典型相關分析
Canonical Correlation Analysis

吳漢銘

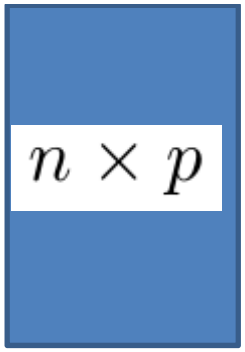
國立臺北大學 統計學系

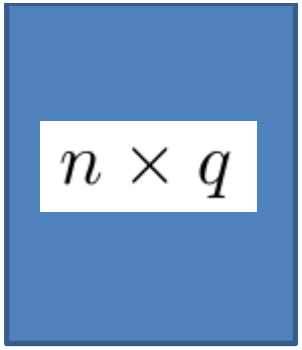
http://www.hmwu.idv.tw

Canonical correlation analysis

- The main purpose of CCA (Hotelling, 1936) is the exploration of sample correlations between two sets of quantitative variables observed on the same experimental units.

$$X = [X_1, X_2, \cdots, X_p]^T \qquad Y = [Y_1, Y_2, \cdots, Y_q]^T$$





assumed that $p \leq q$

assumed that the columns of X and Y are standardized

Classical CCA assumes first $p \leq n$ and $q \leq n$

$$\begin{aligned}
U_1 = \mathbf{a}_1^T X &= a_{11}X_1 + a_{12}X_2 + \cdots + a_{1p}X_p & V_1 = \mathbf{b}_1^T Y &= b_{11}Y_1 + b_{12}Y_2 + \cdots + b_{1q}Y_q \\
U_2 = \mathbf{a}_2^T X &= a_{21}X_1 + a_{22}X_2 + \cdots + a_{2p}X_p & V_2 = \mathbf{b}_2^T Y &= b_{21}Y_1 + b_{22}Y_2 + \cdots + b_{2q}Y_q \\
&\vdots \quad \vdots & &\vdots \quad \vdots \\
U_p = \mathbf{a}_p^T X &= a_{p1}X_1 + a_{p2}X_2 + \cdots + a_{pp}X_p & V_q = \mathbf{b}_q^T Y &= b_{q1}Y_1 + b_{q2}Y_2 + \cdots + b_{qq}Y_q
\end{aligned}$$

$$\rho_1 = corr(U_1, V_1) = \max corr(\mathbf{a}_1^T X, \mathbf{b}_1^T Y) \text{ subject to } Var(\mathbf{a}_1^T X) = Var(\mathbf{b}_1^T Y) = 1.$$

Mathematical aspects

$$Corr(U_i, V_i) = \frac{Cov(U_i, V_i)}{\sqrt{Var(U_i)}\sqrt{VarV_i}}, \quad i = 1, 2, \cdots, r = \min(p, q).$$

The first pair canonical variables is defined by $Corr(U_1, V_1) = \rho_1$.

- $\rho_1^2 \geq \rho_2^2 \geq \cdots \geq \rho_r^2$: the eigenvalues of the matrix $\Sigma_{XX}^{-1/2}\Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{XY}^T\Sigma_{XX}^{-1/2}$.
- Σ_{XX}: the variance-covariance of X.
- Σ_{YY}: the variance-covariance of Y.
- Σ_{XY}: the covariance matrix of the random vector X and Y.
- ρ_1: the first canonical correlation (the square root of the largest eigenvalue).

$$\frac{\mathbf{a}'\mathbf{\Sigma}_{XY}\mathbf{b}}{\sqrt{\mathbf{a}'\mathbf{\Sigma}_X\mathbf{a}}\sqrt{\mathbf{b}'\mathbf{\Sigma}_Y\mathbf{b}}}$$

The second pair canonical variable is $Corr(U_2, V_2) = \rho_2$ and so on.

$$\rho_1 = corr(U_1, V_1) = \max corr(\mathbf{a}_1^T X, \mathbf{b}_1^T Y) \text{ subject to } Var(\mathbf{a}_1^T X) = Var(\mathbf{b}_1^T Y) = 1.$$

Mathematical aspects

$$\Sigma_{XX} = \operatorname{cov}(X, X) \qquad \Sigma_{YY} = \operatorname{cov}(Y, Y)$$

maximize $$\rho = \frac{a' \Sigma_{XY} b}{\sqrt{a' \Sigma_{XX} a} \sqrt{b' \Sigma_{YY} b}}.$$

define $$c = \Sigma_{XX}^{1/2} a,$$

$$d = \Sigma_{YY}^{1/2} b.$$

$$\rho = \frac{c' \Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2} d}{\sqrt{c'c} \sqrt{d'd}}.$$

By the Cauchy–Schwarz inequality, $\left|\langle \mathbf{u}, \mathbf{v} \rangle\right|^2 \le \langle \mathbf{u}, \mathbf{u} \rangle \cdot \langle \mathbf{v}, \mathbf{v} \rangle,$

$$\left(c' \Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2}\right) d \le \left(c' \Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1/2} \Sigma_{YY}^{-1/2} \Sigma_{YX} \Sigma_{XX}^{-1/2} c\right)^{1/2} (d'd)^{1/2},$$

$$\rho \le \frac{\left(c' \Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} \Sigma_{XX}^{-1/2} c\right)^{1/2}}{(c'c)^{1/2}}.$$

https://en.wikipedia.org/wiki/Canonical_correlation

Mathematical aspects

$$\rho \leq \frac{\left(c' \Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} \Sigma_{XX}^{-1/2} c\right)^{1/2}}{(c'c)^{1/2}}.$$

