

L'Analisi canonica discriminante

Andrea Onofri

28/02/2017

Indice

1	Introduzione	1
2	Un caso studio	2
2.1	Qual è il principio di fondo?	4
2.2	Come si calcolano le variabili canoniche?	5
2.3	I coefficienti canonici	8
2.4	Perchè i coefficienti canonici standardizzati differiscono da un programma all'altro?	10
2.5	Punteggi canonici dei centroidi	13
2.6	La struttura canonica	14
3	Interpretazione dei risultati	16
3.1	Esame dei punteggi canonici	16
3.2	Esame dei coefficienti canonici e della struttura canonica . . .	16
3.3	Visualizzazione geometrica dei risultati dell'analisi	16
4	L'analisi canonica con R	19
5	Altre letture	25

1 Introduzione

Nell'ambito della statistica multivariata, il problema della riduzione della dimensionalità dei dati è centrale per ogni applicazione sperimentale. In particolare, se guardiamo il nostro dataset multivariato (con n soggetti/righe e p variabili/colonne) come un insieme di punti nello spazio a p dimensioni, è utile cercare di riorganizzare le osservazioni in uno spazio con un ridotto

numero di dimensioni, che ci dia un punto di vista migliore per l'interpretazione dei risultati. Se le osservazioni sono tra loro indipendenti, l'analisi delle componenti principali opera questa riduzione di dimensionalità, cercando di conservare la distanza Euclidea tra i soggetti. In questo modo è possibile, ad esempio, riportare i soggetti in uno spazio a due dimensioni (biplot) con la certezza che la distanza tra i punti nel biplot è una buona approssimazione della loro distanza originale in p dimensioni.

In molti problemi sperimentali agronomici e biologici, tuttavia, le osservazioni sperimentali non sono indipendenti l'una dall'altra, ma sono divisibili in gruppi noti a priori. Guardiamo l'esempio in Figura 1, dove sono individuabili tre gruppi (noti a priori) e i loro 'centroidi' (medie di ogni variabile).

In questa situazione la nostra attenzione può essere rivolta alla discriminazione tra gruppi ed, eventualmente, alla classificazione di nuovi individui in uno dei gruppi pre-esistenti. Di conseguenza, potremmo essere interessati a rappresentare il dataset in una nuova prospettiva che, con un ridotto numero di dimensioni, evidenzi al meglio le differenze tra gruppi, trascurando invece le differenze tra individui. Per questo fine, è chiaro che la prospettiva offerta dalle componenti principali non corrisponde adeguatamente alle nostre esigenze, perché guarda ai soggetti, ma non ai gruppi.

In ambito agronomico, due esempi sono particolarmente rilevanti: la valutazione della qualità produttiva nelle sperimentazioni varietali e la valutazione dell'efficacia di diverse strategie di diserbo. In entrambi i casi, viene osservato un numero elevato di variabili (ad esempio: contenuto in proteine, in amidi, valore nutritivo nelle sperimentazioni varietali e ricoprimento delle diverse specie infestanti nel confronto erbicida), che debbono essere preferibilmente valutate nel loro complesso e non separatamente l'una dall'altra. Inoltre, considerando ad esempio una sperimentazione varietale ripetuta in più anni, i dati registrati per ogni varietà costituiscono un gruppo da valutare nel suo complesso, mettendo quindi in luce le differenze tra varietà, e non tanto quelle tra ogni singolo dato.

In altre parole, l'analisi canonica discriminante trova applicazione quando si tratta di discriminare tra gruppi (e non tra soggetti) noti a priori. Da questo punto di vista è evidente la differenza con l'analisi *cluster* che permette di suddividere a posteriori i dati in gruppi omogenei.

2 Un caso studio

Come esempio di analisi delle variabili canoniche riportiamo una sperimentazione varietale eseguita su grano duro, in più annate consecutive.

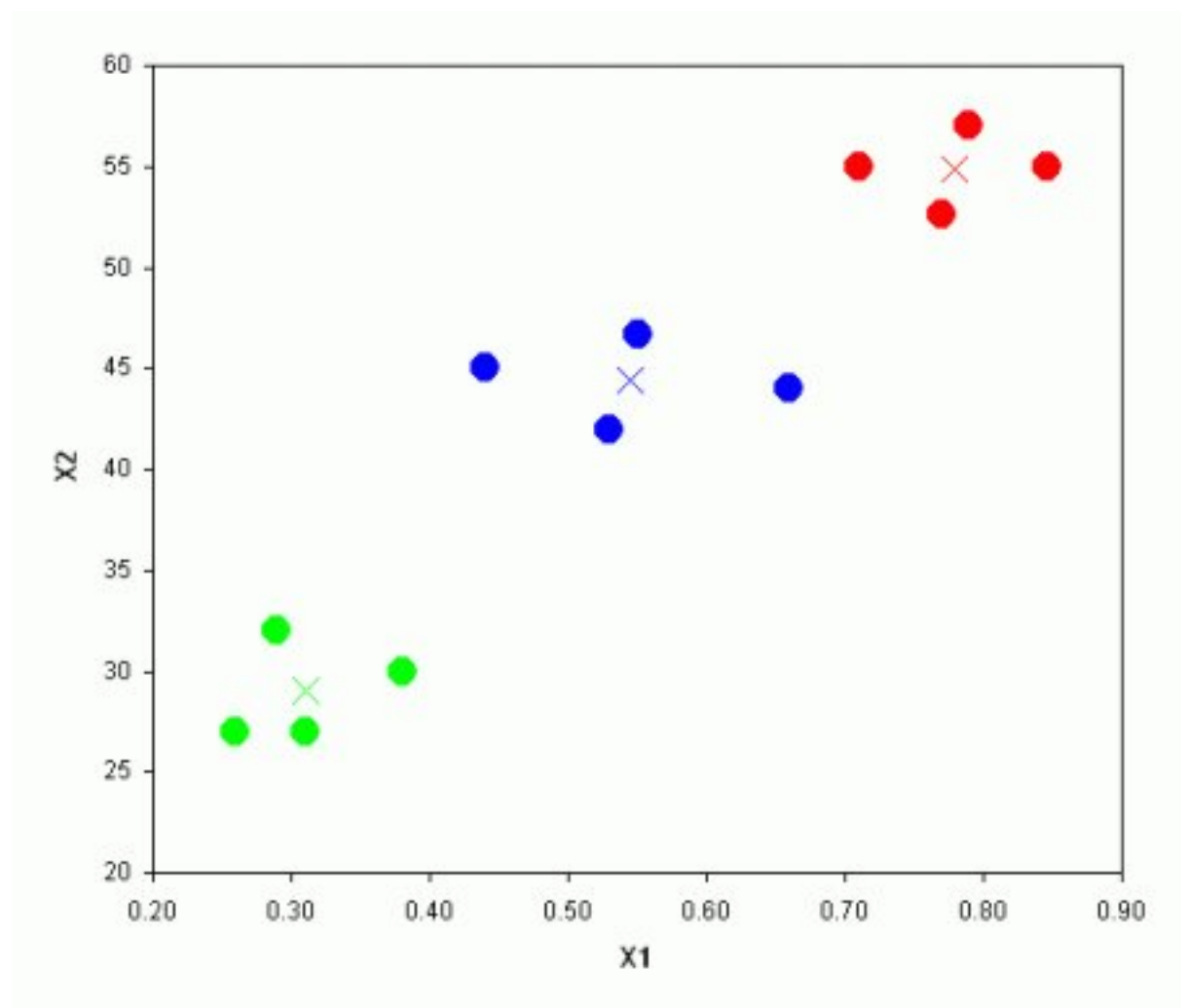


Figura 1:

Questo esempio costituisce una buona base di partenza per altri tipi di applicazioni.

