.30	2
Sr	215	548	5	0.95	2
Ba	217	502	14	0.89	2
Ra[e]	283	509	10	0.90	2
Al	143	577	43	1.60	13
Ga	122	577	29	1.60	13
In	163	556	29	1.80	13
Tl	170	590	19	2.00	13
Ge[f]	122	784	116	2.00	14
Sn	141	707	116	2.00	14
Pb	175	716	35	2.30	14
Sb[f]	141	834	103	2.10	15
Bi	155	703	91	2.00	15
Po[f]	167	812	174	2.00	16
Nonmetal					
H	30	1310	73	2.20	–
B[f]	88	799	27	2.00	13
C	77	1090	122	2.60	14
Si[f]	117	786	134	1.90	14
N	75	1400	−7	3.00	15
P	110	1011	72	2.20	15
As[f]	121	947	78	2.20	15
O	66	1310	141	3.40	16
S	104	1000	200	2.60	16
Se	117	941	195	2.60	16
Te[f]	137	870	190	2.10	16
F	58	1680	328	4.00	17
Cl	99	1255	349	3.20	17
Br	114	1140	325	3.00	17
I	133	1008	295	2.70	17
At	140[g]	1037	270	2.00	17

Vidueira Ferreira et al. Educación Química 26 (2015) 195-201

The following results were obtained, in terms of % Error rate, according to the number of nearest neighbours adopted:

Table 3 Number of misclassifications and error rate for each *k* value.

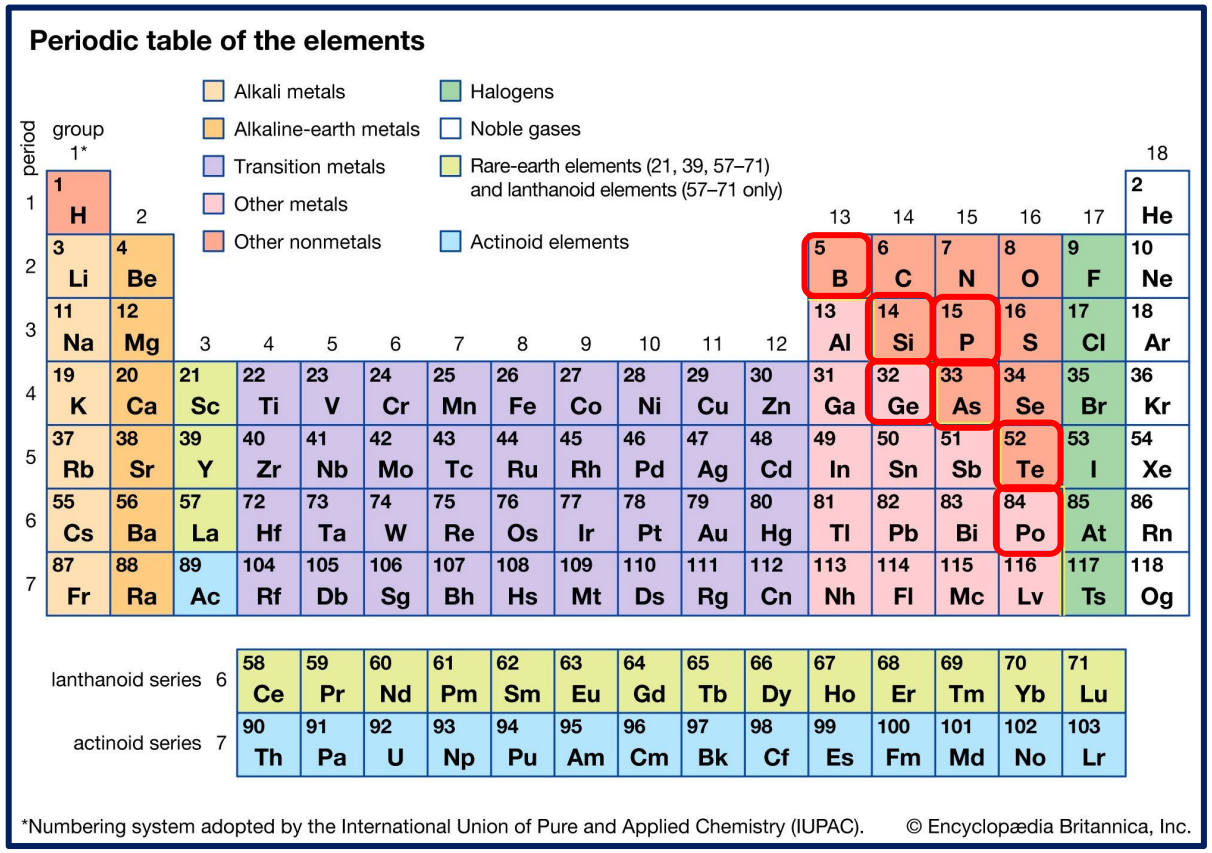
Class	Number of elements	Number of elements misclassified									
		*k*1	*k*2	*k*3	*k*4	*k*5	*k*6	*k*7	*k*8	*k*9	*k*10
Metal	22	2	2	0	0	0	0	1	0	2	0
Nonmetal	16	2	2	3	2	2	2	4	3	3	3
% Error rate		10.5	10.5	7.9	5.3	5.3	5.3	13.2	7.9	13.2	7.9

In particular:

B, Si, P, As and Te were nonmetals misclassified as metals

Ge and Po were metals misclassified as nonmetals

where the initial classifications were based on a periodic table like the one reported on the right.



The table on the right shows which properties of those elements had values compatible with ranges characteristic of the classes in which they were misclassified

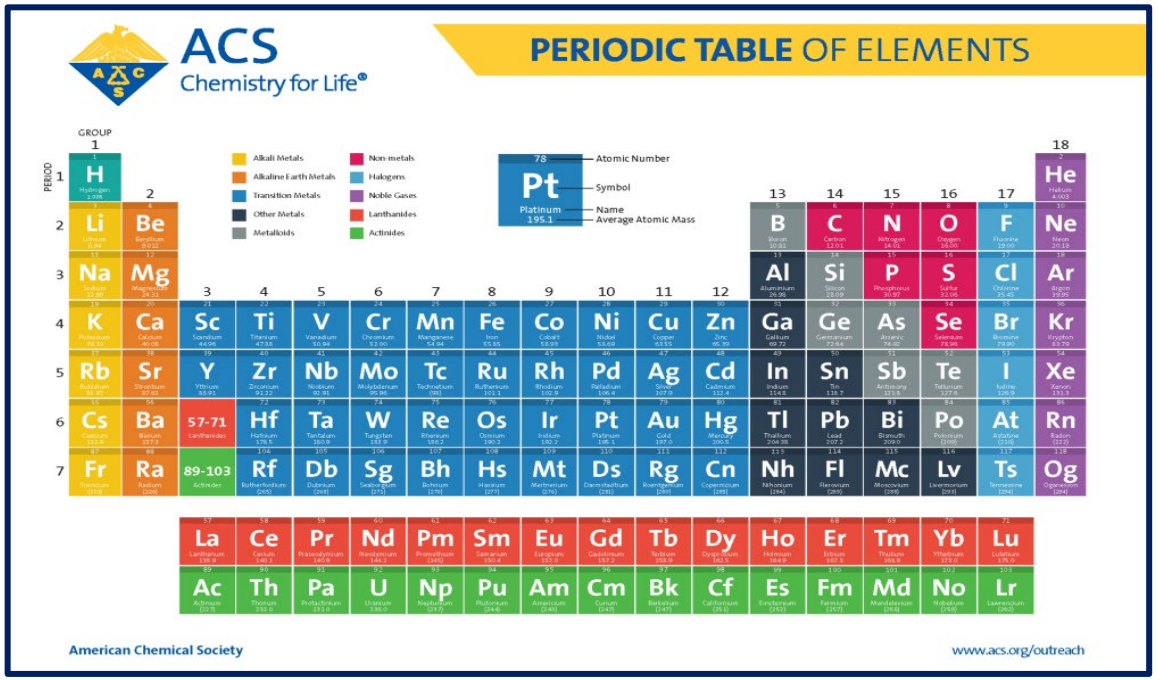
Misclassification was clearly due to the partial overlap of properties values for the two classes for certain elements.

A less misleading version of the periodic table of elements can thus be the one in which most elements misclassified using the KNN method are actually marked as metalloids.

Table 4 Ranges (minimum and maximum) for the periodic properties and the misclassified elements.

	Periodic property			
	Atomic radius (pm)	Ionization energy (kJ mol^{-1})	Electron affinity (kJ mol^{-1})	Electronegativity[a]
	Metals			
	Min: 113; Max: 283	Min: 376; Max: 900	Min: −67; Max: 174	Min: 0.70; Max: 2.30
Nonmetals misclassified as metals	- Si (117) - As (121) Te (137)	B (799) Si (786) - - Te (870)	B (27) Si (134) P (72) As (78) -	B (2.00) Si (1.90) P (2.20) As (2.20) Te (2.10)
	Periodic property			
	Atomic radius (pm)	Ionization energy (kJ mol^{-1})	Electron affinity (kJ mol^{-1})	Electronegativity[a]
	Nonmetals			
	Min: 30; Max: 140	Min: 786; Max: 1680	Min: −7; Max: 349	Min: 1.90; Max: 4.00
Metals misclassified as nonmetals	Ge (122) -	- Po (812)	Ge (116) Po (174)	Ge (2.00) Po (2.00)

[a] Pauling scale.



Discriminant analysis

The term Discriminant Analysis is used to indicate a group of methodologies that, starting from a sampling set of N p-dimensional data **X**, divided into k classes (C_1, C_2, …, C_k), enable the assignment of a generic object to one of the k classes.

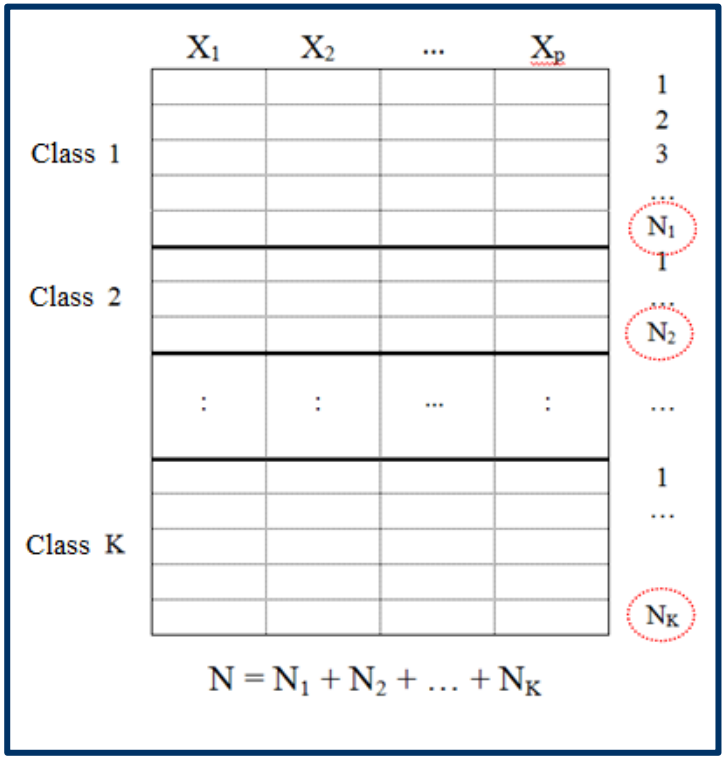
Multivariate discriminant analysis was introduced in 1936 by the British statistician Ronald A. Fisher, while he was studying the assignment of fossil evidences to primates or humanoids starting from measurements taken on them.

The most used methodology for discriminant analysis is Linear Discriminant Analysis (LDA).

The data set typical of LDA is represented by a N × p matrix.

Each row in the matrix represents an object (sample), characterized by p variables (X_1, X_2, …, X_p).

Each class includes N_i rows, each corresponding to an object included in the class.



In the following example, the data matrix arises from 15 apple juice samples, divided into 3 classes including the same number of samples ($N_i = 5$), corresponding to as many varieties.

Each sample is described by the concentrations (g L^{-1}) of sucrose, glucose, fructose and sorbitol (one of the names used to indicate the alditol corresponding to glucose), thus $p = 4$.

Variety	Sucrose	Glucose	Fructose	Sorbitol
A	20	6	40	4.3
A	27	11	49	2.9
A	26	10	47	2.5
A	34	5	47	2.9
A	29	16	40	7.2
B	6	26	49	3.8
B	10	22	47	3.5
B	14	21	51	6.3
B	10	20	49	3.2
B	8	19	49	3.5
C	8	17	55	5.3
C	7	21	59	3.3
C	15	20	68	4.9
C	14	19	74	5.6
C	9	15	57	5.4

The questions to which LDA has to answer are:

1) do the four classes, defined *a priori*, differ also with respect to values assumed by the four explicative variables?

2) If so, is it possible to define a decision rule applicable to a new object, whose class is unknown?

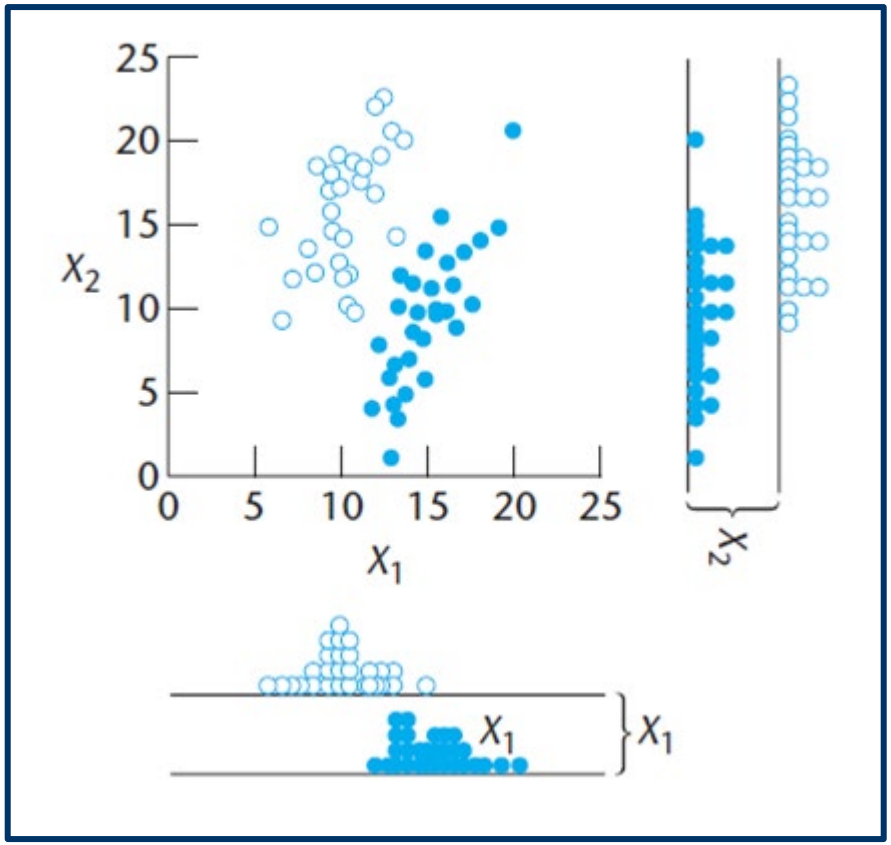
As an example, how should an apple juice with concentrations 11, 23, 50 and 3.8 g L^{-1} for sucrose, glucose, fructose and sorbitol, respectively, be classified?

In the specific example, a 4-dimensional space would be required to represent all the original 15 samples, corresponding to the training set, and then verify if the original distribution between classes is confirmed.

Afterwards, the unknown sample should be represented in the same space and its position with respect to those belonging to the three classes should be evaluated.

In order to explain the general procedure, let us consider the simplest case, i.e., two classes of samples described by two variables.

A graphical representation of samples belonging to the two classes is easily obtained in this case:



Projections of variables on the two axes are useful to evaluate the degree of separation between the two classes.

As apparent, the two classes are poorly separated, especially in terms of the X_2 variable.

The identification of class for a new sample would be very difficult in this case.

A method to increase the separability between classes has thus to be found, e.g., by considering the projection of samples in a direction differing from those of the variable axes. This approach is the base of the technique called Linear Discriminant Analysis (LDA).

Linear Discriminant Analysis (LDA)

Let us consider a set of N p-dimensional data, of which N_i belong to class C_i, with i going from 1 to k.

The N × p matrix of data **X** can be reduced to a N × 1 vector **z** through an appropriate linear combination.

As discussed for Principal Components Analysis, this operation can be interpreted, from a geometric point of view, as the projection of a set of points in a p-dimensional space on an axis defined by vector **z**.

Using matricial notation, the operation can be expressed through the following equation:

$$\boldsymbol{z} = \boldsymbol{X}\ \boldsymbol{w} \qquad \text{where} \qquad \boldsymbol{w} = (w_1, w_2, \ldots, w_p)^T$$

$(N \times 1) = (N \times p) \cdot (p \times 1)$

w represents the vector of weights given to each variable in the linear combination.

Given the i-th object (sample), described by p variables, the operation leads to the following scalar, with $\mathbf{x}_i$ representing the i-th row of the **X** matrix transformed into a column vector:

$$z_i = \boldsymbol{w}^T \mathbf{x}_i = w_1 x_{i1} + w_2 x_{i2} + \ldots + w_p x_{ip}$$

$1 = (1 \times p) \cdot (p \times 1)$

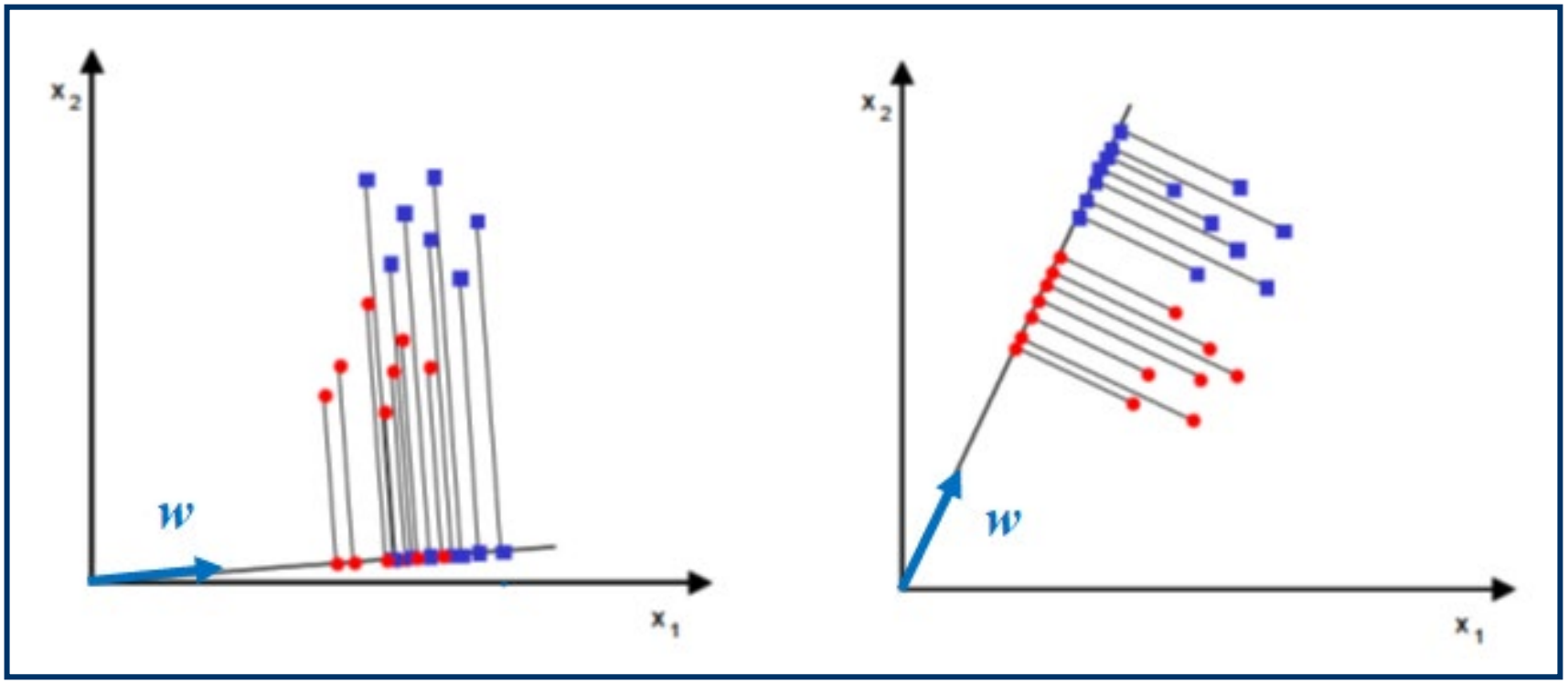
If vector **w** norm is equal to 1 (i.e., **w** represents a normalized weights vector), the linear combination corresponds to a projection of data on a line whose direction is indicated by vector **w**, passing through the origin of axes.