I dati sono relativi a quattro prove sperimentali eseguite tra il 1999 ed il 2003, i cui risultati sono stati già pubblicati altrove (Belocchi *et al.*, 2003; Ciricofolo *et al.*, 2001; Ciricofolo *et al.*, 2002; Desiderio *et al.*, 2000). Le unità sperimentali sono 64 (16 varietà di grano duro x quattro anni) e su ognuna di esse sono state rilevate quattro caratteristiche qualitative, cioè il peso ettolitrico (WPH), la percentuale di cariossidi bianconate (YB), il peso di 1000 cariossidi (TKW) e il contenuto di proteina grezza (PC). In questa sede operiamo sulle medie delle tre repliche di ciascuna annata. I dati sono disponibili nella ‘repository’ indicata nel codice sottostante.

```
rm(list=ls())
dati <- read.csv("https://www.statforbiology.com/pca/WheatQuality.csv", header=T)
head(dati)
```

##	Genotype	Year	WPH	YB	TKW	PC
## 1	ARCOBALENO	1999	81.67	46.67	44.67	12.71
## 2	ARCOBALENO	2000	82.83	19.67	43.32	11.90
## 3	ARCOBALENO	2001	83.50	38.67	46.78	13.00
## 4	ARCOBALENO	2002	78.60	82.67	43.03	12.40
## 5	BAIO	1999	80.30	41.00	51.83	13.91
## 6	BAIO	2000	81.40	20.00	41.43	12.80

Se su questo data set calcolassimo le componenti principali riusciremmo ad evidenziare al massimo le differenze tra individui, cioè tra ogni singolo dato produttivo ottenuto in ogni anno da ciascuna varietà. In realtà il nostro obiettivo è invece quello di evidenziare al massimo le differenze tra varietà, al di là dei risultati di ogni singolo anno. Se operassimo separatamente con la statistica univariata sulle singole variabili, perderemmo ogni informazione sulle relazioni tra di esse; optiamo pertanto per l’analisi canonica discriminante.

2.1 Qual è il principio di fondo?

Come nel caso delle componenti principali, dobbiamo prendere la matrice delle osservazioni originali e ‘ruotarla’, in modo da definire delle nuove variabili, dette variabili canoniche, che abbiano le caratteristiche desiderate. La rotazione è un’operazione di moltiplicazione matriciale, del tipo:

$$VC = X \times V$$

dove X e VC hanno n righe e p colonne (n soggetti e p variabili), mentre V è una matrice quadrata $v \times v$, che definisce la rotazione. Il nostro problema è

quindi quello di trovare la matrice V , in modo che le variabili ruotate (VC) permettano di osservare meglio le differenze tra gruppi.

Possiamo iniziare pensando che i gruppi sono facili da discriminare quando gli individui nel gruppo sono molto simili (bassa variabilità entro gruppo) e le medie dei gruppi sono invece molto diverse (alta variabilità tra gruppi). In questa situazione il valore di F nell'ANOVA univariata risulta molto alto e significativo. L'analisi delle variabili canoniche sfrutta questo principio: V viene scelta in modo che le variabili canoniche assicurino la minima variabilità entro gruppo e la massima variabilità tra gruppi.

2.2 Come si calcolano le variabili canoniche?

Per il calcolo, procediamo inizialmente alla standardizzazione dei dati. Anche se ciò non è necessario, ci consente di evitare che una variabile assuma un peso maggiore rispetto ad un'altra; indicheremo con Z la matrice dei dati standardizzati, della quale mostriamo solo le prime righe:

```
dataset <- dati[,3:6]
Z <- apply(dataset, 2, scale, center=T, scale=T)
head(Z)
```

```
##           WPH           YB           TKW           PC
## [1,]  0.3375814 -0.003873758 -0.37675661 -0.5193726
## [2,]  0.7681267 -1.154020460 -0.59544503 -1.5621410
## [3,]  1.0168038 -0.344657966 -0.03495471 -0.1460358
## [4,] -0.8018791  1.529655178 -0.64242254 -0.9184568
## [5,] -0.1709075 -0.245404565  0.78310197  1.0254693
## [6,]  0.2373683 -1.139963112 -0.90160881 -0.4035095
```

ATTENZIONE: alcuni programmi statistici (come SPSS) considerano anche un ulteriore tipo di standardizzazione che impiega al posto della deviazione standard totale la deviazione standard entro gruppi. Ne parleremo tra poco.

Indichiamo inoltre con U la matrice ottenuta da Z , sostituendo ad ogni dato in ogni colonna la media del gruppo (varietà) a cui il dato appartiene (matrice dei centroidi).

```
groups <- as.factor(dati[,1])
maovst <- manova(Z ~ groups)
U <- fitted(maovst)
head(U)
```

```
##           WPH           YB           TKW           PC
## 1 0.33015821  0.006775749 -0.41239472 -0.7865016
```

```
## 2 0.33015821 0.006775749 -0.41239472 -0.7865016
## 3 0.33015821 0.006775749 -0.41239472 -0.7865016
## 4 0.33015821 0.006775749 -0.41239472 -0.7865016
## 5 0.04529309 -0.337735787 -0.04224433 0.8548930
## 6 0.04529309 -0.337735787 -0.04224433 0.8548930
```

Con questi ingredienti siamo in grado di iniziare i calcoli. Calcoliamo la matrice delle devianze/codevianze totali (T):

```
T <- t(Z) %*% Z
T
```

```
##           WPH           YB           TKW           PC
## WPH  63.000000 -26.212412  34.638616 -1.293172
## YB  -26.212412  63.000000 -3.271389 -7.052713
## TKW  34.638616 -3.271389  63.000000 30.501366
## PC  -1.293172 -7.052713 30.501366 63.000000
```

Ora calcoliamo la matrice delle devianze/codevianze tra gruppi (B):

```
B <- t(U) %*% U
B
```

```
##           WPH           YB           TKW           PC
## WPH 20.7760120 -2.707135  7.986179 -0.7945792
## YB -2.7071352 12.008904  2.639667 -8.5454789
## TKW 7.9861789  2.639667 27.190907 11.6352972
## PC -0.7945792 -8.545479 11.635297 21.0150194
```

La matrice delle devianze/codevianze entro gruppo (W) è data da $T - B = W$:

```
W <- T - B
W
```

```
##           WPH           YB           TKW           PC
## WPH 42.2239880 -23.505277 26.652437 -0.4985925
## YB -23.5052765 50.991096 -5.911057  1.4927654
## TKW 26.6524371 -5.911057 35.809093 18.8660688
## PC -0.4985925  1.492765 18.866069 41.9849806
```

Vediamo bene che esistono forti correlazioni tra le variabili, sia totali, che tra gruppi, che entro gruppo. Insomma, quattro variabili sono ridondanti per la discriminazione tra gruppi, visto che ognuna veicola parte dell'informazione già veicolata da altre. Possiamo calcolare il rapporto tra W e B, che rappresenta il rapporto tra le devianze/codevianze tra gruppi e quelle entro gruppo (una sorta di test F, per semplificare). Utilizzando la moltiplicazione matriciale possiamo scrivere:

```
WB <- solve(W)%*%B
WB
```

```
##           WPH           YB           TKW           PC
## WPH  1.2427342 -0.4431553 -1.0966307 -0.5766991
## YB   0.4148842  0.1283142 -0.2258146 -0.3764203
## TKW -0.8168720  0.7037502  1.8275515  0.5566547
## PC   0.3481454 -0.5295936 -0.5490796  0.2569372
```

Gli autovalori della matrice WB sono:

```
A <- eigen(WB)$values
A
```

```
## [1] 2.1583182 0.7538314 0.4772379 0.0661495
```

Mentre gli autovettori associati con ciascun autovalore sono:

```
V1 <- eigen(WB)$vectors
V1
```

```
##           [,1]           [,2]           [,3]           [,4]
## [1,] -0.6147150 -0.4677403  0.73129521  0.1977323
## [2,] -0.1543769 -0.5681709  0.06455426  0.7153740
## [3,]  0.7218096 -0.2985051  0.13437782 -0.3705890
## [4,] -0.2780001  0.6076969  0.66556943  0.5583957
```