The choice of vector **w** has to satisfy some criteria:

1) weights w_1, w_2, …, w_p should be chosen so that the distribution of objects $\mathbf{x}_i$ between classes is reproduced by scalars z_i in the best possible way;

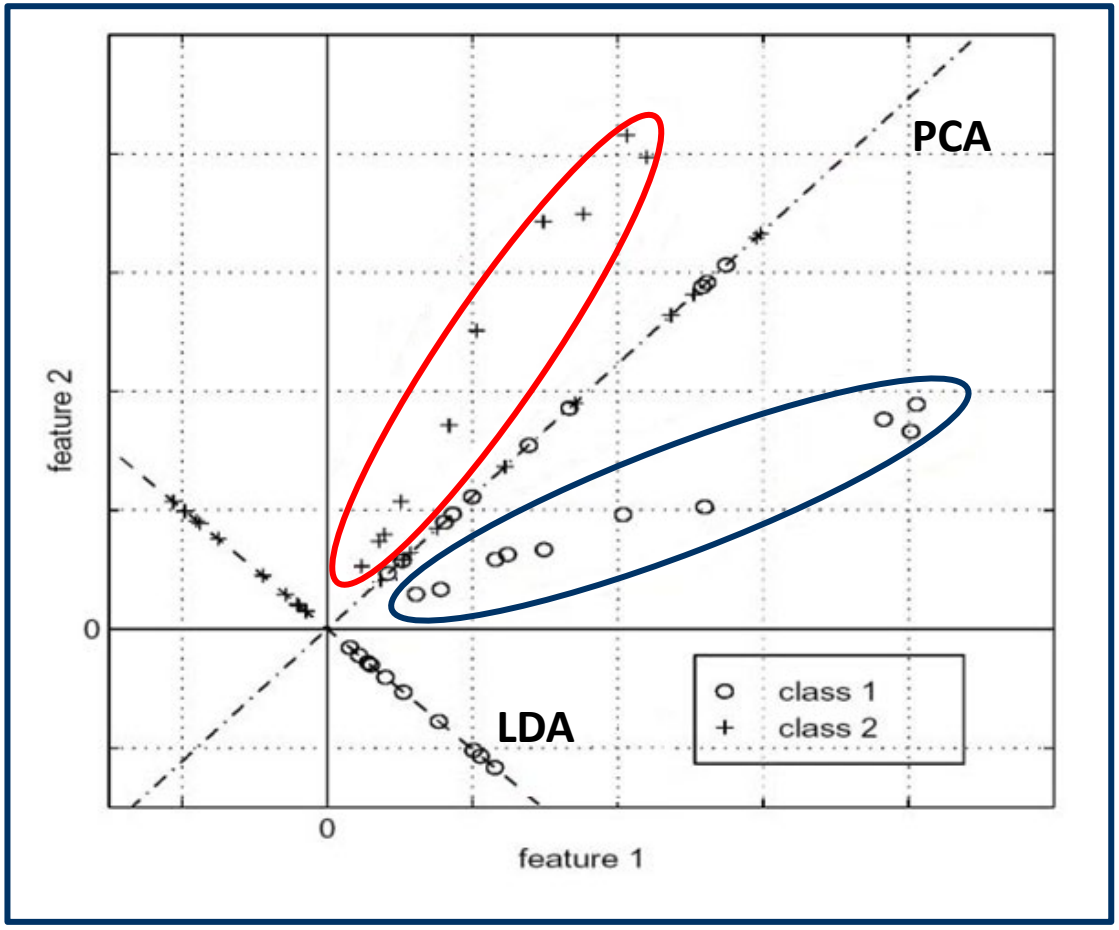
2) the separation between scalars z_i belonging to different classes should be maximized, aiming at the best possible discrimination between classes, thus enabling a reliable assignment of a new object (sample) to its class.

Let us consider the simplest case, represented by objects described by two variables and distributed between two classes.
In the following figure, two different vectors **w** are represented:



As apparent, only the direction adopted for vector **w** in the right panel is able to fulfil the two requirements described before, since the projections referred to objects beloging to the same class are close and the two groups of projections referred to the respective classes are well separated.

An interesting comparison can be made between Principal Component Analysis (PCA) and (Fisher) Linear Discriminant Analysis (LDA):



As apparent, the maximum variance direction, which is prioritary for PCA, is totally inadequate for a classification of the two objects, which is the goal of LDA.

Actually, the minimum variance direction enables the best separation between the two classes.

In order to find the best vector **w**, a measure of separation between classes has to be defined. The distance between classes centroids can be adopted.

In the case of two classes including objects defined by two variables, classes centroids co-ordinates are the following:

$$\mu_{11} = \frac{1}{N_1} \sum_{i \in C_1} x_{i1} \qquad \mu_{12} = \frac{1}{N_1} \sum_{i \in C_1} x_{i2}$$

$$\mu_{21} = \frac{1}{N_2} \sum_{i \in C_2} x_{i1} \qquad \mu_{22} = \frac{1}{N_2} \sum_{i \in C_2} x_{i2}$$

where C_1 and C_2 are the set of values assumed by the two variables for the two classes, respectively.

The centroids for each class in the direction obtained using vector **w** are:

$$\bar{z}_1 = \frac{1}{N_1} \sum_{i \in C_1} z_i = \frac{1}{N_1} \sum_{i \in C_1} \mathbf{w}^T \mathbf{x}_i$$

$$\bar{z}_2 = \frac{1}{N_2} \sum_{i \in C_2} z_i = \frac{1}{N_2} \sum_{i \in C_2} \mathbf{w}^T \mathbf{x}_i$$

If column vectors μ_1 and μ_2 are defined as follows:

$$\boldsymbol{\mu}_1 = \begin{pmatrix} \mu_{11} \\ \mu_{12} \end{pmatrix} \qquad \boldsymbol{\mu}_2 = \begin{pmatrix} \mu_{21} \\ \mu_{22} \end{pmatrix}$$

the two centroids can be expressed as inner products between the row vectors of weights adopted for variables and column vectors expressing the centroids of the two classes in the original co-ordinates:

$$\bar{z}_1 = \boldsymbol{w}^T \boldsymbol{\mu}_1$$

$$\bar{z}_2 = \boldsymbol{w}^T \boldsymbol{\mu}_2$$

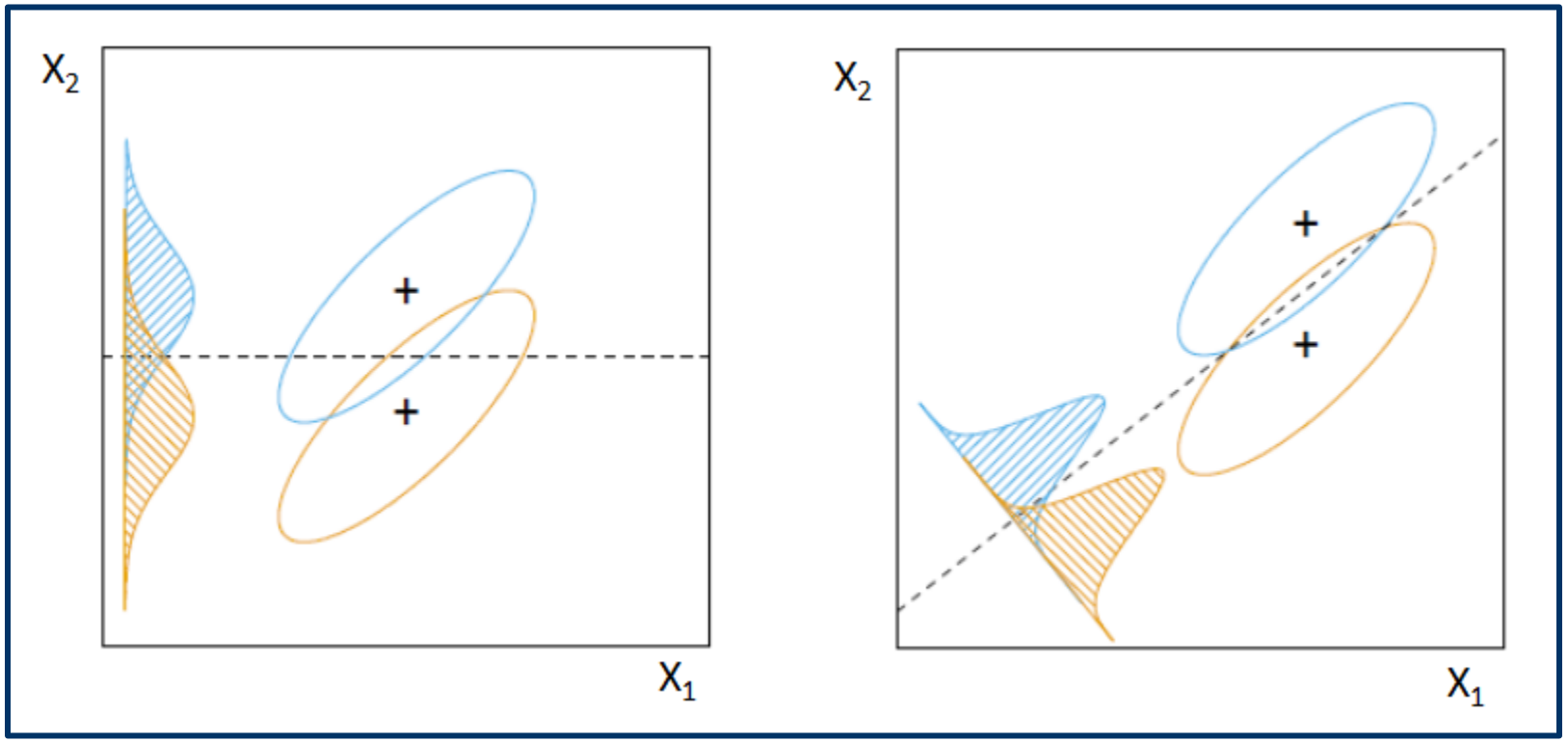
A target parameter, i.e., a quantity to be maximized, could thus be the distance, in absolute value, between centroid projections on vector **w**:

$$J(\boldsymbol{w}) = \left| \bar{z}_1 - \bar{z}_2 \right| = \left| \boldsymbol{w}^T \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 \right) \right|$$

clearly depending on the vector **w** selected.

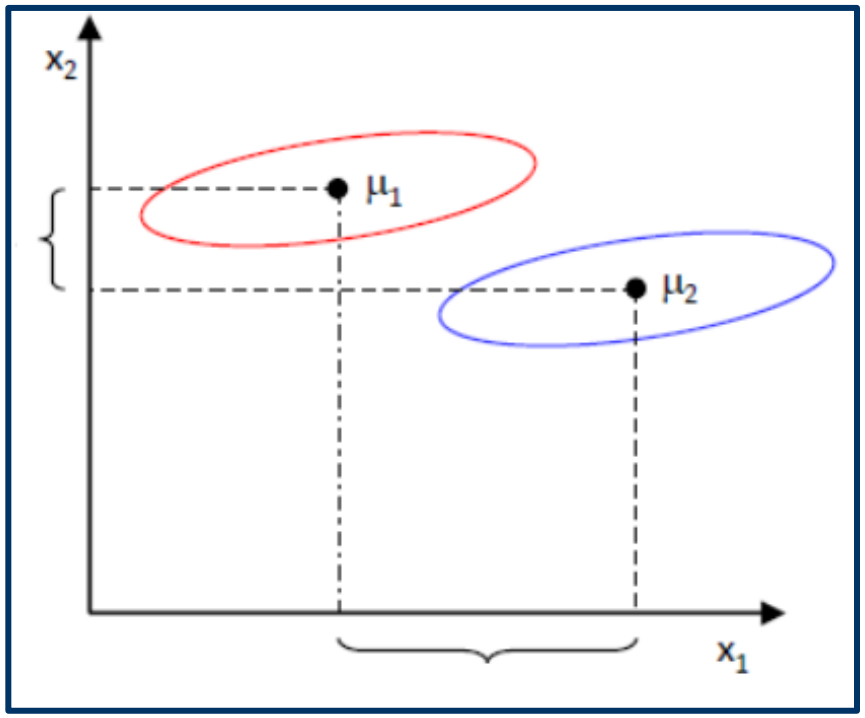
However, the distance between the projections of classes centroids on a specific direction does not take into account the dispersion of objects around centroids.

In the following figure the distributions of objects projections in specific directions are represented by Gaussian functions with equal variance, thus centroids correspond to the Gaussian maxima:



As apparent in the left panel, although direction X_2 enables a good separation between centroids, the overlap between classes projections is remarkable along it.
The direction minimizing the overlap betwen classes is the one shown in the right panel.

In the following further example, the separation between classes is better along direction x_2, although the separation of classes centroids is worse than that observed along direction x_1:



This effect is due to the fact that the spread of objects in the two classes is more remarkable along direction x_1.

For this reason, Fisher proposed an approach based on the maximizaton of centroid distance normalized by a measure of class dispersion (like the within-class scatter).

Once a direction is selected for data projection, scatter is defined, for each class, as:

$$\tilde{s}_i^2 = \sum_{z \in C_i} \left(z - \bar{z}_i\right)^2$$

If two classes are present, the following quantity:

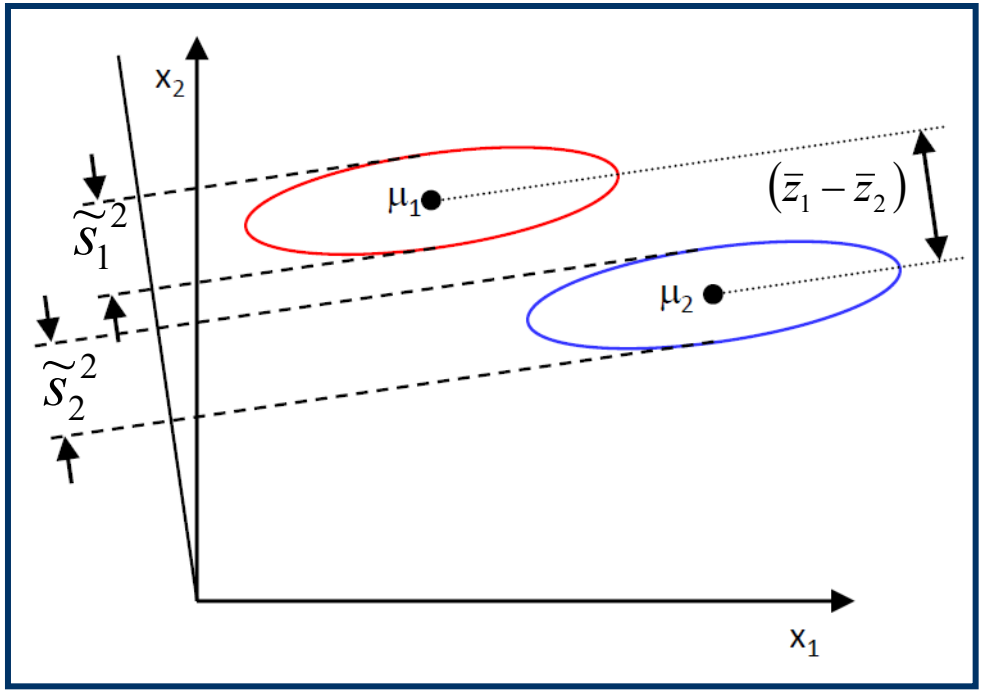
$$\left(\tilde{s}_1^2 + \tilde{s}_2^2\right)$$

is defined as the within-class scatter of projected samples.

The Fisher linear discriminant is thus defined as:

$$J(w) = \frac{\left|\bar{z}_1 - \bar{z}_2\right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

The LDA procedure tries to maximize this quantity by choosing an appropriate vector **w**.



Notably, J(**w**) can be expressed using an alternative equation:

$$J(w) = \frac{w^{T} S_{B} w}{w^{T} S_{W} w}$$

where S_B represents the between-class scatter matrix and $S_w = S_1 + S_2$ represents the within-class scatter matrix.

First, each S_i can be expressed as follows:

$$\mathbf{S_i} = \sum_{x \in C_i} (\mathbf{x} - \boldsymbol{\mu_i})(\mathbf{x} - \boldsymbol{\mu_i})^{T}$$

consequently:

$$\tilde{s}_i^2 = \sum_{z \in C_i} (z - \bar{z}_i)^2 = \sum_{x \in C_i} \left(w^{T}\mathbf{x} - w^{T}\boldsymbol{\mu_i} \right)^2 = \sum_{x \in C_i} \left(w^{T}\mathbf{x} - w^{T}\boldsymbol{\mu_i} \right)\left(w^{T}\mathbf{x} - w^{T}\boldsymbol{\mu_i} \right)^{T}$$

$$= \sum_{x \in C_i} w^{T}(\mathbf{x} - \boldsymbol{\mu_i})(\mathbf{x} - \boldsymbol{\mu_i})^{T} w = w^{T} \mathbf{S_i} w$$

thus:

$$\tilde{s}_1^2 + \tilde{s}_2^2 = w^{T} \mathbf{S_W} w$$

The numerator of the J(**w**) quantity can be expressed as follows:

$$(\bar{z}_1 - \bar{z}_2)^2 = \left(w^T \mu_1 - w^T \mu_2 \right)^2 = w^{\mathrm{T}} (\mu_1 - \mu_2)(\mu_1 - \mu_2)^{\mathrm{T}} w = w^{\mathrm{T}} \mathbf{S}_{\mathbf{B}} w$$

The following equation can thus be written:

$$J(w) = \frac{\left| \bar{z}_1 - \bar{z}_2 \right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{w^{\mathrm{T}} \mathbf{S}_{\mathbf{B}} w}{w^{\mathrm{T}} \mathbf{S}_{\mathbf{W}} w}$$

moreover, the total scatter is defined as:

$$\mathbf{S}_T = \sum_i (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^{\mathrm{T}}$$

where the sum is now extended to all samples, $\mathbf{x}_i$ is a vector containing variable values for the i-th sample and $\boldsymbol{\mu}$ is a vector containing average values for all variables.

Since $\mathbf{S}_T = \mathbf{S}_W + \mathbf{S}_B$, J(**w**) can be expressed also as: $J(w) = \dfrac{w^{\mathrm{T}} \mathbf{S}_T w}{w^{\mathrm{T}} \mathbf{S}_{\mathbf{W}} w} - 1$

Maximization of the J(w) function

An important property of the J(**w**) function is its invariance with respect to a re-scaling of vector **w**, i.e., it remains identical when a new vector, obtained by multiplying vector **w** by a scalar α, is considered.

A specific vector **w**, able to fulfil the equality $\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\boldsymbol{w}=1$, can be thus selected and the maximization of J(**w**) corresponds to the maximization of $\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\boldsymbol{w}$.

By analogy with the procedure followed for Principal Component Analysis, the problem can be solved using the approach of Lagrangian multipliers, i.e., by maximizing the following function:

$$L(\boldsymbol{w},\lambda)=\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\boldsymbol{w}-\lambda(\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\boldsymbol{w}-1)$$

The maximization corresponds to solving the following system:

$$\begin{cases}\dfrac{\partial L(\boldsymbol{w},\lambda)}{\partial \boldsymbol{w}^{\mathrm{T}}}=2\mathbf{S}_{\mathrm{B}}\boldsymbol{w}-2\lambda\mathbf{S}_{\mathrm{W}}\boldsymbol{w}=0\\ \dfrac{\partial L(\boldsymbol{w},\lambda)}{\partial \lambda}=\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\boldsymbol{w}-1=0\end{cases}$$

$$\Rightarrow\quad\begin{cases}\mathbf{S}_{B}\boldsymbol{w}=\lambda\mathbf{S}_{w}\boldsymbol{w}\\ \boldsymbol{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\boldsymbol{w}=1\end{cases}$$

Each member of the first equation of the system can be multiplied by $\mathbf{S_w^{-1}}$, thus obtaining the following equation:

$$\mathbf{S}_{\mathbf{w}}^{-1}\mathbf{S}_B\,\boldsymbol{w} = \lambda \boldsymbol{w}$$

The maximization of the J(**w**) function can thus be considered as the solution of an eigenvalue problem, were λ is an eigenvalue of $\mathbf{S_w^{-1}S_B}$ and **w** is the corresponding eigenvector.

Once **w** has been determined, the linear discriminant function is: $z = \mathbf{w}^T \mathbf{x}$

If objects are represented by two variables (p = 2) this equation can be written as:

$$z = w_1 X_1 + w_2 X_2$$

where w_1 and w_2 are defined Fisher non standardized coefficients.

If variables are expressed using different units of measurement it is better to consider Fisher standardized coefficients, which are obtained by considering standardized values of variables.

In any case, larger coefficients are related to variables with a higher discriminating capacity.