La matrice degli autovettori può essere utilizzata per ‘ruotare’ la matrice dei dati originali, analogamente a quanto avviene nell’analisi delle componenti principali. In questo modo otterremo quattro variabili ‘ruotate’ dette *variabili canoniche centrate* (VCC). Queste variabili hanno:

1. media = 0
2. varianze unitarie
3. correlazioni nulle

```
VCC <- Z %*% V1
colnames(VCC) <- c("VCC1", "VCC2", "VCC3", "VCC4")
apply(VCC, 2, mean)
```

```
##           VCC1           VCC2           VCC3           VCC4
## -4.770490e-17  2.155394e-16  6.158268e-17  2.595580e-16
```

```
cor(VCC)
```

```
##           VCC1           VCC2           VCC3           VCC4
## VCC1  1.000000e+00  2.891080e-16  1.626754e-15  3.843447e-16
## VCC2  2.891080e-16  1.000000e+00 -1.021881e-15 -2.740982e-16
## VCC3  1.626754e-15 -1.021881e-15  1.000000e+00 -1.692481e-15
```

```
## VCC4 3.843447e-16 -2.740982e-16 -1.692481e-15 1.000000e+00
```

Se sottoponiamo ad ANOVA le quattro VCC, possiamo calcolare le deviazioni standard dei residui:

```
zseq <- apply(VCC, 2, function(x) lm(x ~ Genotype, data = dati))
RMSE <- lapply(zseq, function(mod) summary(mod)$sigma)
RMSE
```

```
## $VCC1
## [1] 0.3116788
##
## $VCC2
## [1] 0.7872216
##
## $VCC3
## [1] 0.9997169
##
## $VCC4
## [1] 0.8114784
```

Queste deviazioni standard risulteranno utili tra poco.

2.3 I coefficienti canonici

Sappiamo che il calcolo degli autovettori di una matrice non ammette un'unica soluzione, ma infinite. In questo caso, gli autovettori indicati in precedenza sono stati scelti in modo che la loro norma fosse unitaria. Infatti:

```
t(V1) %*% V1
```

```
##           [,1]           [,2]           [,3]           [,4]
## [1,]  1.000000000 -0.009164206 -0.54753700 -0.65471505
## [2,] -0.009164206  1.000000000 -0.01438206 -0.04898404
## [3,] -0.547537002 -0.014382063  1.00000000  0.51263328
## [4,] -0.654715051 -0.048984043  0.51263328  1.00000000
```

Più comunemente, gli autovettori vengono scalati dividendoli proprio per la deviazione standard entro gruppi della variabile canonica centrata corrispondente. In questo modo si ottengono i *coefficienti canonici standardizzati* (Vst):

```
scaling <- diag(1/unlist(RMSE))
Vst <- V1 %*% scaling
Vst
```

```
##           [,1]           [,2]           [,3]           [,4]
```



```
## [1,] -1.9722706 -0.5941660 0.73150227 0.2436692
## [2,] -0.4953078 -0.7217420 0.06457254 0.8815688
## [3,] 2.3158763 -0.3791882 0.13441587 -0.4566838
## [4,] -0.8919442 0.7719516 0.66575788 0.6881214
```

I coefficienti canonici standardizzati sono usualmente utilizzati al posto degli autovettori V1 per ruotare le variabili originali, ottenendo le **Variabili Canoniche Standardizzate**, dette anche punteggi canonici (*canonical scores*):

```
VCst <- Z %*% Vst
colnames(VCst) <- c("VC1", "VC2", "VC3", "VC4")
cor(VCst)

##           VC1           VC2           VC3           VC4
## VC1 1.000000e+00 2.693048e-16 1.612146e-15 3.244759e-16
## VC2 2.693048e-16 1.000000e+00 -1.138091e-15 -2.357008e-16
## VC3 1.612146e-15 -1.138091e-15 1.000000e+00 -1.737112e-15
## VC4 3.244759e-16 -2.357008e-16 -1.737112e-15 1.000000e+00

zseq <- apply(VCst, 2, function(x) lm(x ~ Genotype, data = dati))
RMSE <- lapply(zseq, function(mod) summary(mod)$sigma)
RMSE
```

```
## $VC1
## [1] 1
##
## $VC2
## [1] 1
##
## $VC3
## [1] 1
##
## $VC4
## [1] 1
```

```
MSB <- lapply(zseq, function(mod) anova(mod)[1,2])
MSW <- lapply(zseq, function(mod) anova(mod)[2,2])
unlist(MSB)/unlist(MSW)
```

```
##           VC1           VC2           VC3           VC4
## 2.1583182 0.7538314 0.4772379 0.0661495
```

Vediamo più sopra che le variabili canoniche standardizzate: 1. non sono correlate tra loro (notare gli zero fuori dalla diagonale principale nella matrice di correlazione) 2. hanno varianze entro gruppi unitarie 3. i rapporti devianza tesi/devianza errore sono pari agli autovalori. La loro somma è

pari a $2.158 + 0.754 + 0.477 + 0.066 = 3.455$, il che eguaglia esattamente la somma della diagonale principale di $W^{-1}B$.

Quest'ultimo aspetto può essere ulteriormente evidenziato se calcoliamo per le variabili canoniche la matrice $W_y^{-1}B_y$, in analogia alla matrice $W^{-1}B$ già calcolata per le variabili originali.

```
maovst2 <- manova(VCst ~ groups)
U2 <- maovst2$fitted
T2 <- t(VCst)%*%VCst
B2 <- t(U2)%*%U2
W2 <- T2 - B2
solve(W2)%*%B2
```

```
##              VC1              VC2              VC3              VC4
## VC1  2.158318e+00 -1.259259e-16  5.735773e-16  7.809273e-16
## VC2 -7.139095e-16  7.538314e-01  7.998439e-19  1.625743e-16
## VC3 -2.669248e-15  3.283524e-16  4.772379e-01 -2.328594e-16
## VC4  1.119665e-15  5.061405e-16  5.192494e-16  6.614950e-02
```

Possiamo osservare che questa matrice conserva la traccia di $W^{-1}B$, a dimostrazione che il contributo totale delle quattro variabili canoniche alla discriminazione tra gruppi è lo stesso delle quattro variabili originali, ma cambiano i rapporti. Ora la prima variabile canonica da sola garantisce una discriminazione pari al 62.5% del totale delle variabili originali ($2.155/3.540 \cdot 100$), mentre due variabili canoniche arrivano quasi all'85% del totale.

****** Se quindi usiamo due variabili canoniche invece che le quattro variabili originali, riduciamo la dimensionalità del problema perdendo solo una piccola parte del potere discriminatorio iniziale. E' un bel guadagno!******

A questi risultati si arriva in modi certamente più ortodossi di quelli qui accennati, ma per questo si rimanda ai testi specializzati (Sadocchi, 1986). A me basta che abbiate compreso i criteri fondamentali di calcolo.

2.4 Perchè i coefficienti canonici standardizzati differiscono da un programma all'altro?

In questo tutorial abbiamo iniziato l'analisi standardizzando i dati, cioè sottraendo da ogni dato la media della colonna e dividendo per la sua deviazione standard. Alcuni software statistici (come SPSS o WINSTAT 6.0) eseguono un diverso tipo di standardizzazione, dividendo ogni dato centrato per la deviazione standard entro gruppo. Nel codice sottostante calcoliamo, con quattro ANOVA, la deviazione standard del residuo per le quattro variabili e la utilizziamo per standardizzare i dati, producendo la matrice Zw .

```
rm(scaling, T, T2, U, U2, W, W2, WB)

#Calculate the RMSE for the original variables
zseq <- apply(dataset, 2, function(x) lm(x ~ Genotype, data = dati))
RMSE <- lapply(zseq, function(mod) summary(mod)$sigma)

#Standardise by using the above RMSE
Zw <- t(t(apply(dati[,3:6], 2, function(x) scale(x, center=T, scale=F)) )/unlist(RMSE))
head(Zw)
```

```
##           WPH           YB           TKW           PC
## [1,]  0.3599312 -0.003758425 -0.43619917 -0.5553320
## [2,]  0.8189811 -1.119662034 -0.68939104 -1.6702976
## [3,]  1.0841220 -0.334396532 -0.04046967 -0.1561468
## [4,] -0.8549680  1.484113053 -0.74378040 -0.9820472
## [5,] -0.1822225 -0.238098183  0.90665544  1.0964689
## [6,]  0.2530834 -1.106023212 -1.04385964 -0.4314470
```

Se iniziamo le analisi da Zw invece che da Z , i coefficienti canonici standardizzati che otteniamo (Vw), sono diversi da quelli calcolati in precedenza:

```
#Calcolo coefficienti canonici a partire da Zw
T <- t(Zw) %*% Zw
maovst <- manova(Zw ~ groups)
U <- fitted(maovst)
B <- t(U)%*%U
W <- T - B
WB <- solve(W)%*%B
V1w <- eigen(WB)$vectors
VCCw <- Zw %*% V1w
zseq <- apply(VCCw, 2, function(x) lm(x ~ Genotype, data = dati))
RMSE <- lapply(zseq, function(mod) summary(mod)$sigma)
scaling <- diag(1/unlist(RMSE))
Vw <- V1w %*% scaling
Vw
```

```
##           [,1]           [,2]           [,3]           [,4]
## [1,] -1.8498031 -0.5572715  0.68607988  0.2285387
## [2,] -0.5105070 -0.7438897  0.06655404  0.9086210
## [3,]  2.0002828 -0.3275147  0.11609849 -0.3944497
## [4,] -0.8341882  0.7219655  0.62264811  0.6435635
```

Possiamo tuttavia notare che il rapporto tra Vw e Vst è costante (0.938, 1.031, 0.864 e 0.935) e riflette il rapporto tra la deviazione standard totale

e la deviazione standard entro gruppi delle variabili originali.

`Vw/Vst`

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.9379053 0.9379053 0.9379053 0.9379053
## [2,] 1.0306864 1.0306864 1.0306864 1.0306864
## [3,] 0.8637261 0.8637261 0.8637261 0.8637261
## [4,] 0.9352471 0.9352471 0.9352471 0.9352471

sdTot <- apply(dataset, 2, sd)
zseq <- apply(dati[3:6], 2, function(x) lm(x ~ Genotype, data = dati))
RMSE <- lapply(zseq, function(mod) summary(mod)$sigma)
unlist(RMSE)/sdTot

##          WPH          YB          TKW          PC
## 0.9379053 1.0306864 0.8637261 0.9352471
```

Se ruotiamo Zw utilizzando Vw arriviamo alle stesse variabili canoniche già determinate in precedenza (anche qui mostriamo solo le prime sei righe).

`head(Zw %*% Vw)`

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] -1.0731535 -0.4558524 -0.1497271 -0.10648959
## [2,] -0.9289930 -0.6036012 -0.6326765 -1.63319215
## [3,] -1.7853954 -0.4548743  0.6196159 -0.14060305
## [4,]  0.1553135 -1.0929723 -1.1856243  0.81447716
## [5,]  1.3575325  0.7733359  0.6471100  0.09003153
## [6,] -1.6316285  0.7121127 -0.2898050 -0.81303000
```

Oltre ai coefficienti canonici standardizzati, abbiamo anche i cosiddetti *raw coefficients* (Vr) che si ottengono iniziando le analisi dalla matrice dei dati centrati e non standardizzati (Zc).

```
#Calcolo coefficienti canonici a partire da Zc
rm(scoring, T, U, W, WB)
Zc <- apply(dati[,3:6], 2, function(x) scale(x, center=T, scale=F))
T <- t(Zc) %*% Zc
maovst <- manova(Zc ~ groups)
U <- fitted(maovst)
B <- t(U)%*%U
W <- T - B
WB <- solve(W)%*%B
V1r <- eigen(WB)$vectors
VCCr <- Zc %*% V1r
zseq <- apply(VCCr, 2, function(x) lm(x ~ Genotype, data = dati))
```

```
RMSE <- lapply(zseq, function(mod) summary(mod)$sigma)
scaling <- diag(1/unlist(RMSE))
Vr <- Vlr %*% scaling
Vr
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] -0.73202750 -0.22053053 0.271504213 0.09044022
## [2,] -0.02109913 -0.03074479 0.002750663 0.03755309
## [3,] 0.37515210 -0.06142523 0.021774217 -0.07397885
## [4,] -1.14826063 0.99378600 0.857075589 0.88586568
```

Anche in questo caso, se ruotiamo Zc utilizzando Vr , arriviamo agli stessi punteggi canonici.

```
head(Zc %*% Vr)
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] -1.0731535 -0.4558524 -0.1497271 -0.10648959
## [2,] -0.9289930 -0.6036012 -0.6326765 -1.63319215
## [3,] -1.7853954 -0.4548743 0.6196159 -0.14060305
## [4,] 0.1553135 -1.0929723 -1.1856243 0.81447716
## [5,] 1.3575325 0.7733359 0.6471100 0.09003153
## [6,] -1.6316285 0.7121127 -0.2898050 -0.81303000
```

Insomma i coefficienti canonici possono cambiare perchè la matrice di partenza può essere diversa. Tuttavia le variabili canoniche non cambiano: esse possono essere ottenute (indifferentemente):

1. moltiplicando Z per Vst (come abbiamo fatto più sopra);
2. moltiplicando Zw per Vw ;
3. moltiplicando Zc per Vr (“raw coefficients”).

2.5 Punteggi canonici dei centroidi

Riepilogando, i coefficienti canonici sono un metodo per riorganizzare le nostre osservazioni sperimentali in uno spazio diverso, con meno dimensioni, ma senza perdere la capacità di differenziare i gruppi (ricordiamo che nel caso della PCA si cercava invece di non perdere la distanza tra individui!).

In realtà, più che le variabili canoniche, ci interessa calcolare i punteggi canonici dei centroidi di ogni gruppo (le medie delle variabili originali), cioè la loro posizione nel nuovo spazio di riferimento dopo la rotazione. Le nuove coordinate dei centroidi (punteggi canonici dei centroidi) possono essere ottenute allo stesso modo delle variabili canoniche, moltiplicando la matrice delle medie delle variabili standardizzate (con 16 elementi, cioè le sedici medie per

ogni combinazione varietà x anno) per la matrice dei coefficienti canonici standardizzati (Vst). I punteggi, ottenuti con R, sono i seguenti:

```
medie <- aov(Z ~ groups-1)$coefficients
scores <- medie %*% Vst
scores
```

```
##           [,1]      [,2]      [,3]      [,4]
## groupsARCOBALENO -0.9080571 -0.6518251 -0.3371030 -0.26645191
## groupsBAIO       -0.7823965  0.8928011  0.5747970  0.32086167
## groupsCLAUDIO    -0.5496314 -1.3845288  0.3590569  0.16423169
## groupsCOLORADO   -1.1481765  1.4231696 -0.5535939 -0.13201115
## groupsCOLOSSEO    1.0654126 -1.3108147 -0.1201515 -0.05709275
## groupsCRESO       0.3070820 -0.3947049  0.6469757 -0.09565658
## groupsDUILIO      0.2985911 -0.3083817  0.3302409  0.39103654
## groupsGIOTTO      0.5551808 -0.1276729 -0.6353720 -0.14515354
## groupsGRAZIA     -2.7043586  0.1540981  0.2141260 -0.09434961
## groupsIRIDE       -0.9975478 -0.6185305 -0.7408252 -0.13364342
## groupsMERIDIANO   0.7232796  0.1328308 -0.1189112  0.31997541
## groupsSANCARLO    -0.6322898  0.7065535  0.9005722 -0.01388980
## groupsSIMETO      3.1209198  0.4928771  0.5853148 -0.18719341
## groupsSOLEX       -0.3851960 -0.1854706  0.1043439  0.02715539
## groupsTORREBIANCA 0.9167304  0.5916795  0.2310080 -0.37993393
## groupsVERDI       1.1204575  0.5879195 -1.4404785  0.28211539
```