A numerical example of LDA with two classes

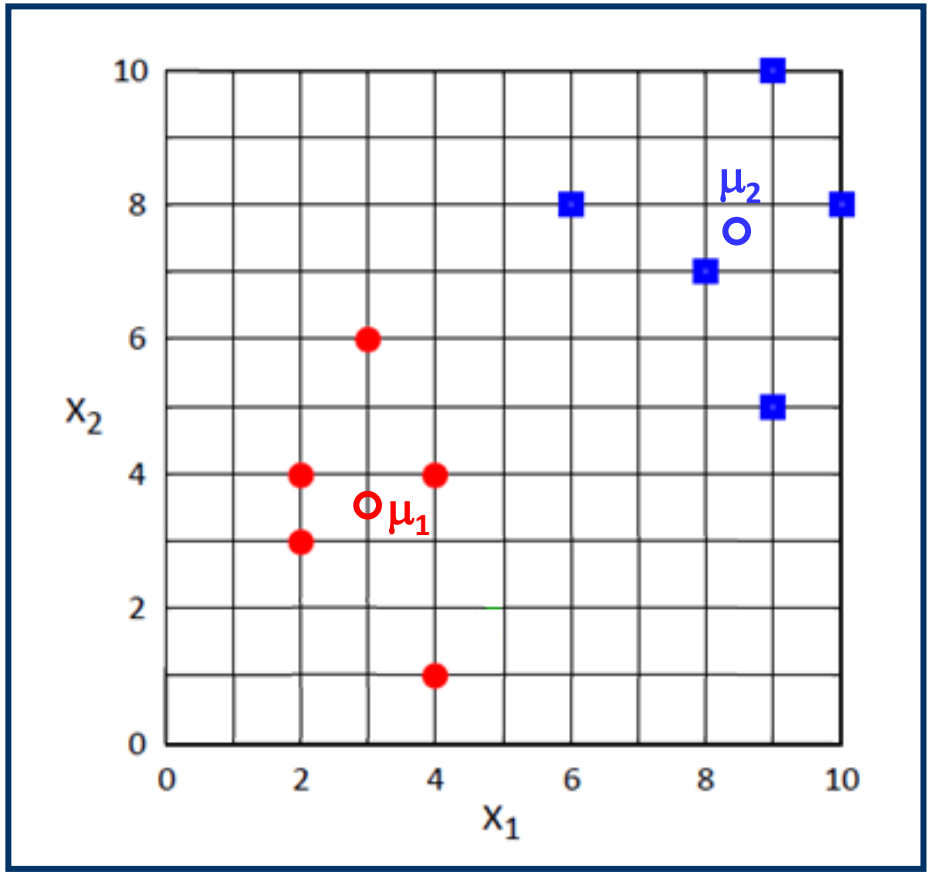
Let us consider bidimensional data reported in the figure on the right, originally divided into two classes:

Class 1

x_1	x_2
4	1
2	4
2	3
3	6
4	4

Class 2

x_1	x_2
9	10
6	8
9	5
8	7
10	8



Class centroids are:

$\mu_1 = [3.0 \;\; 3.6]^T$ $\mu_2 = [8.4 \;\; 7.6]^T$

Since scatter matrices are defined as:

$$\mathbf{S_i} = \sum_{x \in C_i} (\mathbf{x} - \boldsymbol{\mu_i})(\mathbf{x} - \boldsymbol{\mu_i})^{\mathbf{T}}$$

their calculation implies the consideration, for each of the two classes, of the $(\mathbf{x}\text{-}\mu_i)$ matrix and its transpose. As an example, the two matrices for Class 1 are:

$\mathbf{x} - \mu_1$

1	-2.6
-1	0.4
-1	-0.6
0	2.4
1	0.4

$(\mathbf{x} - \mu_1)^T$

1	-1	-1	0	1
-2.6	0.4	-0.6	2.4	0.4

The resulting $\mathbf{S_1}$ matrix is:

4	-2
-2	13.2

In the case of Class 2 the two matrices required for the calculation of S_2 matrix are:

$x - \mu_2$

0.6	2.4
-2.4	0.4
0.6	-2.6
-0.4	-0.6
1.6	0.4

$(x - \mu_2)^T$

0.6	-2.4	0.6	-0.4	1.6
2.4	0.4	-2.6	-0.6	0.4

The resulting S_2 matrix is:

9.2	-0.2
-0.2	13.2

The following step of the calculation implies the sum of S_1 and S_2 to obtain S_W:

S_1

4	-2
-2	13.2

+

S_2

9.2	-0.2
-0.2	13.2

=

S_W

13.2	-2.2
-2.2	26.4

The inverse of S_W matrix, indicated as $(S_W)^{-1}$, is:

0.077	0.006
0.006	0.038

The next step of calculation is finding the between-class scatter matrix, S_B:

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

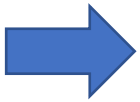
In the specific case, the calculation is:

$(\mu_1 - \mu_2)$

-5.4
-4

$(\mu_1 - \mu_2)^T$

-5.4	-4



$(\mu_1 - \mu_2)\ (\mu_1 - \mu_2)^T$

29.16	21.6
21.6	16

The inverse of S_W matrix and S_B matrix can now be multiplied:

$(S_W)^{-1}$

0.077	0.006
0.006	0.038

$\times$

S_B

29.16	21.6
21.6	16

$=$

$(S_W)^{-1}\ S_B$

2.375	1.759
0.996	0.738

For example: $2.375 = 0.077 \times 29.16 + 0.006 \times 21.6$

Starting from the equation $\mathbf{S}_w^{-1}\mathbf{S}_B\,\boldsymbol{w} = \lambda \boldsymbol{w}$ ⇒ $\mathbf{S}_w^{-1}\mathbf{S}_B - \lambda \boldsymbol{I} = 0$

the following equation can be written (remembering that ***I*** = $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$)

$$\begin{bmatrix} 2.375 - \lambda & 1.759 \\ 0.996 & 0.738 - \lambda \end{bmatrix} = 0 \Rightarrow \lambda^2 - 3.113\,\lambda + (2.375*0.738 - 1.759*0.996) = \lambda\,(\lambda - 3.113) = 0$$

The solutions of this equation are the eigenvalues:

$\lambda_1 = 3.113$ and $\lambda_2 = 0$

The eigenvector can be obtained by reconsidering the equation $\mathbf{S}_w^{-1}\mathbf{S}_B\,\boldsymbol{w} = \lambda \boldsymbol{w}$

and the non zero eigenvalue, i.e., $\lambda_1 = 3.113$:

$$\begin{bmatrix} 2.375 & 1.759 \\ 0.996 & 0.738 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = 3.113 \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

The following system is thus obtained:

$$\begin{cases} 2.375\, w_1 + 1.759\, w_2 = 3.113\, w_1 \\ 0.996\, w_1 + 0.738\, w_2 = 3.113\, w_2 \end{cases} \Rightarrow \begin{matrix} w_1 \\ w_2 \end{matrix} = \begin{matrix} 2.34 \\ 1.0 \end{matrix}$$

Vector **w** can be normalized to its norm, which is: $\sqrt{(2.34)^2 + (1)^2} = 2.545$

Its components thus become: $w_1 = 2.34/2.545 = 0.920$ and $w_2 = 1.0/2.545 = 0.393$

Finally, the scalar $z = \mathbf{w}^T \mathbf{x}$ can be obtained:

$z = 0.920\, X_1 + 0.393\, X_2$

This equation represents the linear discriminant function for the problem under consideration.

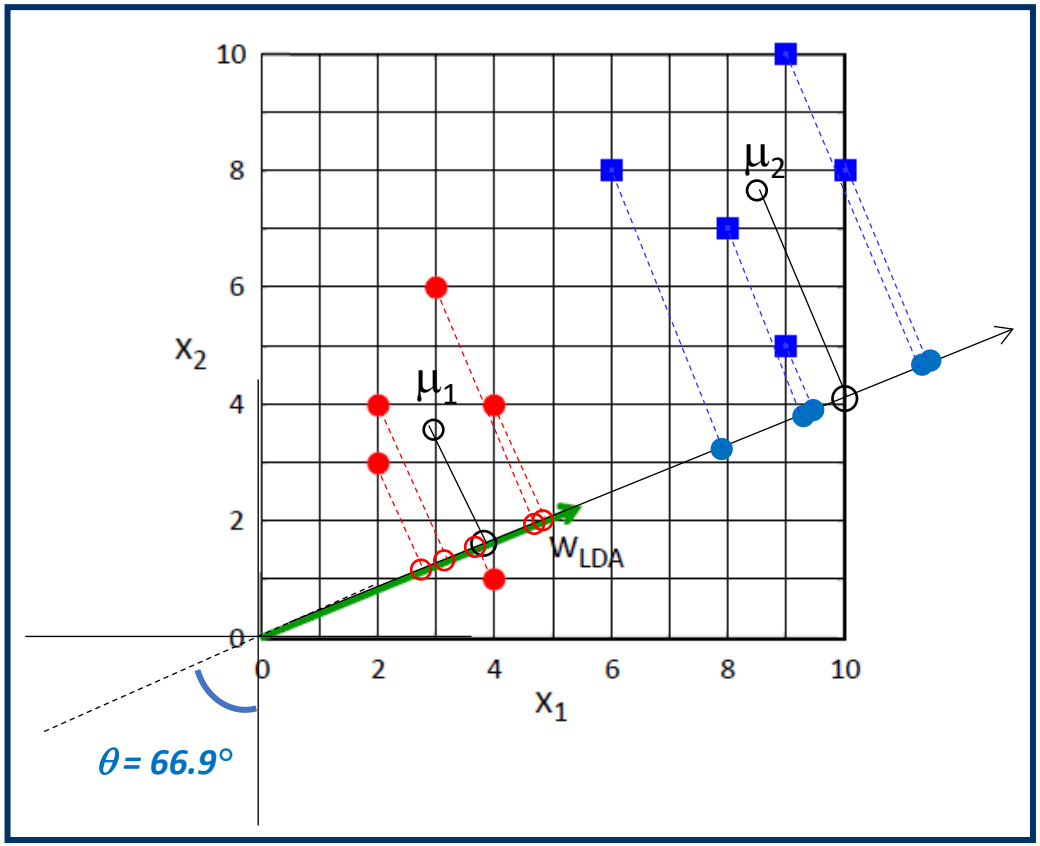
From a geometrical point of view, the equation for z represents a specific direction on the X_1, X_2 plane.

In particular, it represents a line passing through the origin and forming an angle θ of 66.9° with the vertical axis.

Indeed:

$0.920/0.393 = 2.34 = \tan \theta$

$\theta = 66.9°$



As evidenced in the figure, the direction represented by the equation for z enables the best separation between the projections of objects belonging to the two classes.

Classification of a new object using LDA

Once values of projections are found for objects included in the training set, the classification of a new object can be based on the calculation of its projection, z_i, on the direction of vector **w**, starting from the corresponding vector $\mathbf{x}_i$:

$$z_i = \mathbf{w}^T \mathbf{x}_i$$

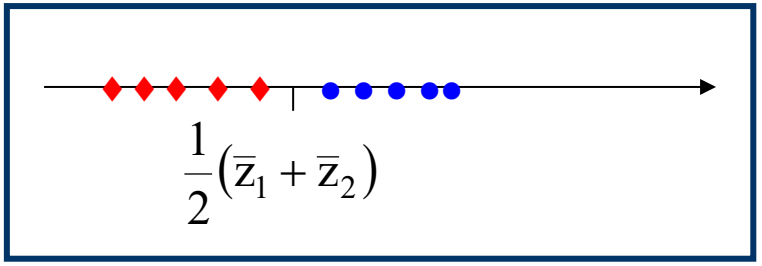
The resulting z_i is compared with z values corresponding to the centroids of classes; the object will be related to the class whose centroid is closest to its projection.
As an example, if two classes are present the new object will be related to Class 1 if:

$$\left| z_i - \bar{z}_1 \right| < \left| z_i - \bar{z}_2 \right|$$

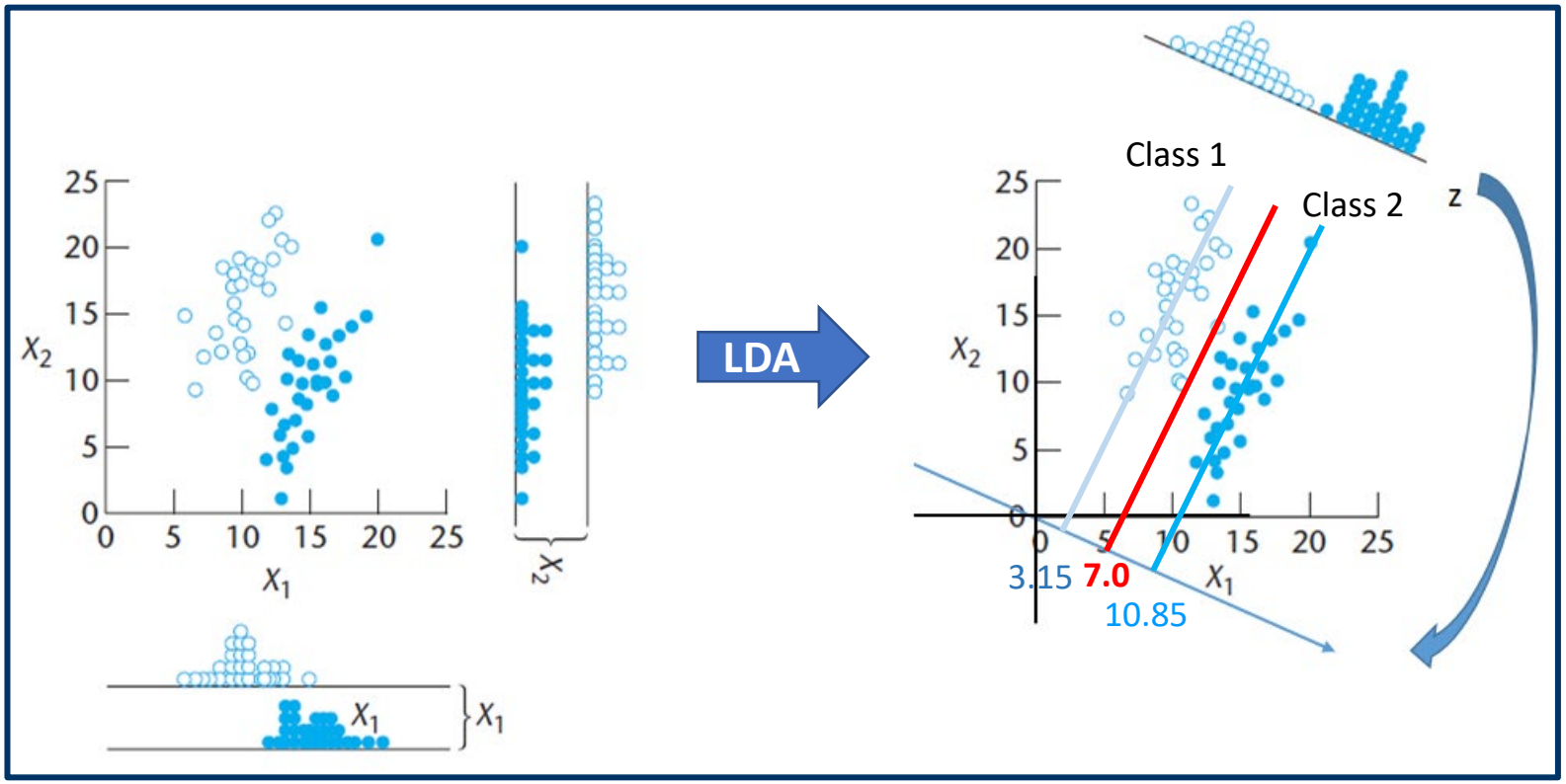
or, equivalently:

$$z_i < \frac{1}{2}\left(\bar{z}_1 + \bar{z}_2\right)$$

This relationship can be easily visualized in geometrical terms by representing projections of objects on an horizontal line corresponding to the direction of vector **w**, as shown in the figure on the right.



If a previously described dataset is re-considered and the output of LDA is shown:



The following values are obtained for class centroids projections:

$$\bar{z}_1 = 3.15 \qquad \bar{z}_2 = 10.85$$

Since the average value of the two centroids projections is equal to 7.0, a new object will be assigned to Class 1 if its projection is lower than 7.0, otherwise it will be assigned to Class 2.

Linear Discriminant Analysis (LDA) with more than two classes

When more than 2 classes are involved in LDA (k > 2) k-1 projection vectors $\mathbf{w}_i$, arranged in columns in a projection matrix **W**, are considered:

$$\mathbf{z} = \mathbf{W}^{\mathrm{T}}\mathbf{x}$$

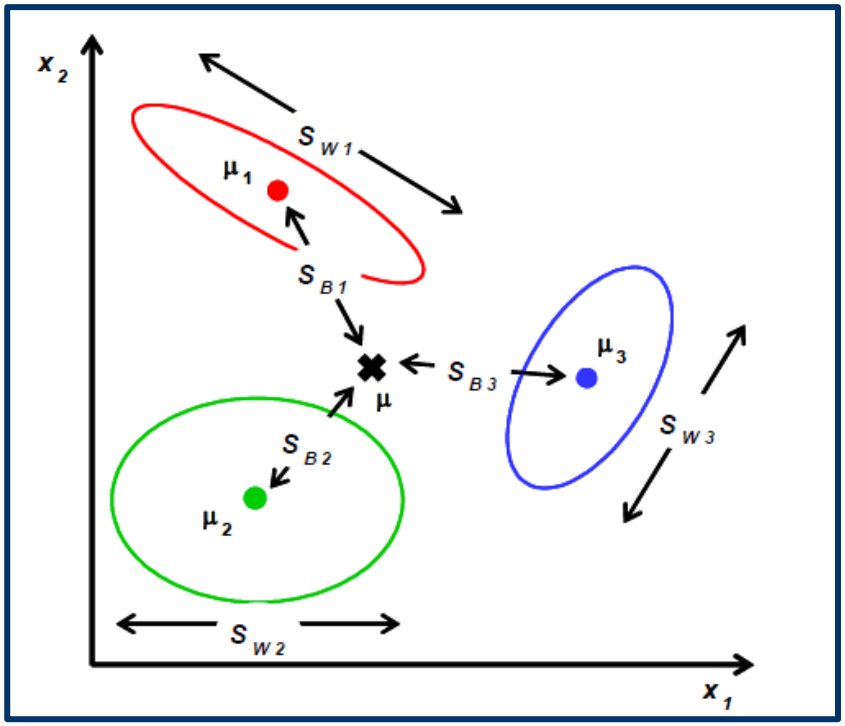
In this case within-class, $\mathbf{S}_W$, and between-class, $\mathbf{S}_B$, scatter matrices are defined as follows:

$$\mathbf{S}_W = \sum_{i=1}^{k} \mathbf{S}_{W_i} = \sum_{i=1}^{k} \sum_{x \in C_i} (\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^{\mathrm{T}}$$

where $\boldsymbol{\mu}_i$ is a vector containing the co-ordinates of the i-th class centroid.

$$\mathbf{S}_B = \sum_{i=1}^{k} \mathbf{S}_{B_i} = \sum_{i=1}^{k} N_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^{\mathrm{T}}$$

where $\boldsymbol{\mu}$ is a vector containing the co-ordinates of the centroid of class centroids.



In the figure $\mathbf{S}_{W_i}$ matrices, contributing to $\mathbf{S}_W$, and $\mathbf{S}_{B_i}$ matrices, contributing to $\mathbf{S}_B$, are represented in geometrical terms for a system including three classes and based on two variables.

The corresponding scatter matrices for projections are:

$$\tilde{\mathbf{S}}_W = \sum_{i=1}^{k} \sum_{z \in C_i} (\mathbf{z} - \overline{\mathbf{z}}_\mathbf{i})(\mathbf{z} - \overline{\mathbf{z}}_\mathbf{i})^\mathbf{T} \quad \text{where:} \quad \overline{\mathbf{z}}_\mathbf{i} = \frac{1}{N_i} \sum_{z \in Ci} \mathbf{z}$$

$$\tilde{\mathbf{S}}_B = \sum_{i=1}^{k} N_i (\overline{\mathbf{z}}_\mathbf{i} - \overline{\mathbf{z}})(\overline{\mathbf{z}}_\mathbf{i} - \overline{\mathbf{z}})^\mathbf{T} \quad \text{where:} \quad \overline{\mathbf{z}} = \frac{1}{N} \sum_{\forall z} \mathbf{z}$$

By analogy with LDA involving two classes, the following equations can be written:

$$\tilde{\mathbf{S}}_\mathbf{W} = \mathbf{W}^\mathbf{T} \mathbf{S}_\mathbf{W} \mathbf{W}$$
$$\tilde{\mathbf{S}}_B = \mathbf{W}^\mathbf{T} \mathbf{S}_B \mathbf{W}$$

$$\mathbf{J(W)} = \frac{\left|\tilde{\mathbf{S}}_\mathbf{B}\right|}{\left|\tilde{\mathbf{S}}_\mathbf{W}\right|} = \frac{\left|\mathbf{W}^\mathbf{T} \tilde{\mathbf{S}}_\mathbf{B} \mathbf{W}\right|}{\left|\mathbf{W}^\mathbf{T} \tilde{\mathbf{S}}_\mathbf{W} \mathbf{W}\right|}$$

In this case determinants of scatter matrices need to be calculated.