2.6 La struttura canonica

Immaginiamo di calcolare, oltre alle variabili canoniche, anche i punteggi canonici relativi alla matrice delle medie delle variabili standardizzate (U) e quella dei residui (differenza tra i dati standardizzati e le loro medie, cioè $Z - U$). Otteniamo rispettivamente Vu e Vzu

```
VU <- U%*%Vst
head(VU)
```

```
##           [,1]      [,2]      [,3]      [,4]
## 1 -7.183974 -0.1496154 -0.08796465  1.099194
## 2 -7.183974 -0.1496154 -0.08796465  1.099194
## 3 -7.183974 -0.1496154 -0.08796465  1.099194
## 4 -7.183974 -0.1496154 -0.08796465  1.099194
## 5  2.490094  6.2612890 -0.01564151 -6.383678
## 6  2.490094  6.2612890 -0.01564151 -6.383678
```

```
VZU <- (Z-U)%*%Vst
head(VZU)
```

```
##          [,1]          [,2]          [,3]          [,4]
## [1,]  6.110820 -0.3062371 -0.06176242 -1.2056833
## [2,]  6.254981 -0.4539858 -0.54471185 -2.7323858
## [3,]  5.398578 -0.3052589  0.70758051 -1.2397967
## [4,]  7.339287 -0.9433569 -1.09765964 -0.2847165
## [5,] -1.132562 -5.4879531  0.66275151  6.4737092
## [6,] -4.121723 -5.5491763 -0.27416353  5.5706477
```

Le nuove variabili così ottenute, non sono correlate tra loro, ma hanno un certo grado di correlazione con le variabili originali. La matrice di correlazione delle variabili canoniche (*VCst*) con le variabili originali (*Z*) prende il nome di **Struttura Canonica Totale**.

```
cor(Z, VCst)
```

```
##          VC1          VC2          VC3          VC4
## WPH -0.3059273 -0.4482850  0.7209655 -0.43088021
## YB   0.1965463 -0.4682288 -0.3028492  0.80648226
## TKW  0.5320720 -0.2548998  0.8064667 -0.03920395
## PC   0.2096499  0.5894317  0.6679134  0.40312384
```

Allo stesso modo, la correlazione tra *U* e *Vu* prende il nome di Struttura Canonica tra gruppi ('between'):

```
cor(U, VU)
```

```
##          [,1]          [,2]          [,3]          [,4]
## WPH  0.1113207 -0.02298862  0.7867065 -0.2081955
## YB  -0.3336955 -0.96964998  0.2556828  0.9840392
## TKW  0.8358955 -0.34964079  0.7717052 -0.0294356
## PC   0.7895205  0.45777788  0.1090482 -0.6304306
```

Ancora, la correlazione tra i residui (*Z - U*) e i loro punteggi canonici (*Vzu*) prende il nome di Struttura Canonica entro gruppi ('within'):

```
cor((Z-U), VZU)
```

```
##          [,1]          [,2]          [,3]          [,4]
## WPH  0.1308540 -0.01011711  0.6889033 -0.18997474
## YB  -0.4177260 -0.96838132  0.4180063  0.98299197
## TKW  0.7929795 -0.34729954  0.6239354 -0.02966873
## PC  -0.1111922 -0.00194042  0.4295819  0.14763277
```

La struttura canonica è estremamente importante perchè rappresenta il collegamento tra le variabili canoniche e le variabili originali e quindi può aiutarci ad interpretare i risultati dell'analisi.

3 Interpretazione dei risultati

3.1 Esame dei punteggi canonici

L'esame dei punteggi canonici mostra che SIMETO e GRAZIA sono più distanti sulla prima variabile canonica, mentre CLAUDIO e COLORADO sono molto distanti sulla seconda. Inoltre, alcune varietà sono caratterizzate da punteggi positivi o negativi su entrambe le variabili canoniche, mentre altre sono caratterizzate da situazioni opposte (positive su una variabile e negative sull'altra). In ogni caso, emergono chiaramente le differenze tra varietà, sotto forma di distanze su una o entrambe le variabili canoniche (anche se le distanze euclidea tra centroidi non rappresentano al meglio le distanze euclidee tra individui, come nel caso delle componenti principali).

3.2 Esame dei coefficienti canonici e della struttura canonica

Le considerazioni appena esposte non hanno molto senso se non si riesce a comprendere come ogni variabile canonica si rapporta con le variabili originali. Per questo è necessario considerare, oltre ai punteggi canonici delle medie, anche la struttura canonica o i coefficienti canonici.

In particolare, questi ultimi costituiscono il contributo unico (peso) che ogni variabile originale ha dato nella creazione dei punteggi canonici delle medie (si veda in precedenza l'esempio d'uso delle funzioni canoniche). Per esempio, considerando il primo vettore dei coefficienti canonici (quello che crea la prima variabile canonica) possiamo notare che l'alto coefficiente relativo al peso di 1000 cariossidi implica che le varietà con valori elevati per questa caratteristica avranno probabilmente un elevato valore positivo sulla prima variabile canonica. Allo stesso modo, il valore negativo per il coefficiente del peso ettolitrico implica che le varietà con un elevato peso ettolitrico avranno probabilmente valori negativi sulla prima variabile canonica.

Considerazioni analoghe potrebbero essere fatte osservando la struttura canonica.

3.3 Visualizzazione geometrica dei risultati dell'analisi

Come già detto più volte, il calcolo delle variabili canoniche ci permette di riorganizzare nello spazio le nostre osservazioni, in modo più agevole da comprendere e spiegare. Ovviamente, per poter rappresentare in un grafico il risultato di questa riorganizzazione dobbiamo limitarci alle prime due variabili canoniche, anche se abbiamo già visto come con esse, in questo

esempio, si riesca a conservare un potere discriminante pari all'85% di quello originale.

Possiamo quindi utilizzare un biplot, nel quale vengono riportati i punteggi canonici delle medie, insieme ai coefficienti canonici standardizzati, in forma di vettori geometrici (uno per ogni variabile originale). Questo biplot (CVA biplot) non ha la stessa interpretazione geometrica di un PCA biplot, perchè è frutto di elaborazioni algebriche/geometriche diverse. Per la sua interpretazione dovremo considerare che:

1 - La distanza euclidea tra i soggetti (le varietà) nel biplot approssima la loro distanza di Mahalanobis nella scala originale ed è quindi la più adatta per valutare le differenze tra gruppi. 2 - I vettori geometrici 'tirano' le medie lungo la loro direzione in modo tanto più intenso quanto più è alto il loro valore originale della caratteristica rilevata. In questo modo si può facilmente comprendere quali caratteristiche contribuiscono più delle altre alla discriminazione.

Si può osservare ad esempio che Colorado, Baio e Sancarolo sono distinguibili per il loro alto contenuto in proteine (il loro punteggio canonico giace lungo la direzione del vettore corrispondente) associato ad un basso grado di bianconatura (il vettore relativo a questa caratteristica punta in direzione opposta). D'altra parte, Claudio e Colosseo sono discriminate dal loro basso contenuto in proteine e da un'alta percentuale di bianconatura.

Grazia ha mostrato un contenuto di proteine abbastanza alto, associato ad un buon peso ettolitrico (la sua posizione in figura risulta dalla combinazione dei due vettori corrispondenti alle caratteristiche sopracitate) ed un basso peso dei 1000 semi.

Arcobaleno e Iride hanno invece mostrato un alto peso ettolitrico, un alto grado di bianconatura, un basso contenuto di proteine e un basso peso dei 1000 semi.

Si deve comunque notare che alcune situazioni sono poco "chiare"; in particolare alcune varietà si trovano vicine all'origine degli assi, il che significa che non hanno mostrato caratteristiche di spicco (in senso negativo o positivo) e si sono mantenute vicino alla media. Inoltre, Verdi mostra un punteggio abbastanza poco chiaro, dato che la sua posizione sul biplot non evidenzia l'alto grado di bianconatura che era invece chiaro dai dati originali (non dimentichiamoci che parte dell'informazione iniziale è comunque persa nel processo di combinazione lineare delle variabili originali).