The optimal projection **W** can be obtained using an eigenvalue equation:

$$\mathbf{S}_\mathbf{w}^{-1} \mathbf{S}_B \mathbf{W} = \Lambda \mathbf{W}$$

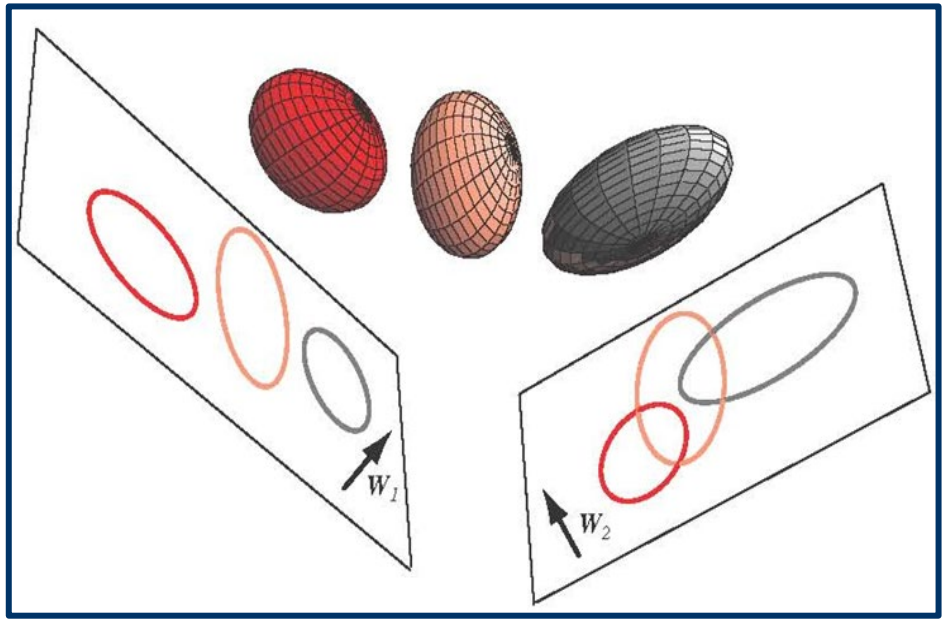
If three classes are considered (k = 3), two of the three possible eigenvalues are different from zero.

The largest eigenvalue, λ_1, is considered first and the corresponding eigenvector ($\mathbf{w}_1$), and then also the first linear discriminating function, that is the one providing the better separation of classes, are obtained.

The second eigenvalue, λ_2, lower than λ_1, is considered subsequently, thus eigenvector $\mathbf{w}_2$ and the second linear discriminating functon are obtained.

It is worth noting that a further constraint has to be introduced when calculating $\mathbf{w}_2$, since $\mathbf{w}_1$ and $\mathbf{w}_2$ must be incorrelated.

A graphical representation referred to a system including three classes and based on three variables is reported in the figure on the right.



In this case three-dimensional classes are projected onto planes, whose normals are represented by vectors $\mathbf{w_1}$ and $\mathbf{w_2}$.

As apparent, a better distinction between classes is observed on the plane normal to $\mathbf{w_1}$.

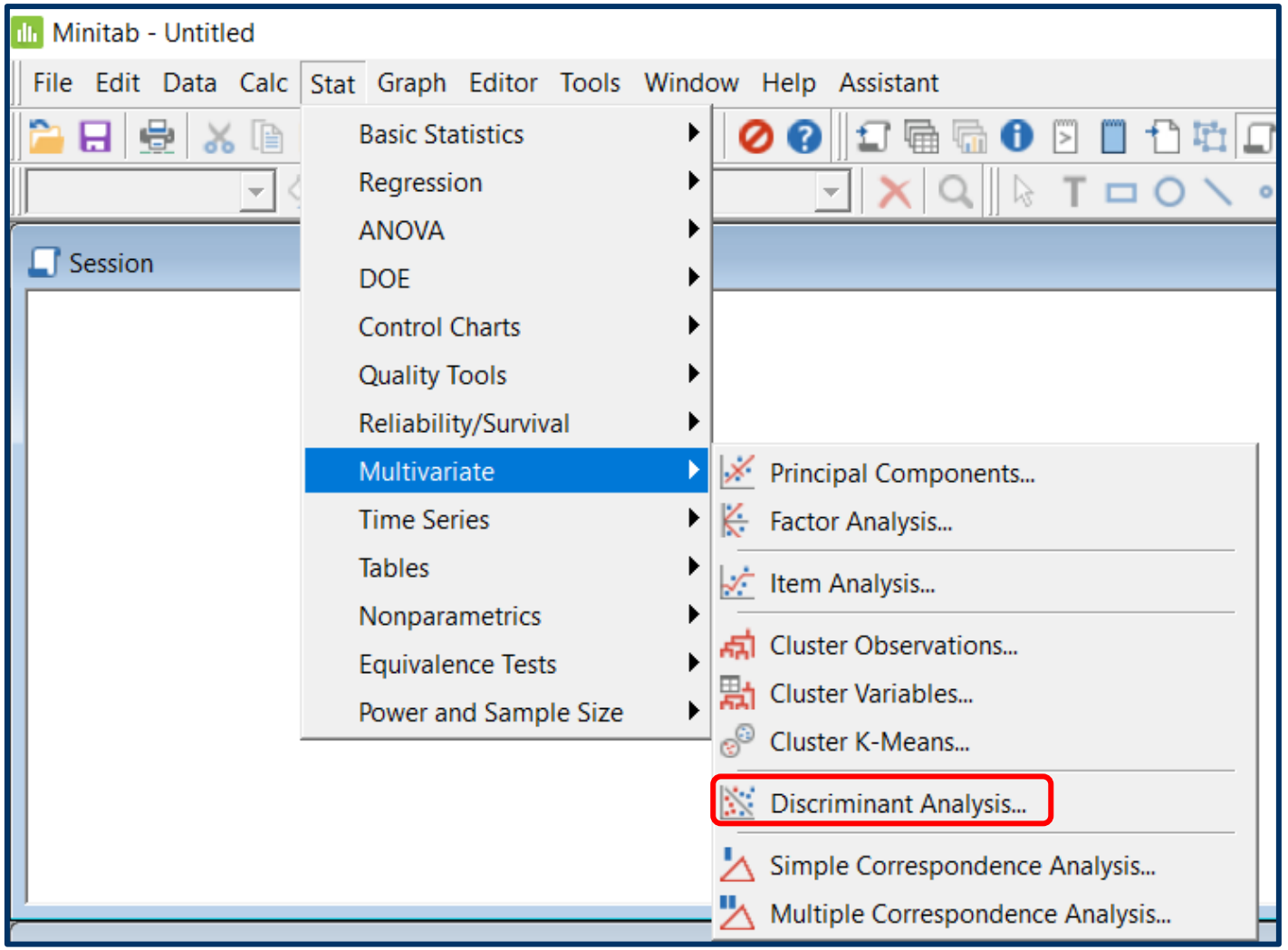
A numerical example of LDA with three classes performed with Minitab 18

Let us re-consider the dataset referred to apple juices described by concentrations (g L^{-1}) found for four carbohydrates, divided into three classes:

Variety	Sucrose	Glucose	Fructose	Sorbitol
A	20	6	40	4.3
A	27	11	49	2.9
A	26	10	47	2.5
A	34	5	47	2.9
A	29	16	40	7.2
B	6	26	49	3.8
B	10	22	47	3.5
B	14	21	51	6.3
B	10	20	49	3.2
B	8	19	49	3.5
C	8	17	55	5.3
C	7	21	59	3.3
C	15	20	68	4.9
C	14	19	74	5.6
C	9	15	57	5.4

The problem to be solved using LDA is classifying an apple juice containing 11, 23, 50 and 3.9 g L^{-1} of sucrose, glucose, fructose and sorbitol, respectively.

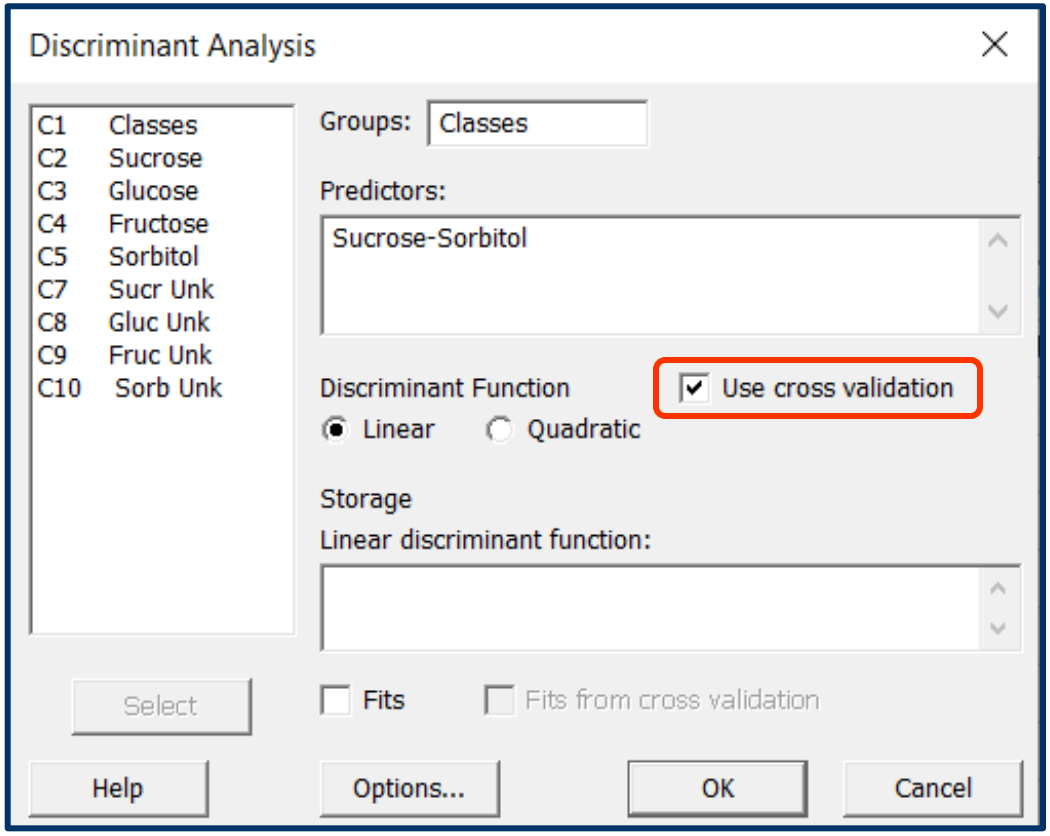
As an example of software application, Minitab 18 has been used to perform LDA, which can be assessed using the following path: Stat > Multivariate > Discriminant Analysis….



In this case, a column of the Worksheet, usually the first, is used to indicate classifications of samples in the training set. The other columns are used for variables:

Worksheet 1 ***

↓	C1-T	C2	C3	C4	C5
	Classes	Sucrose	Glucose	Fructose	Sorbitol
1	A	20	6	40	4.3
2	A	27	11	49	2.9
3	A	26	10	47	2.5
4	A	34	5	47	2.9
5	A	29	16	40	7.2
6	B	6	26	49	3.8
7	B	10	22	47	3.5
8	B	14	21	51	6.3
9	B	10	20	49	3.2
10	B	8	19	49	3.5
11	C	8	17	55	5.3
12	C	7	21	59	3.3
13	C	15	20	68	4.9
14	C	14	19	74	5.6
15	C	9	15	57	5.4



The Discriminant Analysis window is used to select the column indicating classifications in the Groups box, whereas columns referred to variables are selected for the Predictors box.
The program enables the choice between Linear and Quadratic Discriminant Analysis.

Note that four further columns (from C7 to C10) have already been prepared and named as Sucr/Gluc/Fruc/Sorb Unk, since they will be used to enter variables for unknown samples.

The first information obtained as output, included in the Session window of the Minitab software, is the summary of samples belonging to the three classes (groups) and then the confusion matrix.

As apparent, 100% of correct classifications was obtained in the specific case, since all 15 samples were classified in the originally proposed classes.

Groups

Group	A	B	C
Count	5	5	5

Summary of Classification

	True Group		
Put into Group	A	B	C
A	5	0	0
B	0	5	0
C	0	0	5
Total N	5	5	5
N correct	5	5	5
Proportion	1.000	1.000	1.000

Correct Classifications

N	Correct	Proportion
15	15	1.000

Several further results can be found in the Session window:

Squared Distance Between Groups

	A	B	C
A	0.0000	50.3433	88.4046
B	50.3433	0.0000	15.8055
C	88.4046	15.8055	0.0000

Group Means

	Pooled	Means for Group		
Variable	Mean	A	B	C
Sucrose	15.800	27.200	9.600	10.600
Glucose	16.533	9.600	21.600	18.400
Fructose	52.067	44.600	49.000	62.600
Sorbitol	4.3067	3.9600	4.0600	4.9000

Group Standard Deviations

	Pooled	StDev for Group		
Variable	StDev	A	B	C
Sucrose	3.992	5.070	2.966	3.647
Glucose	3.286	4.393	2.702	2.408
Fructose	5.342	4.278	1.414	8.081
Sorbitol	1.441	1.936	1.270	0.930

Covariance matrices for each class, that can be used to generate the corresponding scatter matrices through multiplication by the number of degrees of freedom, are also reported in the Session window (note that some values are missing since they are identical to symmetric values with respect to the main diagonal of matrices):

Covariance matrix for Group A

	Sucrose	Glucose	Fructose	Sorbitol
Sucrose	25.7000			
Glucose	1.3500	19.3000		
Fructose	9.3500	-4.2000	18.3000	
Sorbitol	-0.4650	5.5800	-6.7950	3.7480

Covariance matrix for Group B

	Sucrose	Glucose	Fructose	Sorbitol
Sucrose	8.8000			
Glucose	-3.7000	7.3000		
Fructose	2.0000	-0.5000	2.0000	
Sorbitol	2.7800	0.0300	1.4000	1.6130

Covariance matrix for Group C

	Sucrose	Glucose	Fructose	Sorbitol
Sucrose	13.3000			
Glucose	2.2000	5.8000		
Fructose	26.0500	8.9500	65.3000	
Sorbitol	1.5750	-1.5000	1.9750	0.8650

An important result is also the Summary of Classified Observations, that indicates the True group (the one declared), the Predicted Group and the Cross Validated (X-val) Group for each sample in the training set:

The table enables an evaluation of the attribution of each sample to a specific class.

Summary of Classified Observations

Observation	True Group	Pred Group	X-val Group	Group	Squared Distance Pred	Squared Distance X-val	Probability Pred	Probability X-val
1	A	A	A	A	6.254	25.699	1.00	1.00
				B	41.578	38.545	0.00	0.00
				C	70.267	74.084	0.00	0.00
2	A	A	A	A	1.511	2.569	1.00	1.00
				B	42.411	40.045	0.00	0.00
				C	77.635	73.594	0.00	0.00
...								
14	C	C	C	A	125.689	241.252	0.00	0.00
				B	39.805	97.238	0.00	0.00
				C	6.167	24.702	1.00	1.00
15	C	C	C	A	79.784	74.397	0.00	0.00
				B	16.510	15.410	0.00	0.00
				C	2.268	4.252	1.00	1.00

Minitab 18 also provides the so-called Classification Functions (Linear Discriminant Functions, in the specific case), one for each of the classes. They are used for the classification of unknown samples.

In particular, the general form of a Classification Function in Minitab is:

$$C_j = c_{j1}X_1 + c_{j2}X_2 + \ldots + c_{jp}X_p + c_{j0}$$

Values obtained for coefficients c_{j1}, c_{j2},…, c_{j0} for each class in the specific case are reported in the table shown on the right.

Linear Discriminant Function for Groups

	A	B	C
Constant	-44.19	-74.24	-114.01
Sucrose	0.39	-1.66	-2.50
Glucose	0.42	1.21	0.54
Fructose	1.46	2.53	3.48
Sorbitol	2.19	3.59	5.48

The assignment of a new sample to one of the classes is achieved by introducing the corresponding values of variables into classification (linear discriminant) functions:

Group A: $-44.19 + 0.39 \times 11 + 0.42 \times 23 + 1.46 \times 50 + 2.19 \times 3.9 = 51.301$

Group B: $-74.24 - 1.66 \times 11 + 1.21 \times 23 + 2.53 \times 50 + 3.59 \times 3.9 = 75.831$

Group C: $-114.01 - 2.5 \times 11 + 0.54 \times 23 + 3.48 \times 50 + 5.48 \times 3.9 = 66.282$

In the specific case, the maximum value (score) is obtained from the function referred to Group B, thus the new sample is classified in this group.

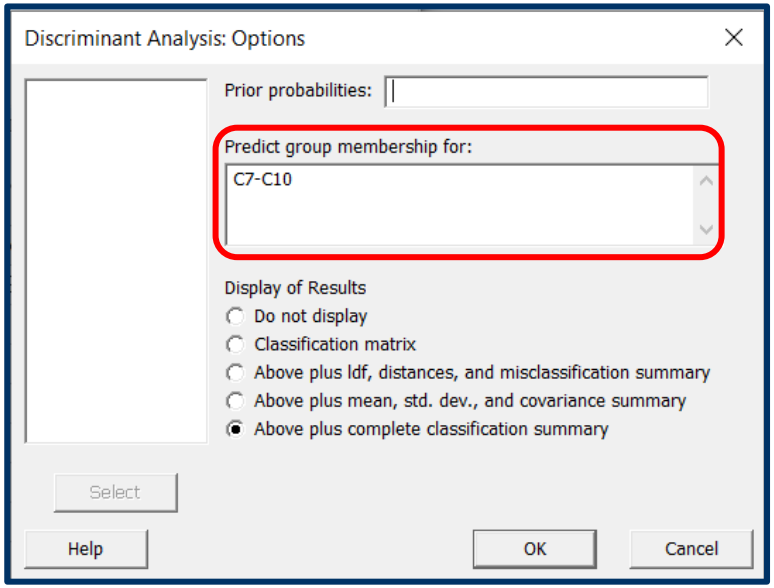
The assignment of one or more new samples to classes can be made automatically using the Minitab 18 software.

First, values of variables referred to new samples are introduced in appropriate columns, different from those including values of variables for samples in the training set:

Worksheet 1 ***

↓	C1-T	C2	C3	C4	C5	C6	C7	C8	C9	C10
	Classes	Sucrose	Glucose	Fructose	Sorbitol		Sucr Unk	Gluc Unk	Fruc Unk	Sorb Unk
1	A	20	6	40	4.3		11	23	50	3.9
2	A	27	11	49	2.9					
3	A	26	10	47	2.5					
4	A	34	5	47	2.9					
5	A	29	16	40	7.2					
6	B	6	26	49	3.8					
7	B	10	22	47	3.5					
8	B	14	21	51	6.3					

Those columns have to be indicated in the «Predict group membership for:» box included in the Discriminant Analysis: Options window:



Once the calculations are performed by the program, a specific table (Prediction for Test Observations) will appear at the end of the Session Window:

Prediction for Test Observations

Observation	Pred Group	From Group	Squared Distance	Probability
1	B			
		A	48.953	0.000
		B	0.469	1.000
		C	19.181	0.000

The new sample (Observation) will be labelled with the number of the worksheet row in which the corresponding values for variables are reported (row #1 in the specific example).

The Predicted Group is B, in accordance with calculations shown before, based on Linear Discriminant Functions.

The probability of this assignment, reported in the last column of the table, is related to the squared distance of the new sample from the centroids of the three groups.

In the specific example, the differences between distances are so large that the probability of assignment to Group B is 1 (100%). In other cases, the assignment is made to the class (group) for which the probability is higher.

Other statistical programs provide further interesting information on LDA in their output. As an example, Statgraphics also provides eigenvalues related to linear discriminant functions

In the case of Statgraphics, values of variables referred to eventual new samples to be classified are introduced in the same worksheet columns used for samples in the training set but the box referred to the classification is obviously left blank:

apple juice.sf6

	sucrose	glucose	fructose	sorbitol	variety
	ppm	ppm	ppm	ppm	
1	20	6	40	4,3	A
2	27	11	49	2,9	A
3	26	10	47	2,5	A
4	34	5	47	2,9	A
5	29	16	40	7,2	A
6	6	26	49	3,8	B
7	10	22	47	3,5	B
8	14	21	51	6,3	B
9	10	20	49	3,2	B
10	8	19	49	3,5	B
11	8	17	55	5,3	C
12	7	21	59	3,3	C
13	15	20	68	4,9	C
14	14	19	74	5,6	C
15	9	15	57	5,4	C
16	11	23	50	3,9	

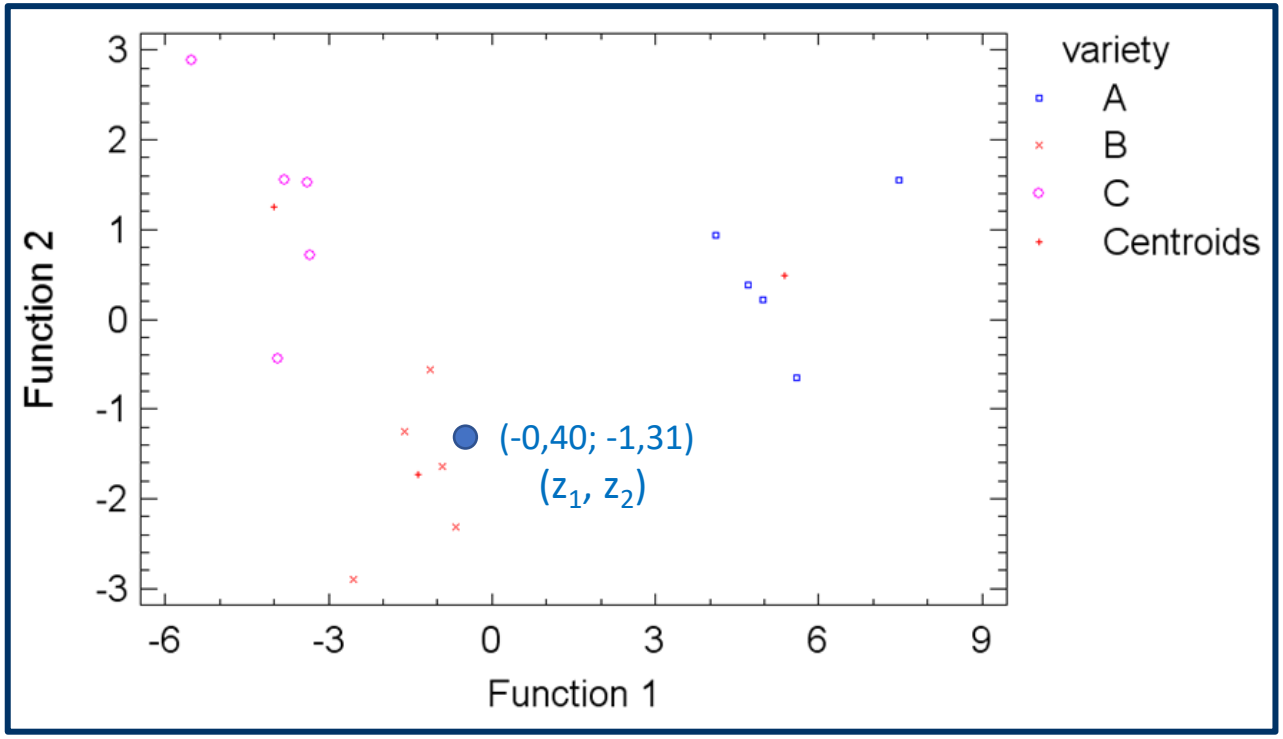
Once calculations are completed, a classification table is reported in the program's output:

Row	Label	Actual Group	Highest Group	Highest Value	Squared Distance	Prob.	2nd Highest Group	2nd Highest Value	Squared Distance	Prob.
1		A	A	32,8158	1,80646	1,0000	B	15,1536	37,1307	0,0000
2		A	A	47,7127	0,453413	1,0000	B	27,2629	41,3529	0,0000
3		A	A	43,1071	0,231869	1,0000	B	21,2304	43,9854	0,0000
4		A	A	44,9776	5,51544	1,0000	B	3,33119	88,8082	0,0000
5		A	A	46,8687	1,35416	1,0000	B	22,6454	49,8007	0,0000
6		B	B	83,5054	2,73056	0,9998	C	75,2009	19,3396	0,0002
7		B	B	65,8984	0,82146	1,0000	C	54,4361	23,746	0,0000
8		B	B	78,1962	1,42734	0,9937	C	73,1415	11,5367	0,0063
9		B	B	67,4658	0,215585	0,9998	C	58,6673	17,8126	0,0002
10		B	B	70,6632	0,276813	0,9973	C	64,7694	12,0643	0,0027
11		C	C	94,4142	0,737153	0,9894	B	89,8755	9,81468	0,0106
12		C	C	102,039	2,82863	0,9396	B	99,2951	8,31619	0,0604
13		C	C	121,563	0,131827	0,9997	B	113,272	16,7144	0,0003
14		C	C	148,225	5,03286	1,0000	B	131,406	38,6706	0,0000
15		C	C	98,3358	0,43643	0,9992	B	91,2148	14,6785	0,0008
16			B	74,4584	0,133832	0,9999	C	65,1024	18,8459	0,0001

As evidenced in row #16, the unknown sample is classified primarily in Group B (probability = 0.9999), due to the value assumed by the corresponding linear discriminant function (Highest value).
The program also reports the second possible assignment (2nd Highest Group) and the corresponding probability (0.0001).

Notably, slight (not significant) differences can be observed between Minitab 18 and Statgraphics in terms of values referred to classification functions and/or squared distances. They are due to small differences in rounding off during calculations.

A plot of discriminant functions is also generated by Statgraphics:



Centroids for different classes are also reported in the plot. Notably, (z_1, z_2) co-ordinates of the unknown sample can be calculated by introducing standardized values of their variables into linear discriminant functions:

$$Z_1 = -1.090 \times (-0.221) - 0.1092 \times 1.23 - 0.4645 \times (-0.2897) + 1.227 \times (-0.525) = -0.40$$

$$Z_2 = 0.7414 \times (-0.221) - 0.8295 \times 1.23 + 0.5028 \times (-0.2897) - 0.037 \times (-0.525) = -1.31$$

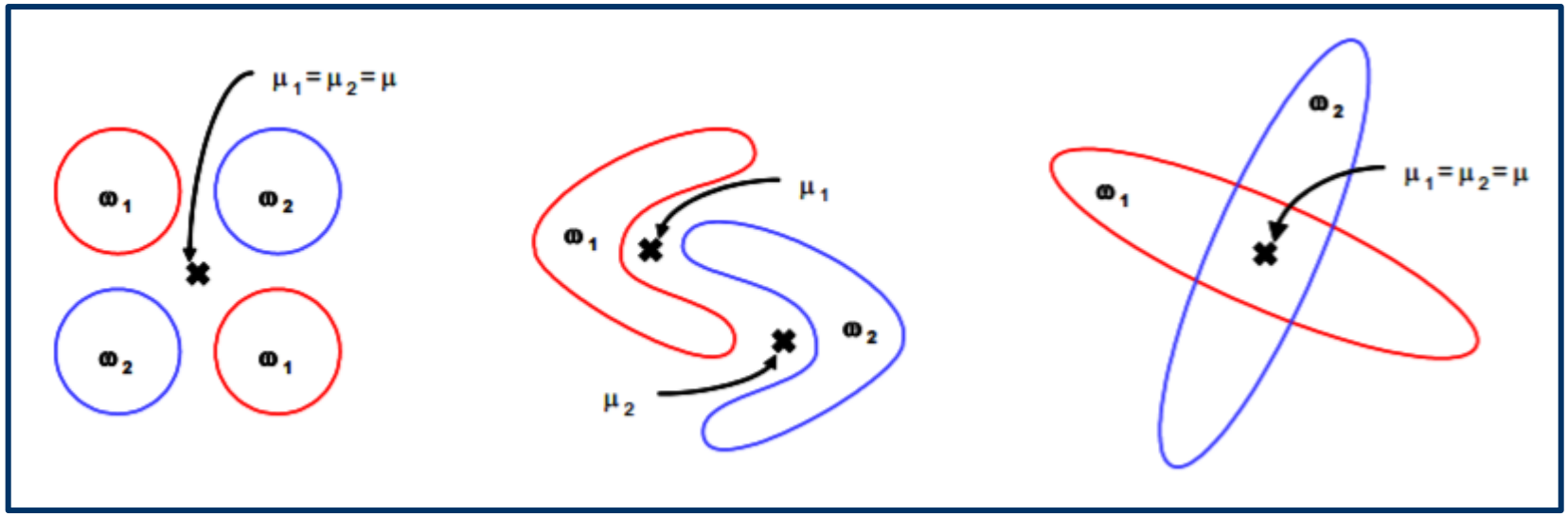
Based on these co-ordinates, the point referred to the unknown sample clearly appears closer to points related to Class B, thus confirming the previous assignment.

Limitations of Linear Discriminant Analysis (LDA)

Three major limitations can be described for LDA:

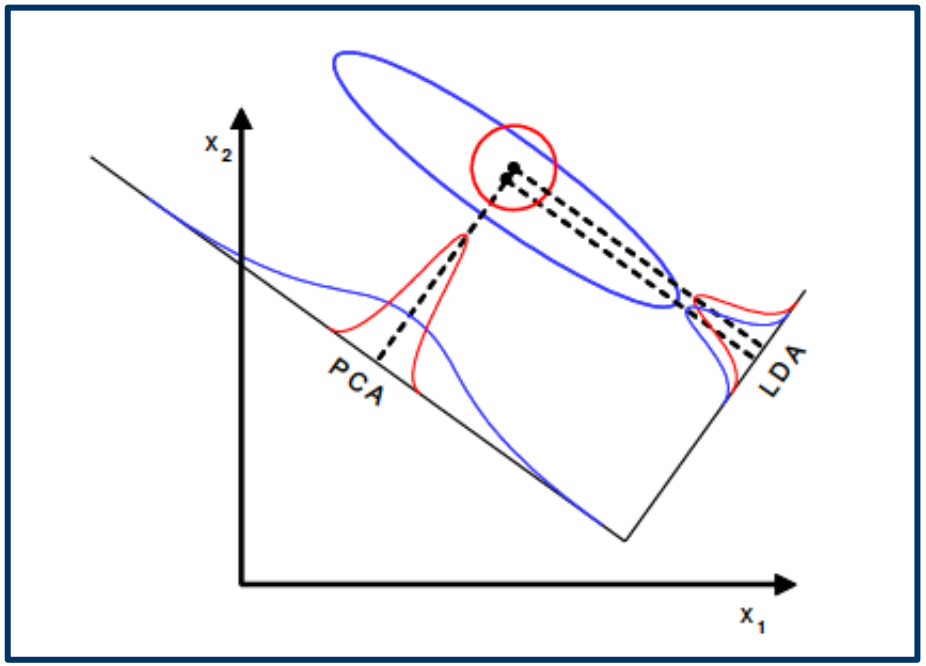
1) LDA produces at most k-1 feature projections, when k classes are considered. If classification error estimates establish that more features are needed, other methods must be employed to provide additional features;

2) LDA is a parametric method. It data distributions are significantly non-Gaussian, LDA projections may not preserve the complex structure included in data needed for classification:



3) LDA will also fail if discriminatory information is embedded in the variance of data, rather than in the mean.

In this case PCA is expected to be more effective than LDA.

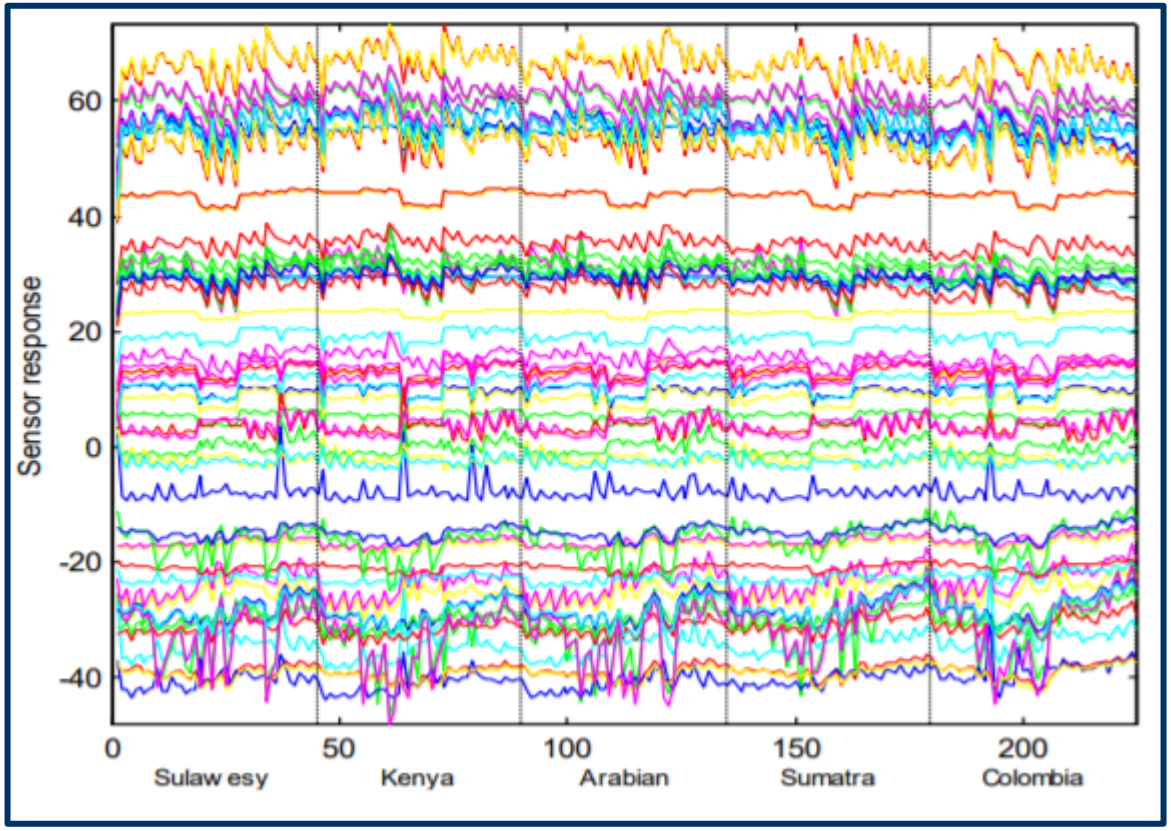


A comparison between LDA and PCA

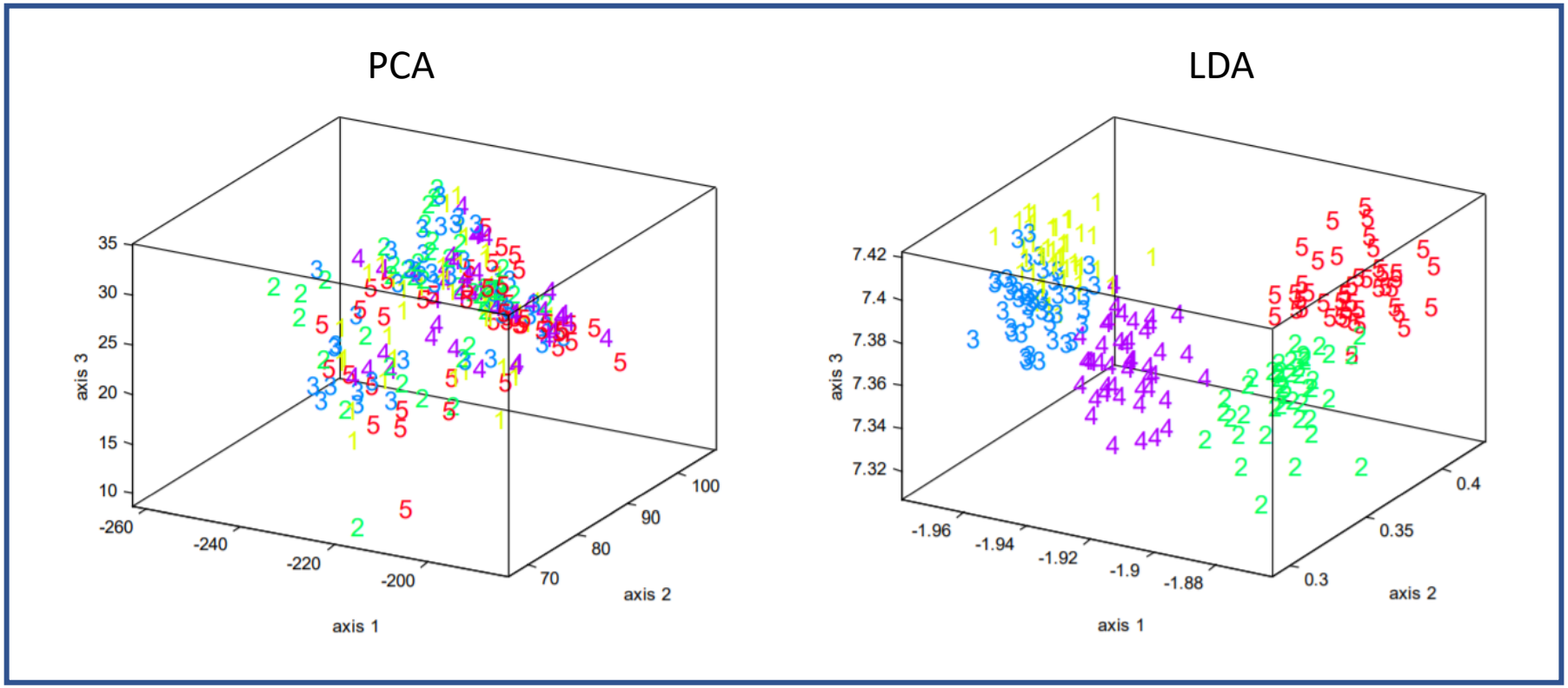
As an example of comparison between LDA and PCA, the recognition of coffee odour will be considered.

Odour released from five varieties of coffee beans, Sulawesy, Kenya, Arabian, Sumatra and Colombia, was analyzed using an array consisting of 60 gas sensors. 45 analyses were performed for each variety.

In the figure shown on the right readings obtained for all samples by each of the 60 sensors are represented as continuous lines with different colors.



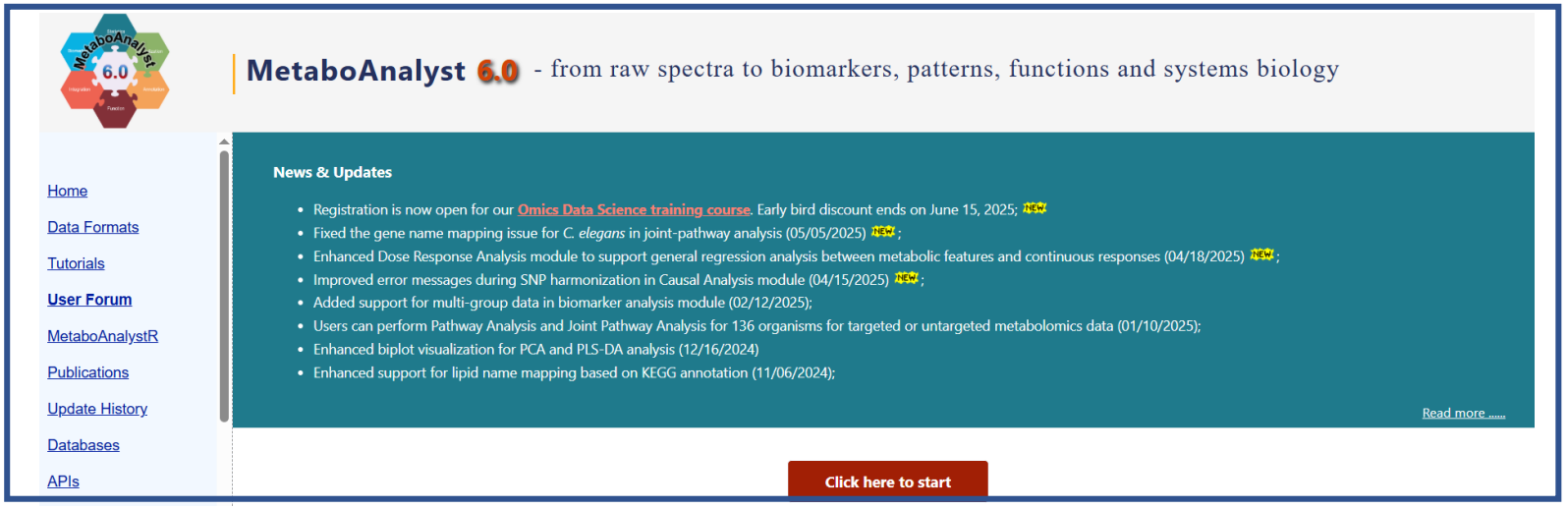
The comparison between 3D-scatter plots obtained using PCA and LDA clearly shows that LDA was more effective in separating the five coffee beans varieties:



This is an example of a system in which discriminatory information is not aligned with the direction of maximum variance.