Insomma, anche se l'analisi canonica riveste un'indubbia utilità per riassumere i dati, essa non potrà mai sostituire un'accurata analisi puntiforme dei risultati dell'esperimento, altrimenti il rischio di perdere o confondere le informazioni può essere abbastanza serio.

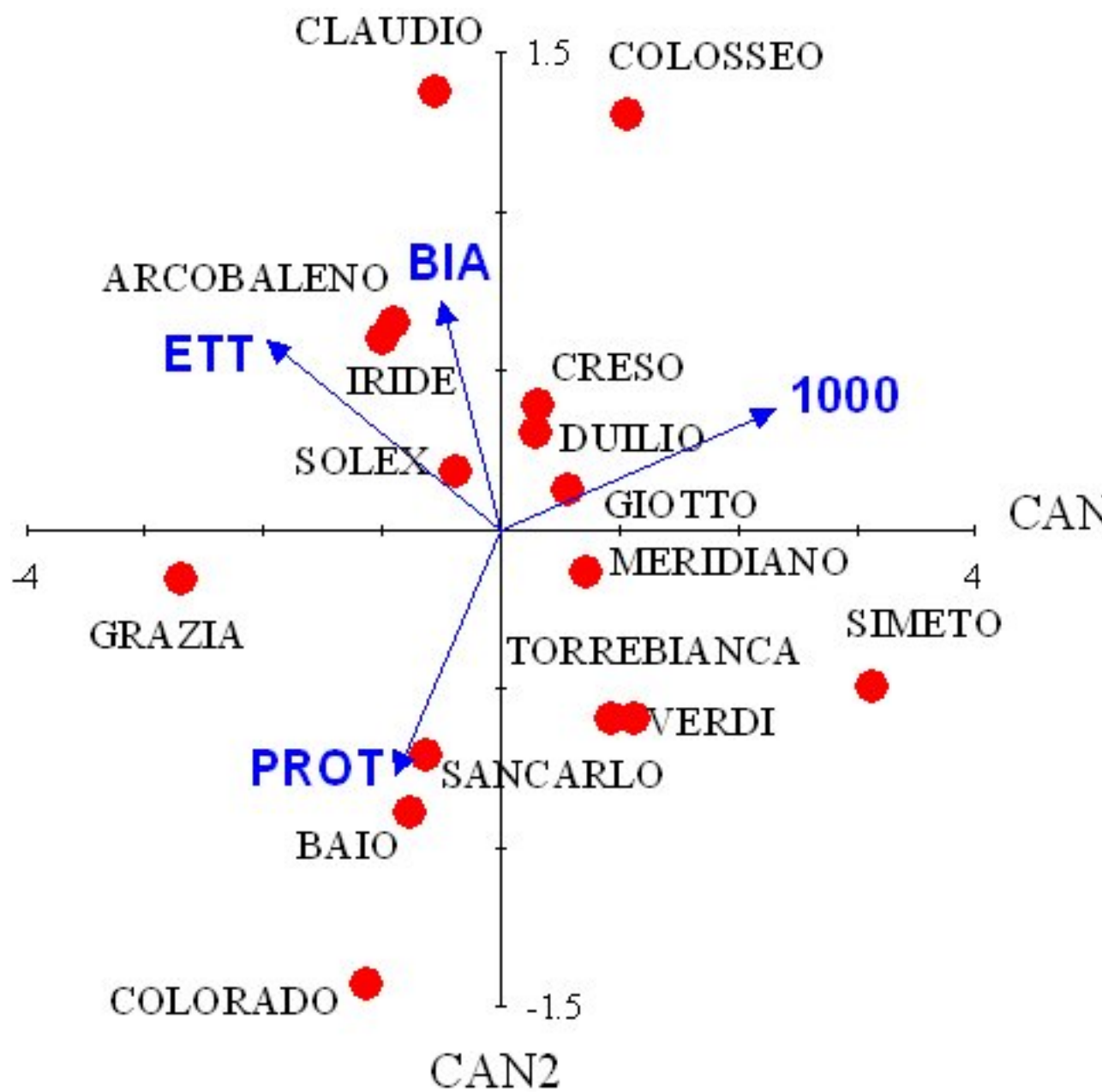


Figura 2:

4 L'analisi canonica con R

L'analisi canonica discriminante, in R, può essere eseguita con il package MASS e la funzione lda(). Vediamo un esempio.

```
library(MASS)
cva1 <- lda(groups ~ WPH + YB + TKW + PC, data=as.data.frame(Zw))
cva2 <- lda(groups ~ WPH + YB + TKW + PC, data=dati)
cva1
```

```
## Call:
## lda(groups ~ WPH + YB + TKW + PC, data = as.data.frame(Zw))
##
## Prior probabilities of groups:
##   ARCOBALENO      BAI0      CLAUDIO      COLORADO      COLOSSEO      CRESO
##      0.0625      0.0625      0.0625      0.0625      0.0625      0.0625
##   DUILIO      GIOTTO      GRAZIA      IRIDE      MERIDIANO      SANCARLO
##      0.0625      0.0625      0.0625      0.0625      0.0625      0.0625
##   SIMETO      SOLEX TORREBIANCA      VERDI
##      0.0625      0.0625      0.0625      0.0625
##
## Group means:
##           WPH           YB           TKW           PC
## ARCOBALENO  0.352016564  0.006574015 -0.47746007 -0.84095592
## BAI0         0.048291751 -0.327680445 -0.04890940  0.91408252
## CLAUDIO     0.864490354  0.519683027  0.44293924 -0.51403700
## COLORADO    -0.680845668 -0.661728257 -1.29986475  0.17421338
## COLOSSEO    0.260008722  0.629826846  0.65487020 -0.66889333
## CRESO       0.616168112 -0.069369425  0.79693897  0.21894965
## DUILIO      0.104683654  0.405922853  0.46825842  0.28433343
## GIOTTO     -0.450331396  0.168380039 -0.29881915 -0.48650698
## GRAZIA      0.708175955 -0.534225937 -0.92007695 -0.20776558
## IRIDE       -0.003153495  0.209606478 -0.88350480 -1.04398978
## MERIDIANO   -0.456267386  0.347441238  0.13254477  0.24992092
## SANCARLO    0.500416310 -0.616885464  0.36182406  0.89343501
## SIMETO     -0.340515584 -0.134670451  1.63809860  1.02420258
## SOLEX       0.219446124  0.020212837 -0.01515049 -0.07355676
## TORREBIANCA -0.076364036 -0.513664380  0.43543725  0.42886601
## VERDI      -1.666219982  0.550577025 -0.98712591 -0.35229816
##
## Coefficients of linear discriminants:
##           LD1           LD2           LD3           LD4
## WPH  1.8498031 -0.5572715  0.68607988  0.2285387
## YB   0.5105070 -0.7438897  0.06655404  0.9086210
```

```
## TKW -2.0002828 -0.3275147 0.11609849 -0.3944497
## PC 0.8341882 0.7219655 0.62264811 0.6435635
##
## Proportion of trace:
## LD1 LD2 LD3 LD4
## 0.6246 0.2182 0.1381 0.0191
```

```
cva2
```

```
## Call:
## lda(groups ~ WPH + YB + TKW + PC, data = dati)
##
## Prior probabilities of groups:
## ARCOBALENO      BAI0      CLAUDIO      COLORADO      COLOSSEO      CRESO
## 0.0625      0.0625      0.0625      0.0625      0.0625      0.0625
## DUILIO      GIOTTO      GRAZIA      IRIDE      MERIDIANO      SANCARLO
## 0.0625      0.0625      0.0625      0.0625      0.0625      0.0625
## SIMETO      SOLEX TORREBIANCA      VERDI
## 0.0625      0.0625      0.0625      0.0625
##
## Group means:
##           WPH      YB      TKW      PC
## ARCOBALENO 81.6500 46.9200 44.4500 12.5025
## BAI0        80.8825 38.8325 46.7350 13.7775
## CLAUDIO     82.9450 59.3350 49.3575 12.7400
## COLORADO    79.0400 30.7500 40.0650 13.2400
## COLOSSEO    81.4175 62.0000 50.4875 12.6275
## CRESO       82.3175 45.0825 51.2450 13.2725
## DUILIO      81.0250 56.5825 49.4925 13.3200
## GIOTTO      79.6225 50.8350 45.4025 12.7600
## GRAZIA      82.5500 33.8350 42.0900 12.9625
## IRIDE       80.7525 51.8325 42.2850 12.3550
## MERIDIANO   79.6075 55.1675 47.7025 13.2950
## SANCARLO    82.0250 31.8350 48.9250 13.7625
## SIMETO      79.9000 43.5025 55.7300 13.8575
## SOLEX       81.3150 47.2500 46.9150 13.0600
## TORREBIANCA 80.5675 34.3325 49.3175 13.4250
## VERDI       76.5500 60.0825 41.7325 12.8575
##
## Coefficients of linear discriminants:
##           LD1           LD2           LD3           LD4
## WPH 0.73202750 -0.22053053 0.271504213 0.09044022
## YB 0.02109913 -0.03074479 0.002750663 0.03755309
## TKW -0.37515210 -0.06142523 0.021774217 -0.07397885
## PC 1.14826063 0.99378600 0.857075589 0.88586568
```

```
##
## Proportion of trace:
##      LD1      LD2      LD3      LD4
## 0.6246 0.2182 0.1381 0.0191

#Standardised coefficients
cva1$scaling

##           LD1           LD2           LD3           LD4
## WPH  1.8498031 -0.5572715 0.68607988  0.2285387
## YB   0.5105070 -0.7438897 0.06655404  0.9086210
## TKW -2.0002828 -0.3275147 0.11609849 -0.3944497
## PC   0.8341882  0.7219655 0.62264811  0.6435635

#Raw coefficients
cva2$scaling

##           LD1           LD2           LD3           LD4
## WPH  0.73202750 -0.22053053 0.271504213  0.09044022
## YB   0.02109913 -0.03074479 0.002750663  0.03755309
## TKW -0.37515210 -0.06142523 0.021774217 -0.07397885
## PC   1.14826063  0.99378600 0.857075589  0.88586568

#Standardised canonical scores (same for the two models)
predict(cva1)$x

##           LD1           LD2           LD3           LD4
## 1  1.073153506 -0.45585244 -0.14972707 -0.106489594
## 2  0.928993049 -0.60360124 -0.63267650 -1.633192146
## 3  1.785395429 -0.45487432  0.61961586 -0.140603051
## 4 -0.155313506 -1.09297229 -1.18562429  0.814477164
## 5 -1.357532525  0.77333586  0.64711000  0.090031534
## 6  1.631628473  0.71211271 -0.28980505 -0.813029998
## 7  1.704556714  0.97861842  1.66582542 -0.105788841
## 8  1.150933490  1.10713723  0.27605759  2.112234000
## 9  0.721401913 -0.53181558  1.30586251  0.559308338
## 10 0.984795331 -3.26773787 -0.66158528 -0.841062067
## 11 -0.007683283 -0.56273924  1.32036119 -0.394483262
## 12 0.500011796 -1.17582234 -0.52841100  1.333163739
## 13 0.498928370  1.75636550  0.38231511  0.046937171
## 14 1.836911158  3.20050567 -2.15911676 -0.381000628
## 15 2.130750369  0.80295634  0.87512235 -0.678547198
## 16 0.126116252 -0.06714915 -1.31269634  0.484566044
## 17 -2.782851176  0.35071464 -0.11046458  0.344013761
## 18 -0.642560311 -2.08374141 -0.63044342 -1.287281383
## 19 -1.327357155 -1.61668515  0.69182061 -0.433948474
```

```

## 20  0.491118338 -1.89354674 -0.43151842  1.148845112
## 21 -0.785935664  0.01652167  0.63400281 -0.064878505
## 22 -0.673122025 -1.22881317 -0.20366993 -1.720566060
## 23 -0.445216904 -0.69734920  1.96035999 -0.122777823
## 24  0.675946569  0.33082119  0.19721004  1.525596069
## 25 -1.221961787  0.49361270 -0.10362775 -0.018866720
## 26  0.908309497  0.48841310  0.95448437  0.062422476
## 27 -0.360343894 -1.16827373  1.11890066  0.185019501
## 28 -0.520368190 -1.04727867 -0.64879357  1.335570898
## 29 -0.173876675  1.43489327  0.20603955  0.400678316
## 30 -0.130535682  0.10971223 -1.22779624 -0.777575490
## 31  0.233187104 -0.79152725  0.33175482 -0.684114760
## 32 -2.149497944 -1.26376986 -1.85148630  0.480397786
## 33  2.239733237  1.23874518  0.93266376  0.647426902
## 34  2.209112384 -0.48293835 -1.68028741 -1.636308824
## 35  2.768367726 -0.57520591  1.26191200 -0.898297516
## 36  3.600220924  0.43579137  0.34221560  1.509781011
## 37  0.566502898  0.69076574 -0.16404874  0.209131771
## 38  0.961275613 -0.47662717 -1.02506000 -1.791531945
## 39  1.216547586 -1.45519677 -0.02523689  0.008630304
## 40  1.245865010 -1.23306378 -1.74895530  1.039196191
## 41  0.820337436  1.83606530 -0.33505589  0.026776247
## 42 -1.619702417  0.75851234 -0.09634714  0.412180227
## 43 -1.239882223 -0.80407986  0.56985685 -0.298762097
## 44 -0.853871231 -1.25917477 -0.61409870  1.139707257
## 45  0.980964869 -0.64434505  0.54229945  1.380265560
## 46  0.073861054  1.52784740  1.73922235 -0.032760932
## 47  1.300593410  1.71419789 -0.87616067 -0.942296732
## 48  0.173739755  0.22851385  2.19692765 -0.460767090
## 49 -4.361464687  0.55613107  0.21418049 -0.190905979
## 50 -2.843463727  0.43399161  1.17028062 -1.312918138
## 51 -3.305806348 -0.18258318  0.81247078 -0.533900499
## 52 -1.972944509  1.16396885  0.14432742  1.288950991
## 53 -0.958548674  0.12205725  0.83616256  0.197656186
## 54  2.099420550 -0.10600435 -1.10095866 -0.899847900
## 55  0.381474017 -0.82852244  1.17650610 -0.347058493
## 56  0.018438101  0.07058706 -0.49433443  1.157871773
## 57 -1.905224275  1.30165432  0.58062557 -0.325129341
## 58  0.704444138  0.80873591 -0.60555754 -1.334544513
## 59 -1.513273370  0.09322248  1.19347623 -0.842976907
## 60 -0.952868063  0.16310523 -0.24451245  0.982915023
## 61 -2.563767080  0.45458014 -1.22610987  0.192770222
## 62 -1.809022268  2.81536839 -2.72445356 -0.823495613
## 63  0.563300847 -0.15546830  0.34514241  0.686425459

```

```
## 64 -0.672341319 -0.76280237 -2.15649295 1.072761489
```

```
predict(cva2)$x
```