Chemometrics based on the MetaboAnalyst web-based platform

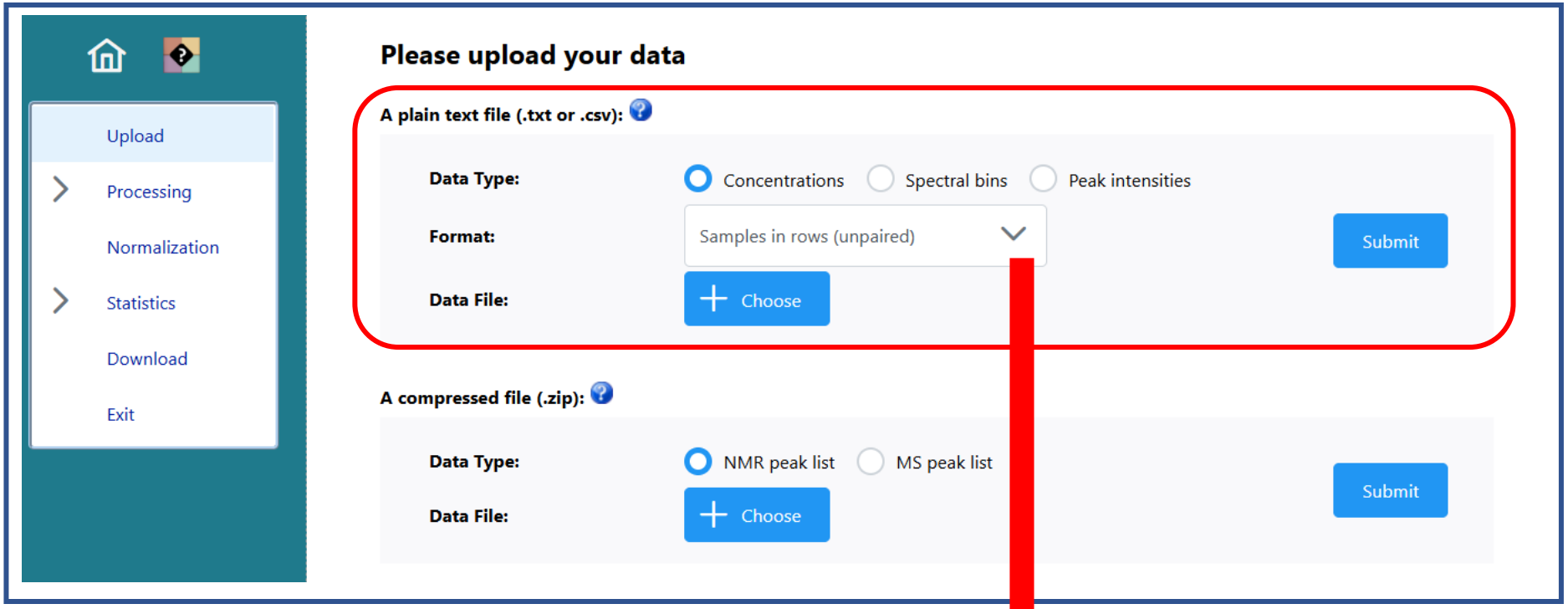
The MetaboAnalyst platform is freely accessible at the web address metaboanalyst.ca



Module Overview

Input Data Type	Available Modules (click on a module to proceed, or scroll down to explore a total of 18 modules including utilities)				
LC-MS Spectra (mzML, mzXML or mzData)			Spectra Processing [LC-MS w/wo MS2]		
MS Peaks (peak list or intensity table)		Peak Annotation [MS2-DDA/DIA]	Functional Analysis [LC-MS]	Functional Meta-analysis [LC-MS]	
Generic Format (.csv or .txt table files)	Statistical Analysis [one factor]	Statistical Analysis [metadata table]	Biomarker Analysis	Statistical Meta-analysis	Dose Response Analysis
Annotated Features (metabolite list or table)		Enrichment Analysis	Pathway Analysis	Network Analysis	
Link to Genomics & Phenotypes (metabolite list)			Causal Analysis [Mendelian randomization]		

If data have been already processed and values of variables are thus known, a txt or an Excel csv (comma-separated values) file, with samples reported in rows, are the simplest files that can be used as input, selecting «Concentrations» as the Data Type:



A	B	C	D	E	F	G	H	I
Sample Number	Sample type	**281.2478**	**255.2323**	**279.2322**	**295.227**	**253.217**	**277.217**	**313.238**
1	wild-type	14.23357	25.22179	2.00511	2.737188	5.102289	0.944709	1.821321
2	wild-type	17.44554	23.61866	1.819832	3.036436	5.527837	0.93008	2.066472
3	wild-type	17.47118	27.63773	1.777519	3.107297	5.625143	0.996083	2.006392
4	wild-type	18.84249	23.5063	1.886507	3.394122	5.061856	0.920076	2.135716
5	wild-type	15.00194	23.22418	2.076255	2.95843	4.976105	1.118026	1.903074
6	wild-type	12.7512	30.28206	1.908383	2.273857	6.334425	1.118904	1.423481
7	wild-type	13.65598	20.95872	1.858636	2.355915	4.443018	1.239853	1.296301
8	wild-type	16.61169	23.34575	1.852342	2.608314	5.365115	1.152143	1.744407
9	wild-type	15.32911	19.0413	2.124511	2.304715	3.764959	1.348352	1.654135
10	wild-type	15.07204	25.55745	2.192229	2.606511	4.547116	0.997745	1.635876
11	wild-type	16.46733	21.21967	2.656335	2.599264	3.92061	1.555587	1.7348
12	wild-type	17.89331	22.19917	2.347418	2.730731	4.798052	1.32717	2.421233
13	wild-type	12.43089	30.96677	1.724293	2.896892	4.941796	0.783904	1.861966
14	wild-type	19.33922	23.6542	1.926951	3.080187	6.094779	1.195816	2.22795

Part of the dataset adopted, including spectral intensities obtained for 30 fatty acid ions (*m/z* values are indicated in the columns) referred to 26 wild and 25 farmed Canadian salmon samples.

Summary of the data integrity check performed by Metaboanalyst:

Data Integrity Check:

- Checking sample names - spaces will replaced with underscore, and special characters will be removed;
- Checking the class labels - at least three replicates are required in each class.
- The data (except class labels) must not contain non-numeric values.
- If the samples are paired, the pair labels must conform to the specified format.
- The presence of missing values or features with constant values (i.e. all zeros).

Data processing information:

Checking data content ...passed.

Samples are in rows and features in columns

The uploaded file is in comma separated values (.csv) format.

The uploaded data file contains 51 (samples) by 30 (compounds) data matrix.

Samples are not paired.

2 groups were detected in samples.

Only English letters, numbers, underscore, hyphen and forward slash (/) are allowed.

Other special characters or punctuations (if any) will be stripped off.

All data values are numeric.

A total of 0 (0%) missing values were detected.

By default, missing values will be replaced by 1/5 of min positive values of their corresponding variables

Click the **Proceed** button if you accept the default practice;

Or click the **Missing Values** button to use other methods.

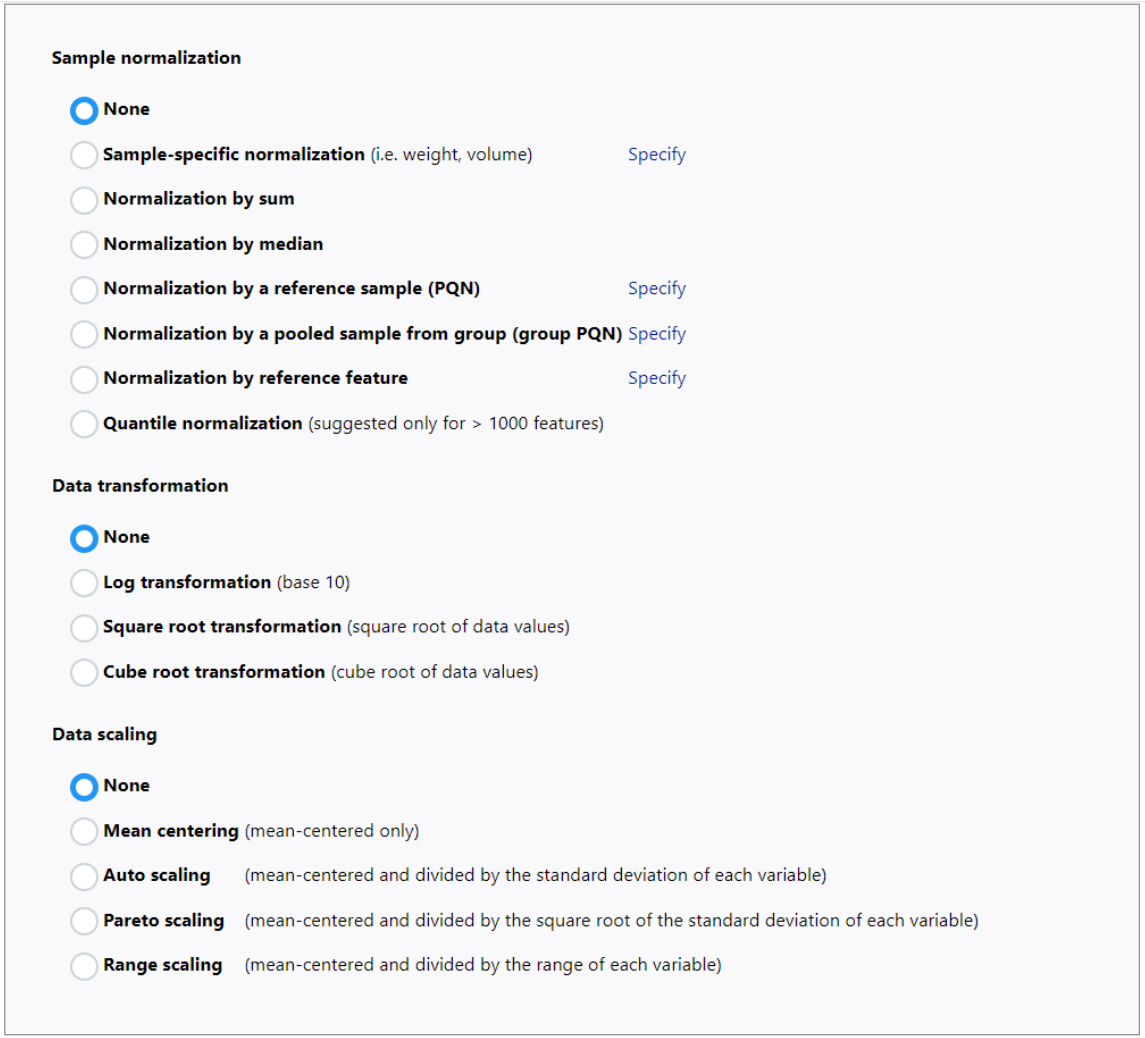
Edit Groups | Missing Values | Proceed

Overview of the available normalization/transformation/scaling approaches:

Normalization Overview:

The normalization procedures are grouped into three categories. You can use one or combine them to achieve better results.

- Sample normalization is for general-purpose adjustment for systematic differences among samples;
- Data transformation applies a mathematical transformation on individual values themselves. A simple mathematical approach is used to deal with negative values in log and square root (FAQs #14)
- Data scaling adjusts each variable/feature by a scaling factor computed based on the dispersion of the variable.



Sample normalization

- (●) **None**
- ( ) **Sample-specific normalization** (i.e. weight, volume) Specify
- ( ) **Normalization by sum**
- ( ) **Normalization by median**
- ( ) **Normalization by a reference sample (PQN)** Specify
- ( ) **Normalization by a pooled sample from group (group PQN)** Specify
- ( ) **Normalization by reference feature** Specify
- ( ) **Quantile normalization** (suggested only for > 1000 features)

Data transformation

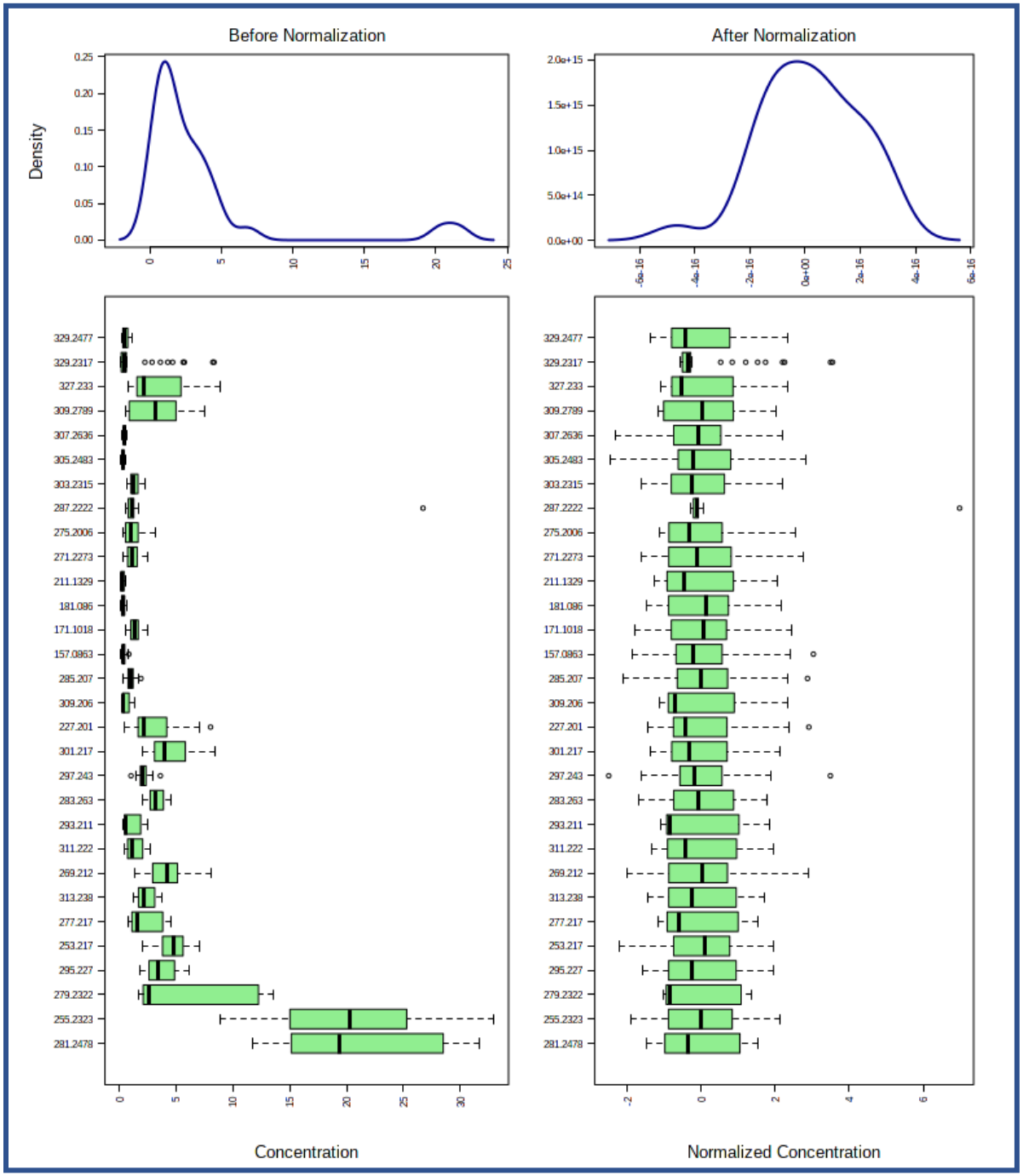
- (●) **None**
- ( ) **Log transformation** (base 10)
- ( ) **Square root transformation** (square root of data values)
- ( ) **Cube root transformation** (cube root of data values)

Data scaling

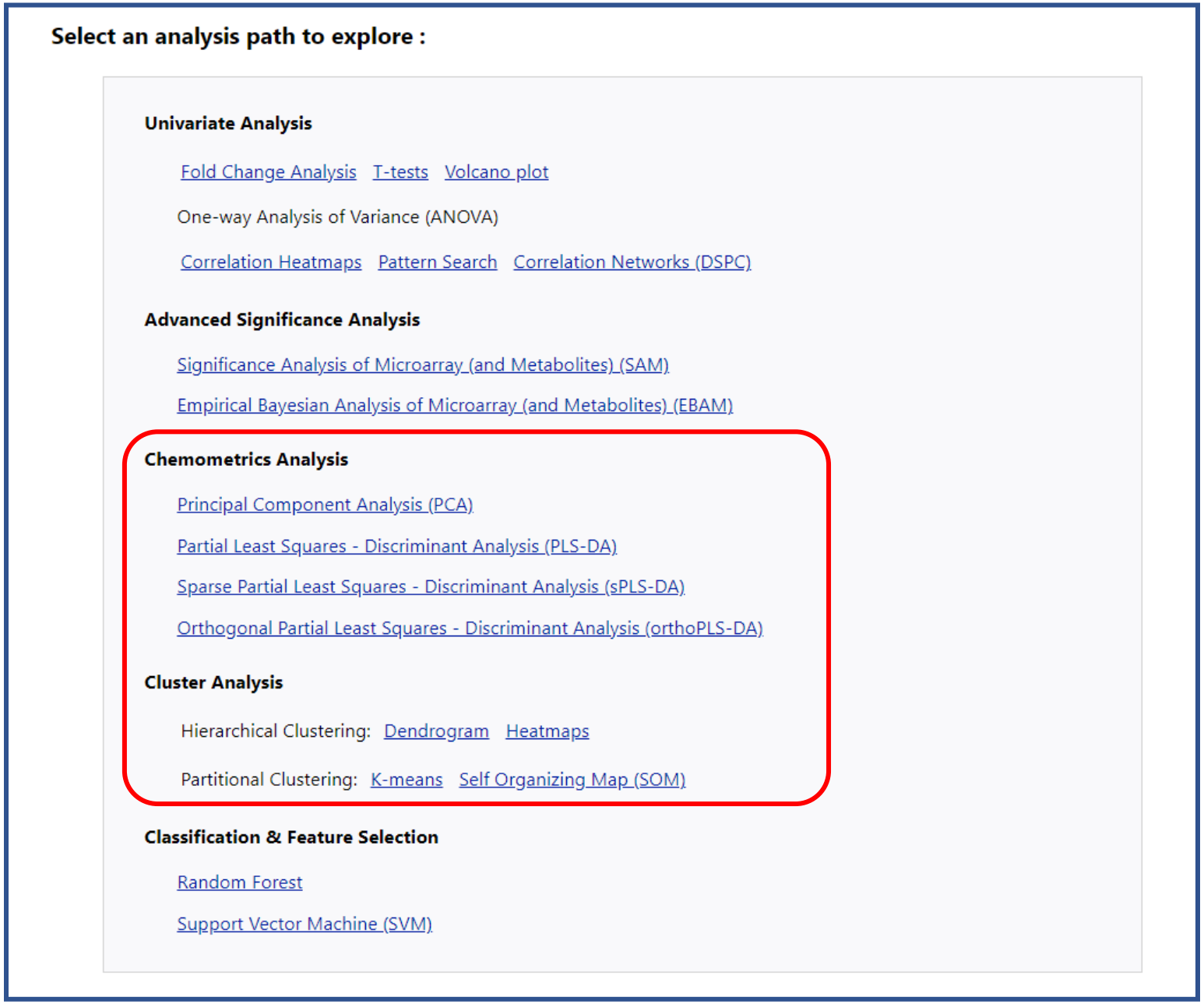
- (●) **None**
- ( ) **Mean centering** (mean-centered only)
- ( ) **Auto scaling** (mean-centered and divided by the standard deviation of each variable)
- ( ) **Pareto scaling** (mean-centered and divided by the square root of the standard deviation of each variable)
- ( ) **Range scaling** (mean-centered and divided by the range of each variable)

In this case data were subjected to autoscaling, since there were significant differences between variables values.

By clicking on the «View results» button, box and whisker plots are shown for each variable before and after normalization.



After data normalization, many different types of elaborations can be selected:

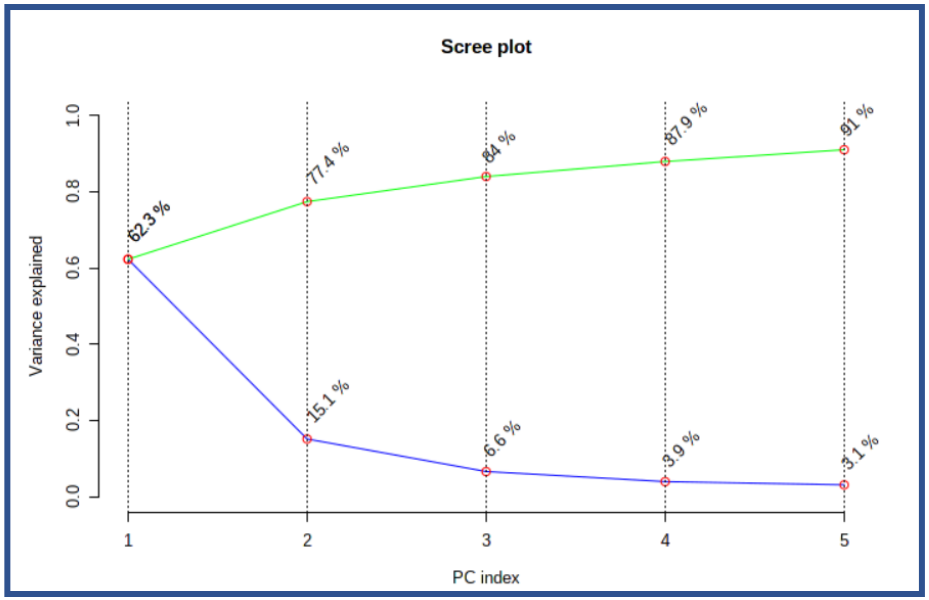


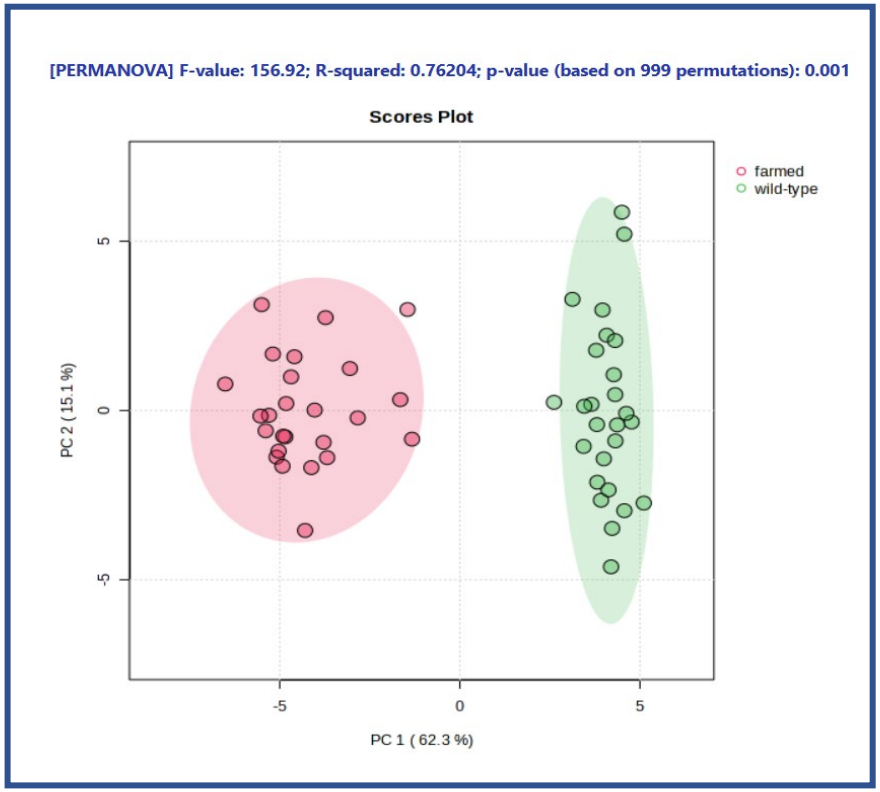
Once PCA is selected and performed, all possible combinations of bi-dimensional score plots can be visualized in the Overview.

In the following figure, all combinations involving the top 5 principal components are shown:

The Scree plot indicates absolute and cumulative contributions of principal components to the overall variance.

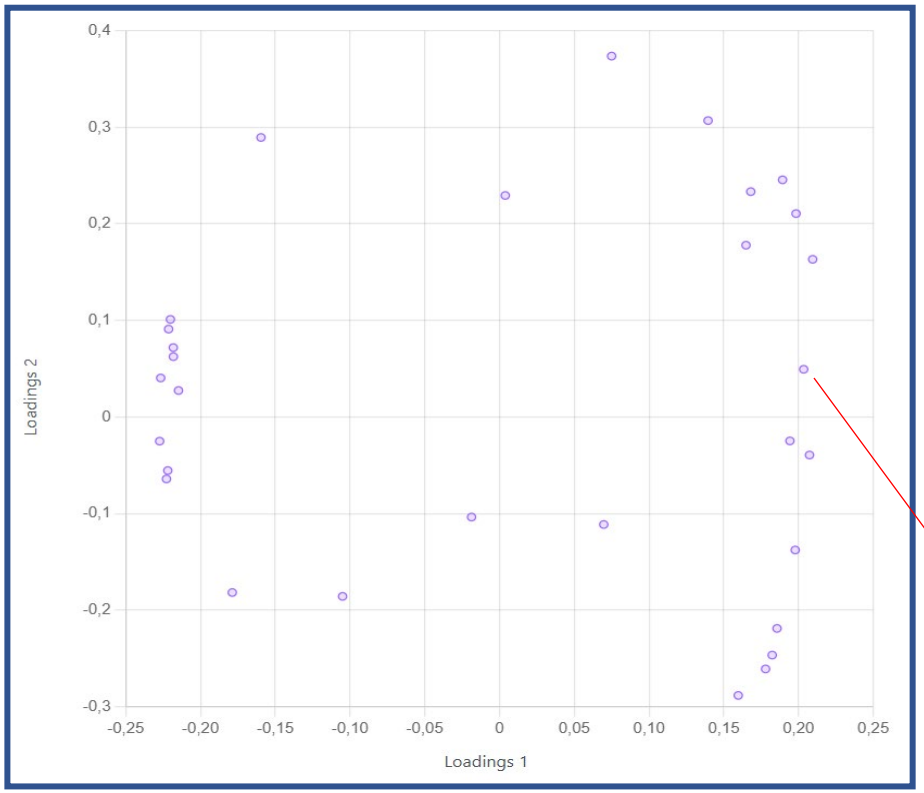
Here, values referred to the first five principal components are shown:



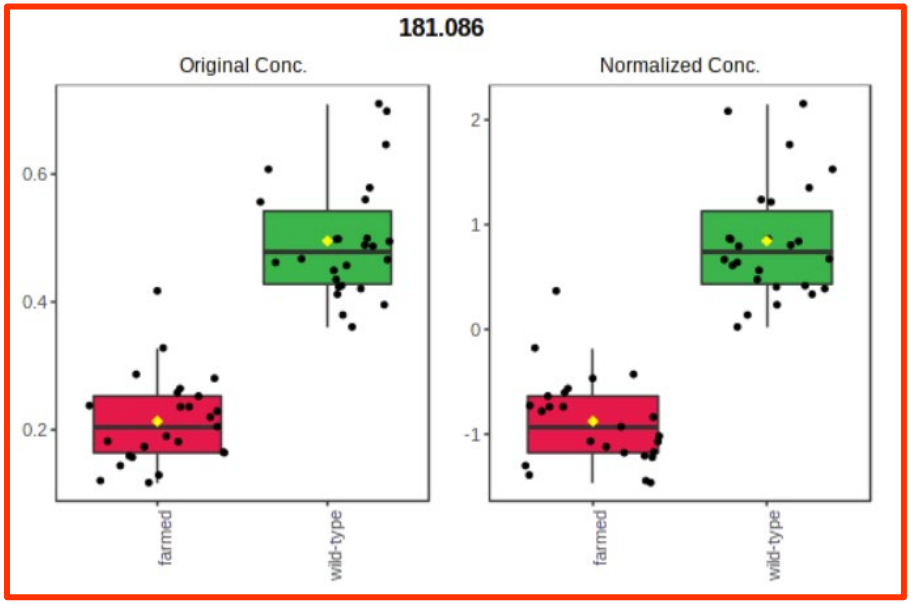


The Score plot obtained for the first two principal components is integrated by ellipses representing the 95% confidence areas related to the two sample groups.

The PERMANOVA calculation evaluates the significance of the separation between the two groups.

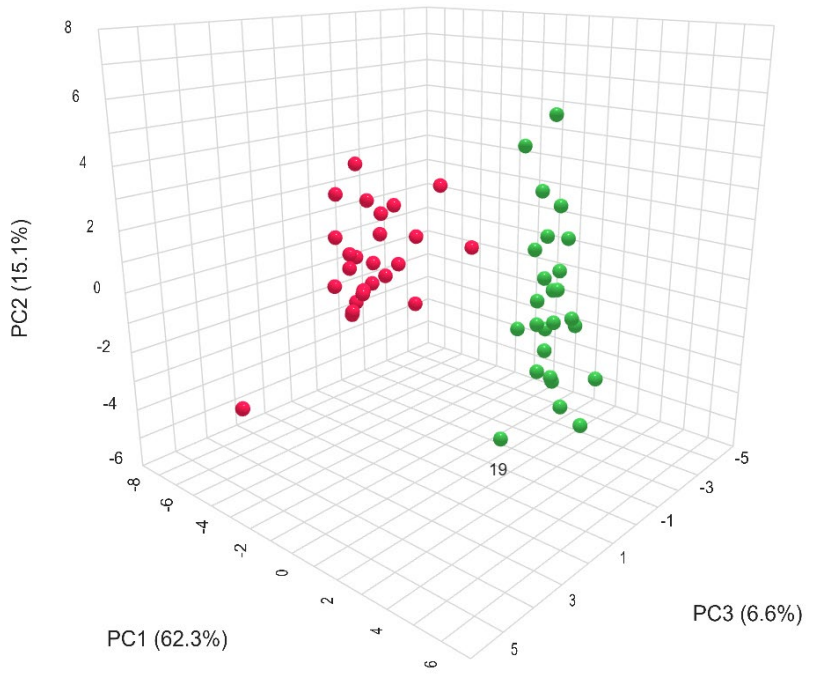


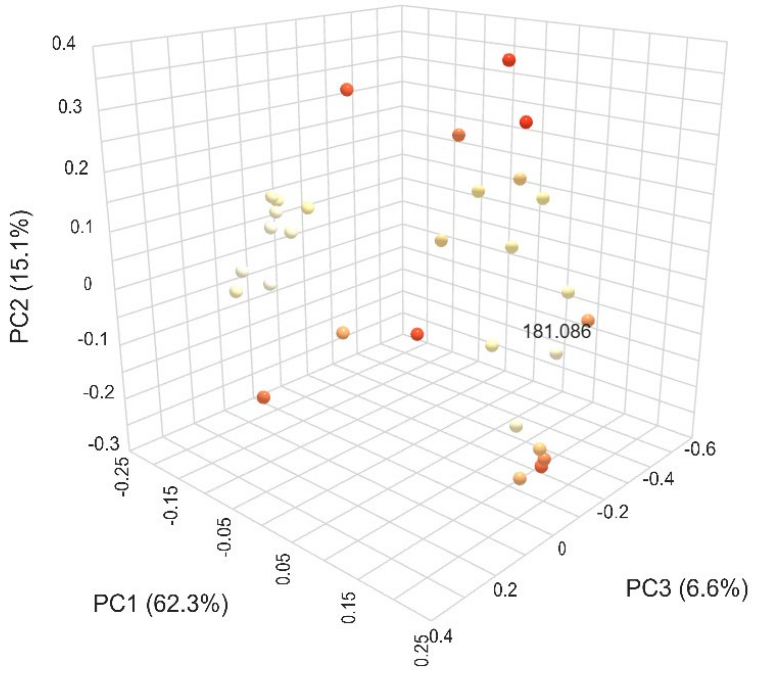
An interactive Loading plot can be visualized for each couple of principal components.



By clicking on the point referred to a specific variable, a new window is opened, reporting box-and-whisker plots for the original and the normalized (auto-scaled, in this case) values of the variable.

Synchronized 3D plots, corresponding to score and loading plots for three principal components, can be also visualized:

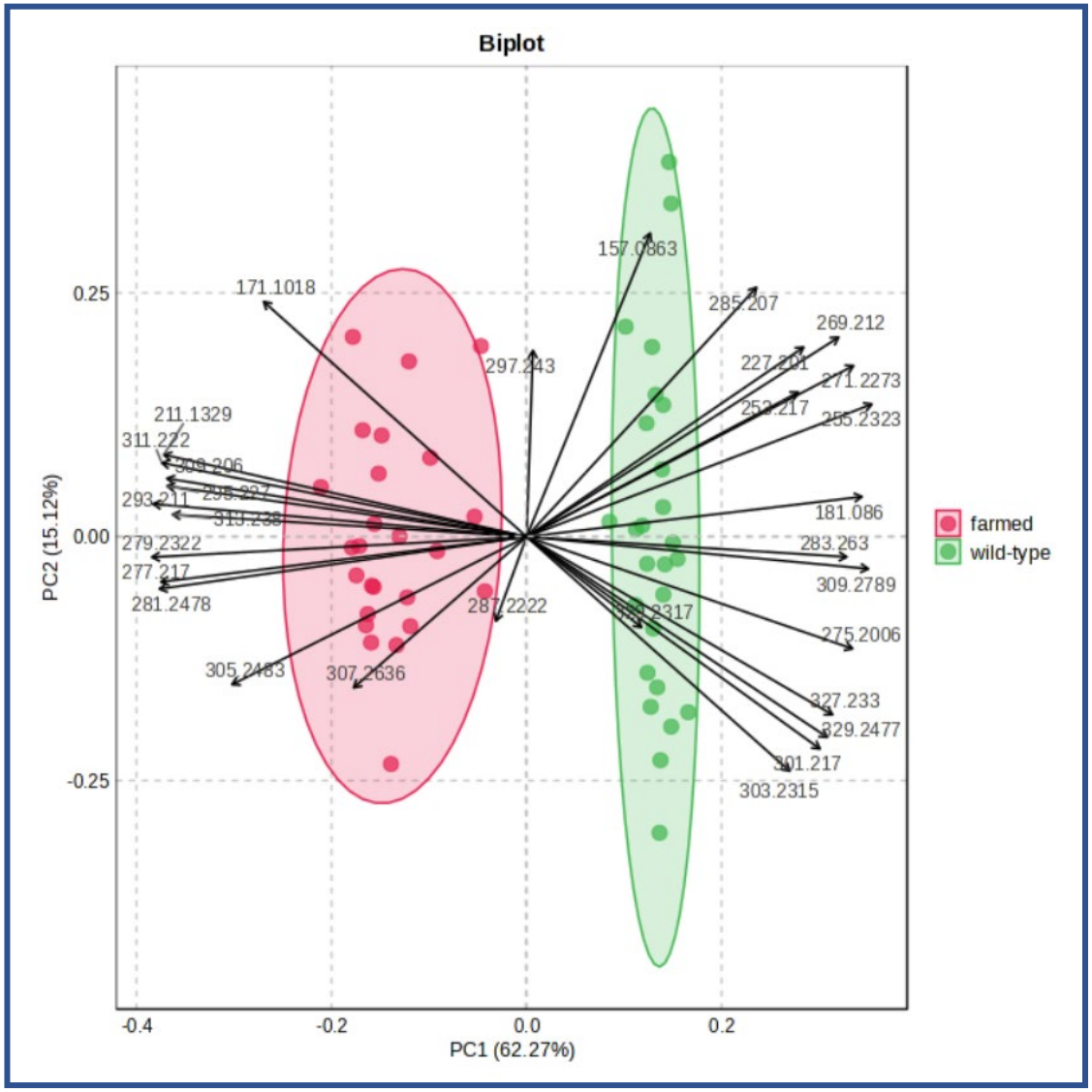




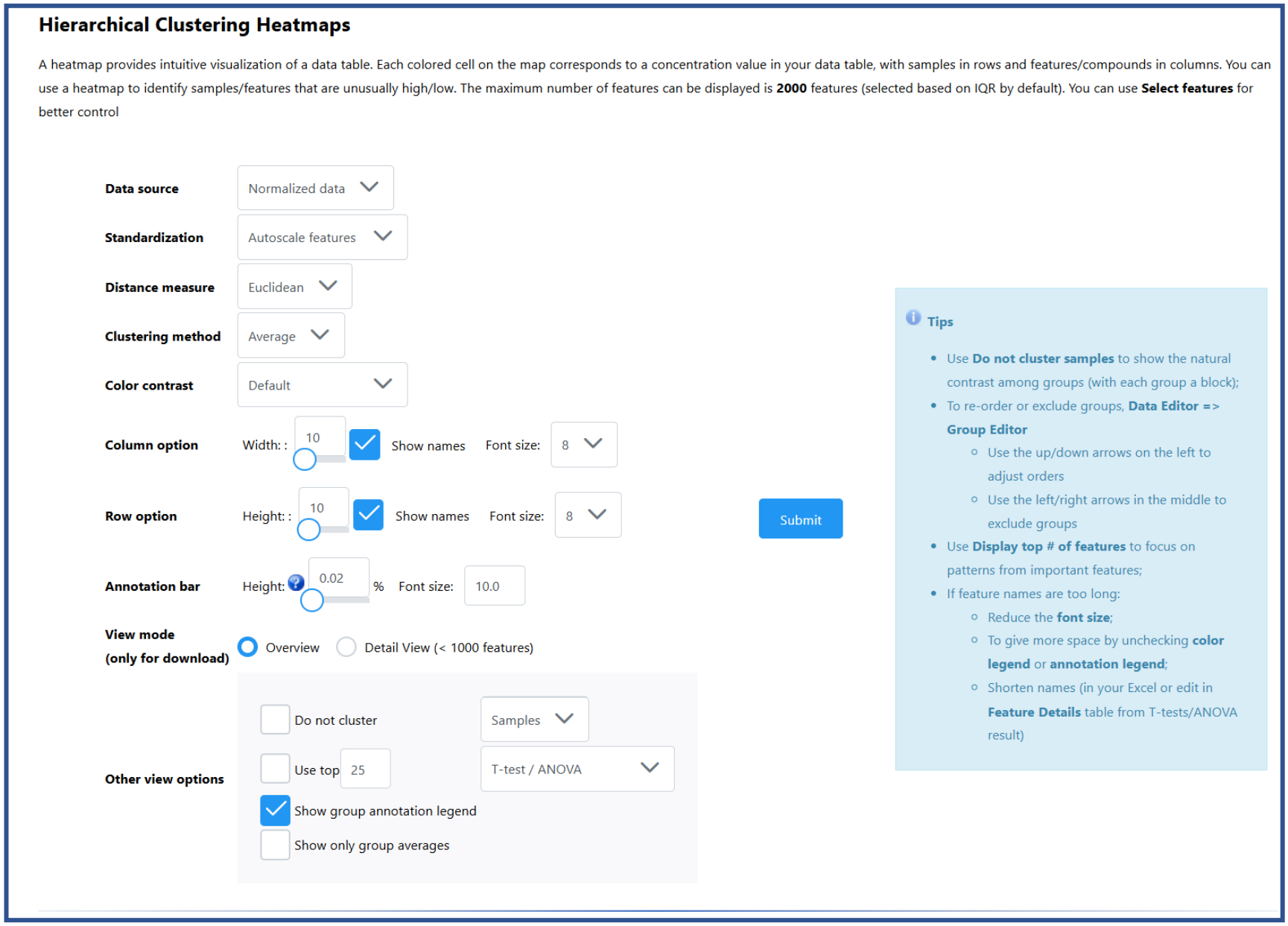
The two plots can be rotated contemporarily around different axes, to emphasize the relationship between principal components and variables.

Each plot is also interactive, thus the number of the specific sample or the name of the variable, according to the case, can be visualized by clicking on specific points.