##		LD1	LD2	LD3	LD4
## 1		1.073153506	-0.45585244	-0.14972707	-0.106489594
## 2		0.928993049	-0.60360124	-0.63267650	-1.633192146
## 3		1.785395429	-0.45487432	0.61961586	-0.140603051
## 4		-0.155313506	-1.09297229	-1.18562429	0.814477164
## 5		-1.357532525	0.77333586	0.64711000	0.090031534
## 6		1.631628473	0.71211271	-0.28980505	-0.813029998
## 7		1.704556714	0.97861842	1.66582542	-0.105788841
## 8		1.150933490	1.10713723	0.27605759	2.112234000
## 9		0.721401913	-0.53181558	1.30586251	0.559308338
## 10		0.984795331	-3.26773787	-0.66158528	-0.841062067
## 11		-0.007683283	-0.56273924	1.32036119	-0.394483262
## 12		0.500011796	-1.17582234	-0.52841100	1.333163739
## 13		0.498928370	1.75636550	0.38231511	0.046937171
## 14		1.836911158	3.20050567	-2.15911676	-0.381000628
## 15		2.130750369	0.80295634	0.87512235	-0.678547198
## 16		0.126116252	-0.06714915	-1.31269634	0.484566044
## 17		-2.782851176	0.35071464	-0.11046458	0.344013761
## 18		-0.642560311	-2.08374141	-0.63044342	-1.287281383
## 19		-1.327357155	-1.61668515	0.69182061	-0.433948474
## 20		0.491118338	-1.89354674	-0.43151842	1.148845112
## 21		-0.785935664	0.01652167	0.63400281	-0.064878505
## 22		-0.673122025	-1.22881317	-0.20366993	-1.720566060
## 23		-0.445216904	-0.69734920	1.96035999	-0.122777823
## 24		0.675946569	0.33082119	0.19721004	1.525596069
## 25		-1.221961787	0.49361270	-0.10362775	-0.018866720
## 26		0.908309497	0.48841310	0.95448437	0.062422476
## 27		-0.360343894	-1.16827373	1.11890066	0.185019501
## 28		-0.520368190	-1.04727867	-0.64879357	1.335570898
## 29		-0.173876675	1.43489327	0.20603955	0.400678316
## 30		-0.130535682	0.10971223	-1.22779624	-0.777575490
## 31		0.233187104	-0.79152725	0.33175482	-0.684114760
## 32		-2.149497944	-1.26376986	-1.85148630	0.480397786
## 33		2.239733237	1.23874518	0.93266376	0.647426902
## 34		2.209112384	-0.48293835	-1.68028741	-1.636308824
## 35		2.768367726	-0.57520591	1.26191200	-0.898297516
## 36		3.600220924	0.43579137	0.34221560	1.509781011
## 37		0.566502898	0.69076574	-0.16404874	0.209131771
## 38		0.961275613	-0.47662717	-1.02506000	-1.791531945
## 39		1.216547586	-1.45519677	-0.02523689	0.008630304

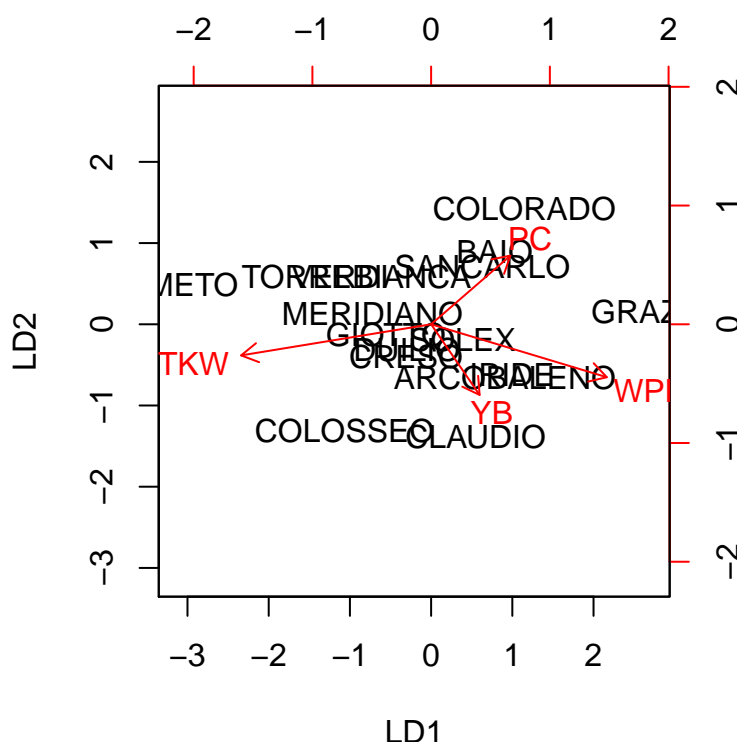
```
## 40  1.245865010 -1.23306378 -1.74895530  1.039196191
## 41  0.820337436  1.83606530 -0.33505589  0.026776247
## 42 -1.619702417  0.75851234 -0.09634714  0.412180227
## 43 -1.239882223 -0.80407986  0.56985685 -0.298762097
## 44 -0.853871231 -1.25917477 -0.61409870  1.139707257
## 45  0.980964869 -0.64434505  0.54229945  1.380265560
## 46  0.073861054  1.52784740  1.73922235 -0.032760932
## 47  1.300593410  1.71419789 -0.87616067 -0.942296732
## 48  0.173739755  0.22851385  2.19692765 -0.460767090
## 49 -4.361464687  0.55613107  0.21418049 -0.190905979
## 50 -2.843463727  0.43399161  1.17028062 -1.312918138
## 51 -3.305806348 -0.18258318  0.81247078 -0.533900499
## 52 -1.972944509  1.16396885  0.14432742  1.288950991
## 53 -0.958548674  0.12205725  0.83616256  0.197656186
## 54  2.099420550 -0.10600435 -1.10095866 -0.899847900
## 55  0.381474017 -0.82852244  1.17650610 -0.347058493
## 56  0.018438101  0.07058706 -0.49433443  1.157871773
## 57 -1.905224275  1.30165432  0.58062557 -0.325129341
## 58  0.704444138  0.80873591 -0.60555754 -1.334544513
## 59 -1.513273370  0.09322248  1.19347623 -0.842976907
## 60 -0.952868063  0.16310523 -0.24451245  0.982915023
## 61 -2.563767080  0.45458014 -1.22610987  0.192770222
## 62 -1.809022268  2.81536839 -2.72445356 -0.823495613
## 63  0.563300847 -0.15546830  0.34514241  0.686425459
## 64 -0.672341319 -0.76280237 -2.15649295  1.072761489
```

```
#Scores for centroids
```

```
cva1.centri <- cva1$means%*%cva1$scaling
```

```
#biplot
```

```
biplot(cva1.centri[,1:2], cva1$scaling[,1:2])
```

5 Altre letture

1. Crossa, J., 1990. Advances in Agronomy 44, 55-85.
2. NIST/SEMATECH, 2004. In "e-Handbook of Statistical Methods". NIST/SEMATECH, <http://www.itl.nist.gov/div898/handbook/>.
3. Sadocchi S., 1981. Manuale di analisi statistica multivariata per le scienze sociali. Franco Angeli Editore, Milano, 274 pp.
4. Moeller F., 1995. Analisi multivariata lineare. Franco Angeli, Milano, pp. 358.
5. Manly F.J., 1986. Multivariate statistical methods: a primer. Chapman & Hall, London, pp. 159.
6. Adugna W. e Labuschagne M. T., 2003. Cluster and canonical variate analyses in multilocation trials of linseed. Journal of Agricultural Science (140), 297-304.
7. Barberi P., Silvestri N. e Bonari E., 1997. Weed communities of winter wheat as influenced by input level and rotation. Weed Research 37, 301-313.
8. Casini P. e Proietti C., 2002. Morphological characterisation and production of Quinoa genotypes (*Chenopodium quinoa* Willd.) in the Mediterranean environment. Agricoltura Mediterranea 132, 15-26.

9. Onofri A. e Ciriciofolo E., 2004. Characterisation of yield quality in durum wheat by canonical variate analysis. Proceedings VIII ESA Congress "European Agriculture in a global context", Copenhagen, 11-15 July 2004, 541-542.
10. Shrestha A., Knezevic S. Z., Roy R. C., Ball-Cohelo B. R. e Swanton C. J., 2002. Effect of tillage, cover crop and crop rotation on the composition of weed flora in a sandy soil. Weed Research 42 (1), 76-87.
11. Streit B., Rieger S. B., Stamp P. e Richner W., 2003. Weed population in winter wheat as affected by crop sequence, intensity of tillage and time of herbicide application in a cool and humid climate. Weed Research 43, 20-32.