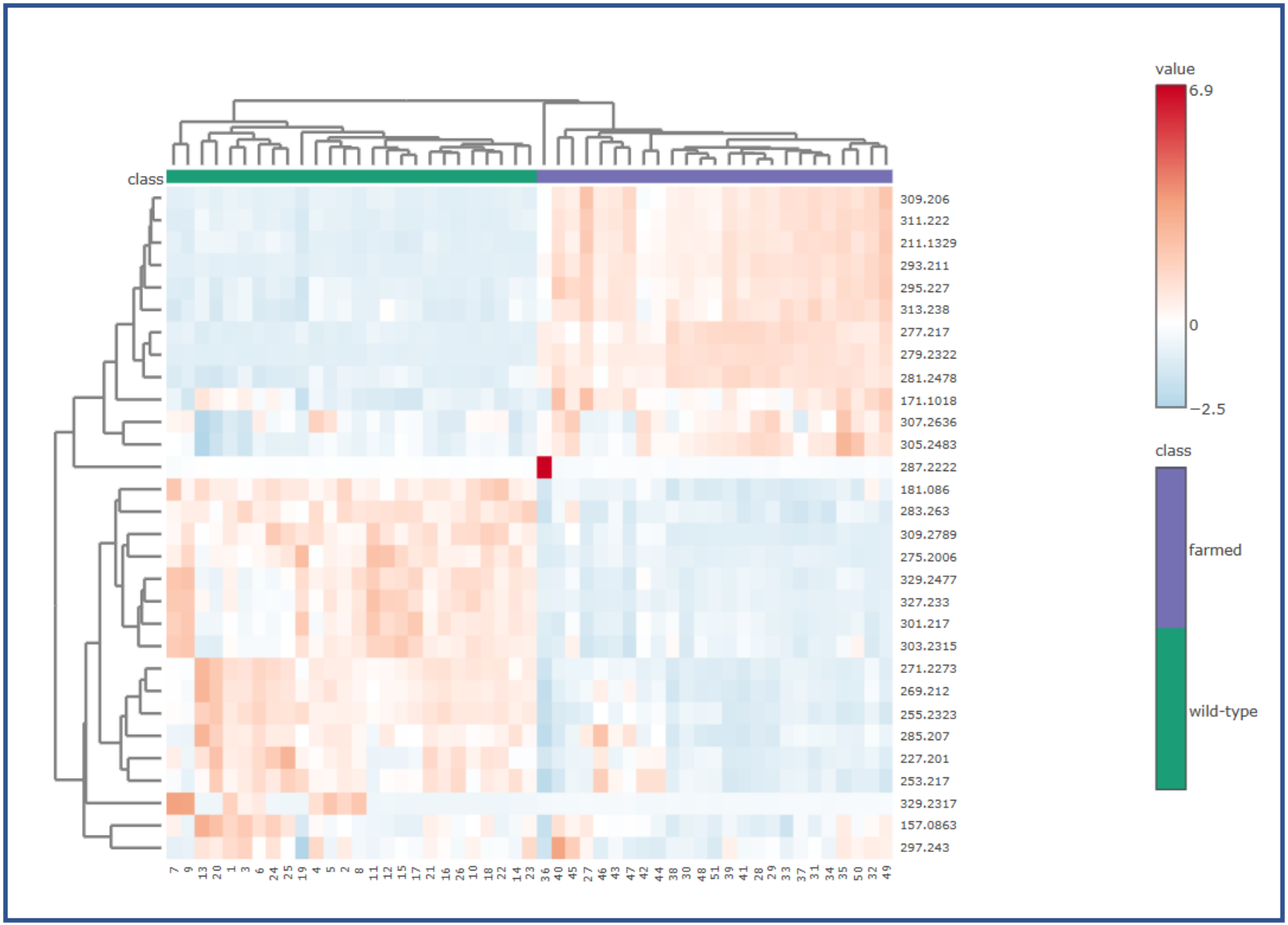
A biplot can be also visualized for each couple of principal components, with arrows referred to variables being reported with an appropriate scale and each sample represented by a colored dot. Colored areas corresponding to different groups of samples are also drawn.



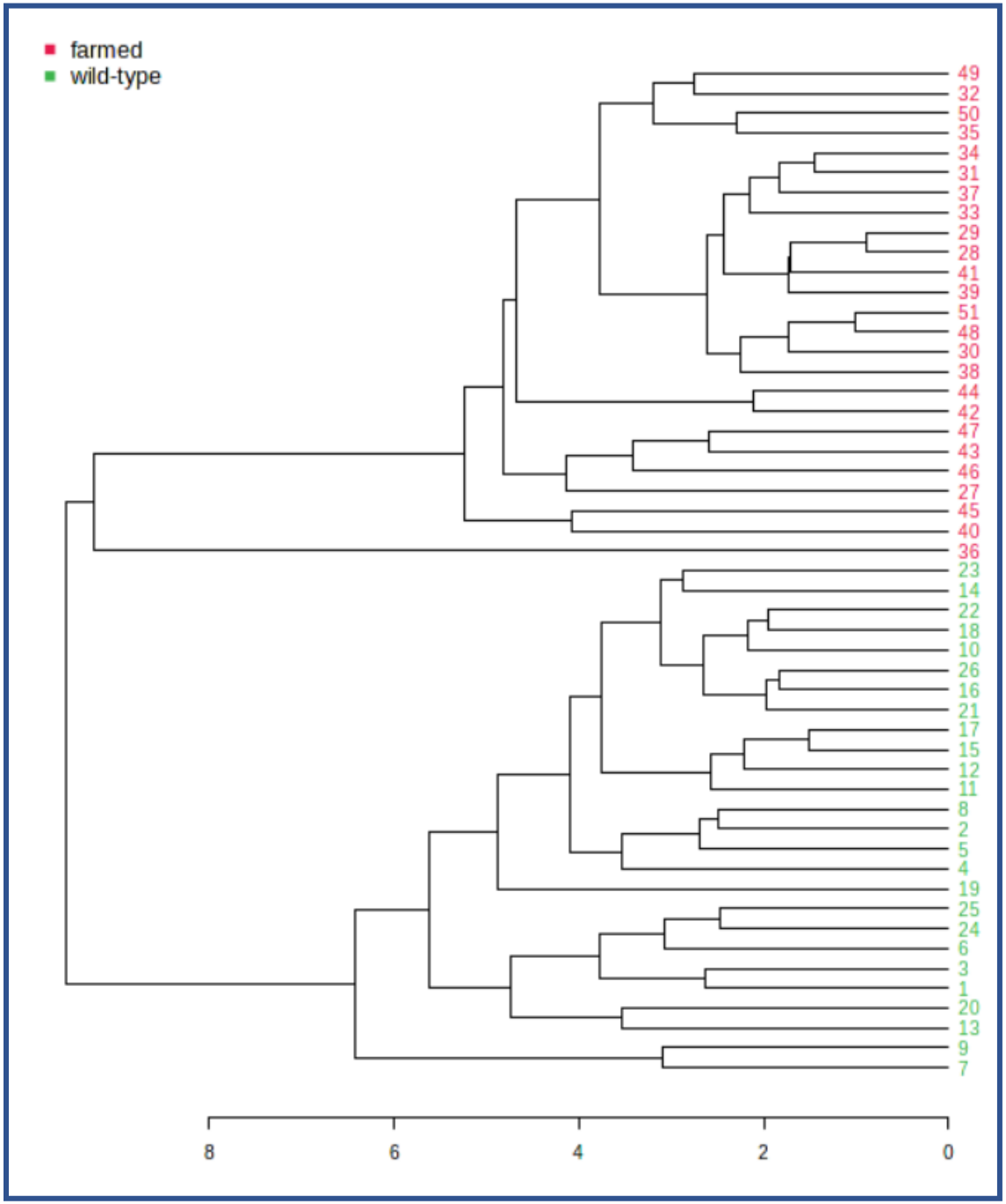
Hierarchical Clustering Heatmaps can be obtained using Metaboanalyst, along with conventional HCA dendrograms. In the following figure the settings adopted for the elaboration of salmon fatty acids data are shown.



The heatmap resulting from salmon fatty acid data emphasizes the good clustering of farmed and wild-type samples, and also the presence of a defined clustering of variables:

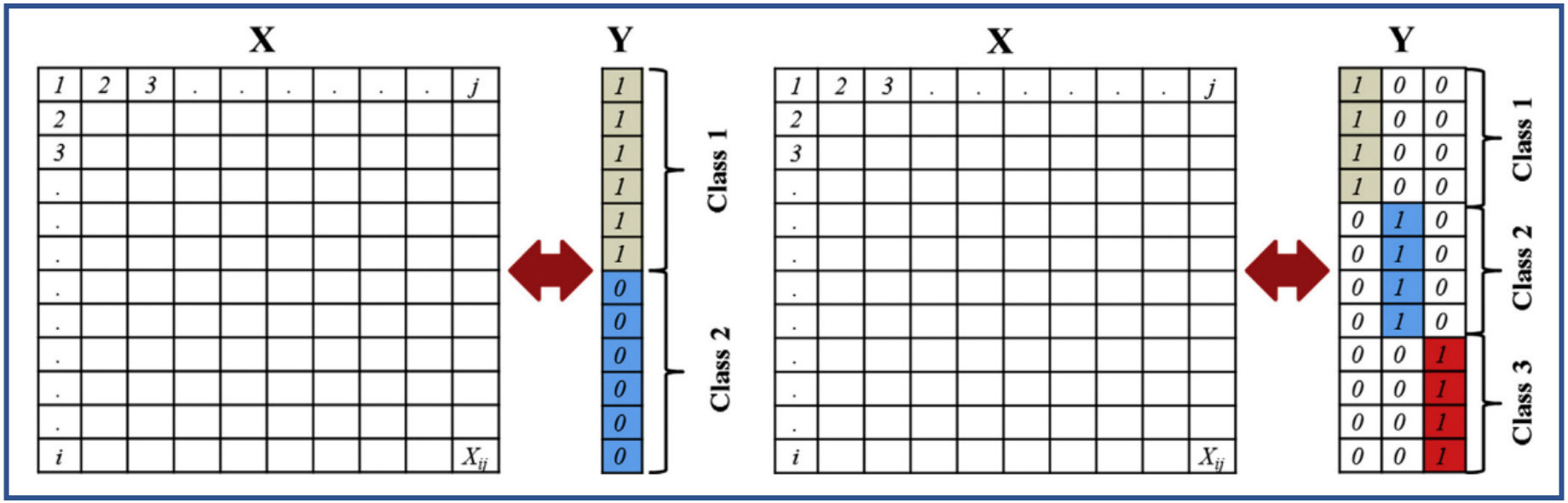


In this case the conventional dendrogram, performed only for samples, enables a better visualization of the relationships existing between specific samples:



The Metaboanalyst platform also includes some approaches to discriminant analysis based on Partial Least Squares. One of the most popular, although sometimes can lead to controversial results, is Partial Least Squares – Discriminant Analysis (PLS-DA).

In this case, the assignment of a sample to a specific class, which is a categorical variable, is preliminarily coded into a numerical variable (like 0 and 1), leading to a vector **y** if just two classes are considered (in this case the algorithm is known as PLS1-DA), and to a matrix **Y** if more than two classes are involved (algorithm PLS2-DA), as exemplified by the following figure:

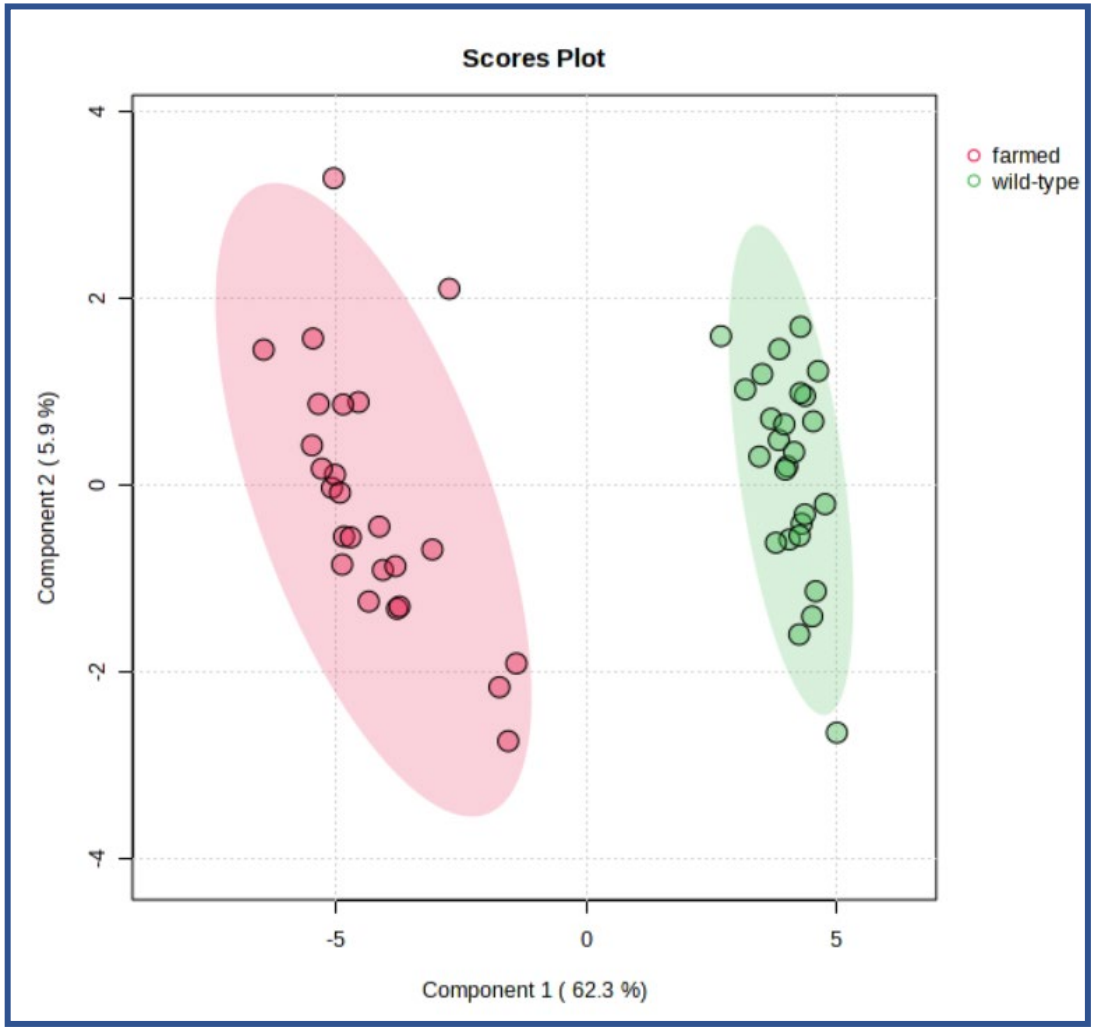


In this case **X** is the matrix containing the values of variables for samples included in the classes, with rows representing samples and columns representing variables, as usual.
Like in PLS regression, the algorithm searches for components arising from the original variables accounting for a relevant portion of the covariance between **X** and **y** (or **Y**) and then finds a regression model that can be used to predict the assignment of a new sample to one of the classes.

The overview obtained for PLS-DA results using Metaboanalyst includes several options and shows plots for couples of components that resemble PCA score plots, although their meaning is different:

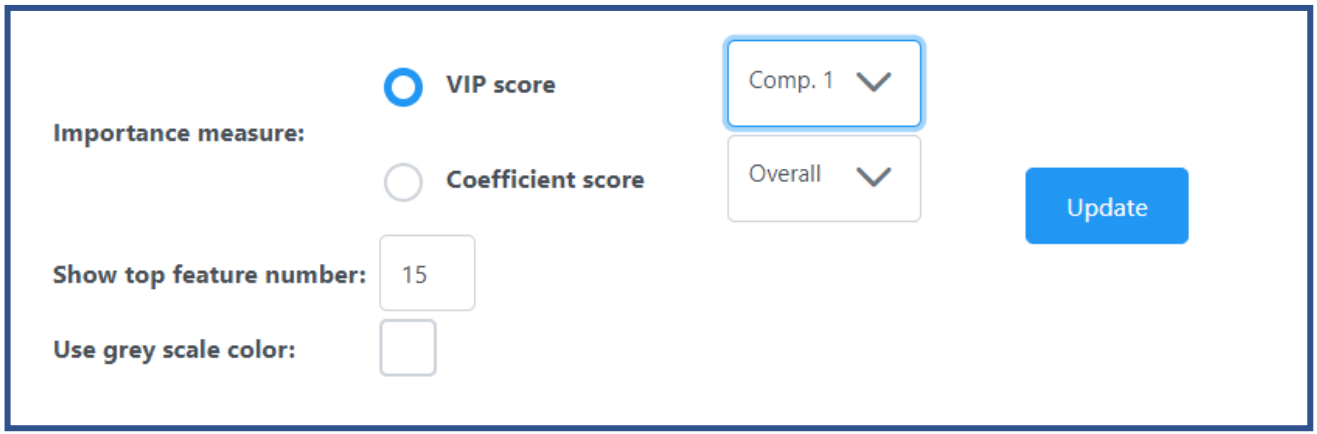
In the case of PLS-DA, components are combination of original variables ordered according to their ability to account for the covariance between matrix **X** and vector **y**/matrix **Y**.

However, the percentages shown in the score plot for each component still correspond to the proportion of matrix **X** variance accounted for by that component. Consequently, it is not impossible that component 2 accounts for a higher variance with respect to component 1.



Loading plots and synchronized 3D plots are reported by Metaboanalyst also for PLS-DA. In this case loadings represent the contributions of variables to a specific component.

Further PLS-DA outputs are included in the Imp. Features link of the Overview menu:



In particular, Variable Importance in Projection (VIP) scores are calculated for each variable j, according to the following formula:

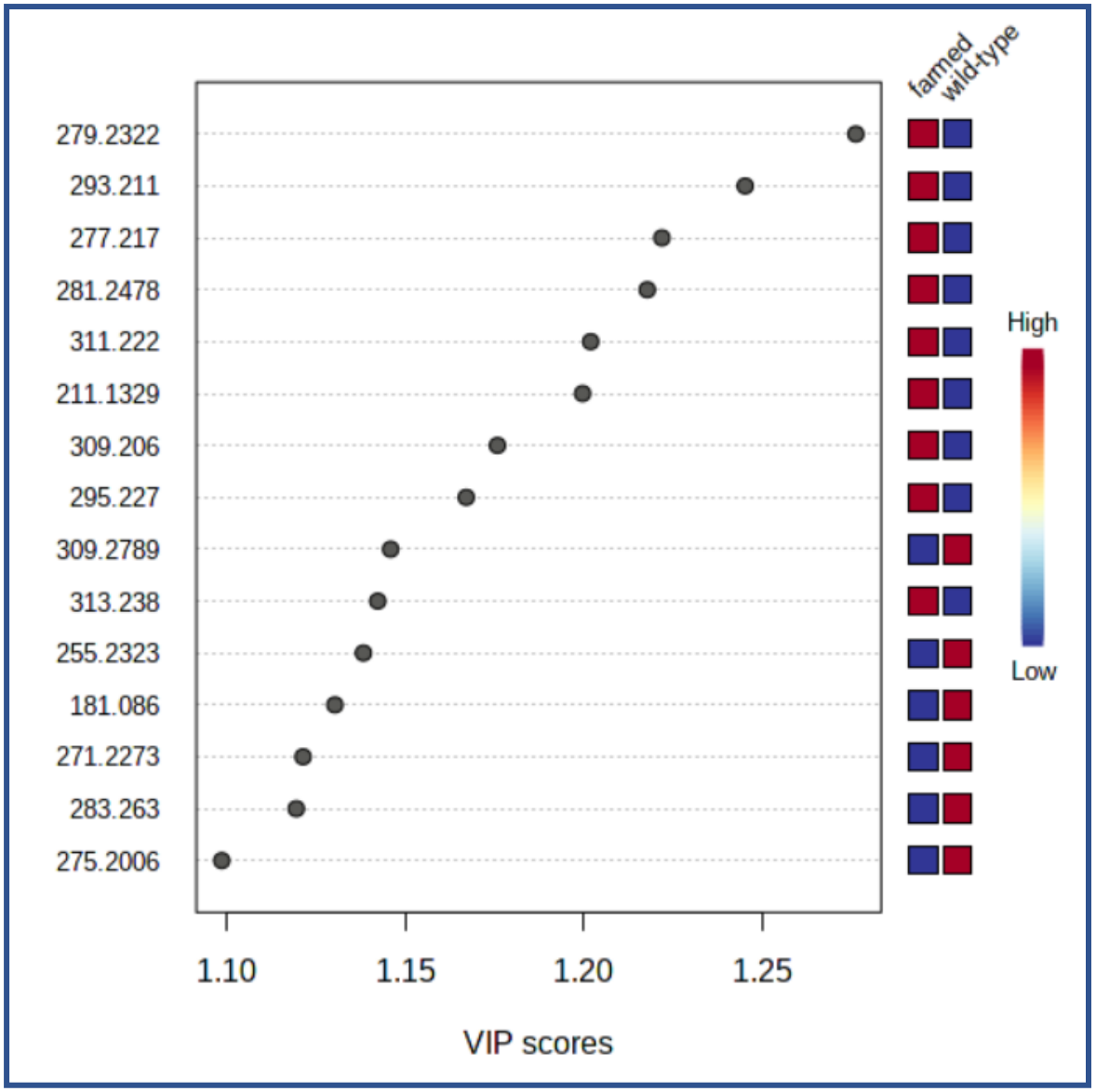
$$\mathrm{VIP}_j^2 = \sum_f w_{jf}^2 \, \mathrm{SSY}_f \; J/(\mathrm{SSY}_{tot.expl.} \; \mathrm{F})$$

where J and F represent the total numbers of original and latent variables, respectively, w^2_{jf} represents the weight of variable j on the latent variable f and SSY_f and $SSY_{tot.expl.}$ represent, respectively, the portion of **y** or **Y** variance explained by latent variable f and the variance explained by all latent variables.

It is common to assume as a threshold a VIP value larger than 1 (i.e., larger than the average of squared VIP values), which means that a selected variable will have an above average influence on the model explaining response Y.

Alternative threshold values include lowering the threshold to 2/3 or considering the average of VIP values.

In any case, the number of variables (features) for which VIP scores are calculated can be increased by the analyst by changing the number of top features to show.



The VIP scores plot generated by Metaboanalyst also includes qualitative information on the values of a specific variable in the classes under comparison, based on a color scale.