There is equality if the vectors d and $\Sigma_{YY}^{-1/2} \Sigma_{YX} \Sigma_{XX}^{-1/2} c$ are collinear.

c is an eigenvector of $\Sigma_{XX}^{-1/2} \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX} \Sigma_{XX}^{-1/2}$

d is an eigenvector of $\Sigma_{YY}^{-1/2} \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY} \Sigma_{YY}^{-1/2}$

- a is an eigenvector of $\Sigma_{XX}^{-1} \Sigma_{XY} \Sigma_{YY}^{-1} \Sigma_{YX}$
- b is an eigenvector of $\Sigma_{YY}^{-1} \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY}$

$$c = \Sigma_{XX}^{1/2} a,$$

$$d = \Sigma_{YY}^{1/2} b.$$

The canonical variables are defined by:

$$U = c' \Sigma_{XX}^{-1/2} X = a' X$$

$$V = d' \Sigma_{YY}^{-1/2} Y = b' Y$$

https://en.wikipedia.org/wiki/Canonical_correlation

LifeCycleSavings {datasets}:
Intercountry Life-Cycle Savings Ratio Data

- Under the life-cycle savings hypothesis as developed by Franco Modigliani, the savings ratio (aggregate personal saving divided by disposable income (可支配收入)) is explained by per-capita disposable income (平均每人可支配所得), the percentage rate of change in per-capita disposable income, and two demographic variables: the percentage of population less than 15 years old and the percentage of the population over 75 years old. The data are averaged over the decade 1960–1970 to remove the business cycle or other short-term fluctuations.

A data frame with 50 observations on 5 variables.
[,1] sr : aggregate personal savings
[,2] pop15: % of population under 15
[,3] pop75: % of population over 75
[,4] dpi: real per-capita disposable income
[,5] ddpi: % growth rate of dpi

```
> head(LifeCycleSavings)
             sr pop15 pop75     dpi ddpi
Australia 11.43 29.35  2.87 2329.68 2.87
Austria   12.07 23.32  4.41 1507.99 3.93
Belgium   13.17 23.80  4.43 2108.47 3.82
Bolivia    5.75 41.89  1.67  189.13 0.22
Brazil    12.88 42.19  0.83  728.47 4.56
Canada     8.79 31.72  2.85 2982.88 2.43
> pop <- LifeCycleSavings[, 2:3]
> oec <- LifeCycleSavings[, -(2:3)]
```

```
> cancor(pop, oec)
$cor
[1] 0.8247966112 0.3652761515

$xcoef
                [,1]           [,2]
pop15 -0.009110856229 -0.03622206049
pop75  0.048647513750 -0.26031158157

$ycoef
            [,1]           [,2]           [,3]
sr   0.00847102  0.0333793558 -0.00515712977
dpi  0.00013073 -0.0000758823  0.00000454370
ddpi 0.00417059 -0.0122678964  0.05188323606

$xcenter
  pop15   pop75
35.0896  2.2930

$ycenter
       sr       dpi      ddpi
   9.6710 1106.7584    3.7576
```

Regularized CCA

- CCA cannot be performed when the $n \leq \max(p, q)$).
- When the number of variables increases, greatest canonical correlations are nearly 1 because of recovering of canonical subspaces that do not provide any meaningful information.
- A standard condition usually advocated for CCA (Eaton and Perlman 1973) is $n \geq p + q + 1$.
- A regularization step in the data processing (Bickel and Li 2006) to perform a regularized canonical correlation analysis (RCCA).

S_{XX} and S_{YY} are replaced respectively by $\Sigma_{XX}(\lambda_1)$ and $\Sigma_{YY}(\lambda_2)$ defined by

$$\Sigma_{XX}(\lambda_1) = S_{XX} + \lambda_1 I_p \quad \text{and} \quad \Sigma_{YY}(\lambda_2) = S_{YY} + \lambda_2 I_q.$$

How to get the "good" values for the regularization parameters?

Bickel PJ, Li B (2006). "Regularization in Statistics." Sociedad de Estadistica e Investigacion Operativa, Test, 15(2), 271–344.

Cross-validation for tuning regularization parameters

- Denote $\alpha = (\alpha_1, \alpha_2)$. For a given value of α, do $i = 1, \cdots, n$:
  - $\rho_\alpha^{(-i)}$: the first canonical correlation computed from the units with rows X^i and Y^i removed.
  - $\mathbf{a}_\alpha^{(-i)}$ and $\mathbf{b}_\alpha^{(-i)}$: the first canonical variates vectors.

$$\Sigma_{XX}(\alpha_1) = S_{XX} + \alpha_1 \mathbf{I}_p$$

$$\Sigma_{YY}(\alpha_2) = S_{YY} + \alpha_2 \mathbf{I}_q$$

- Obtain n pairs of vectors

$$(\mathbf{a}_\alpha^{(-1)}, \mathbf{b}_\alpha^{(-1)}), \cdots, (\mathbf{a}_\alpha^{(-n)}, \mathbf{b}_\alpha^{(-n)}).$$

- The leave-one-out cross validation score for $\alpha = (\alpha_1, \alpha_2)$ is then defined by Leurgans et al. (1993):

$$CV(\alpha_1, \alpha_2) = corr(\{(\mathbf{a}_\alpha^{(-i)})^T X^i\}_{i=1}^n, \{(\mathbf{b}_\alpha^{(-i)})^T Y^i\}_{i=1}^n).$$

- Choose the value of α_1 and α_2 that maximizes this correlation:

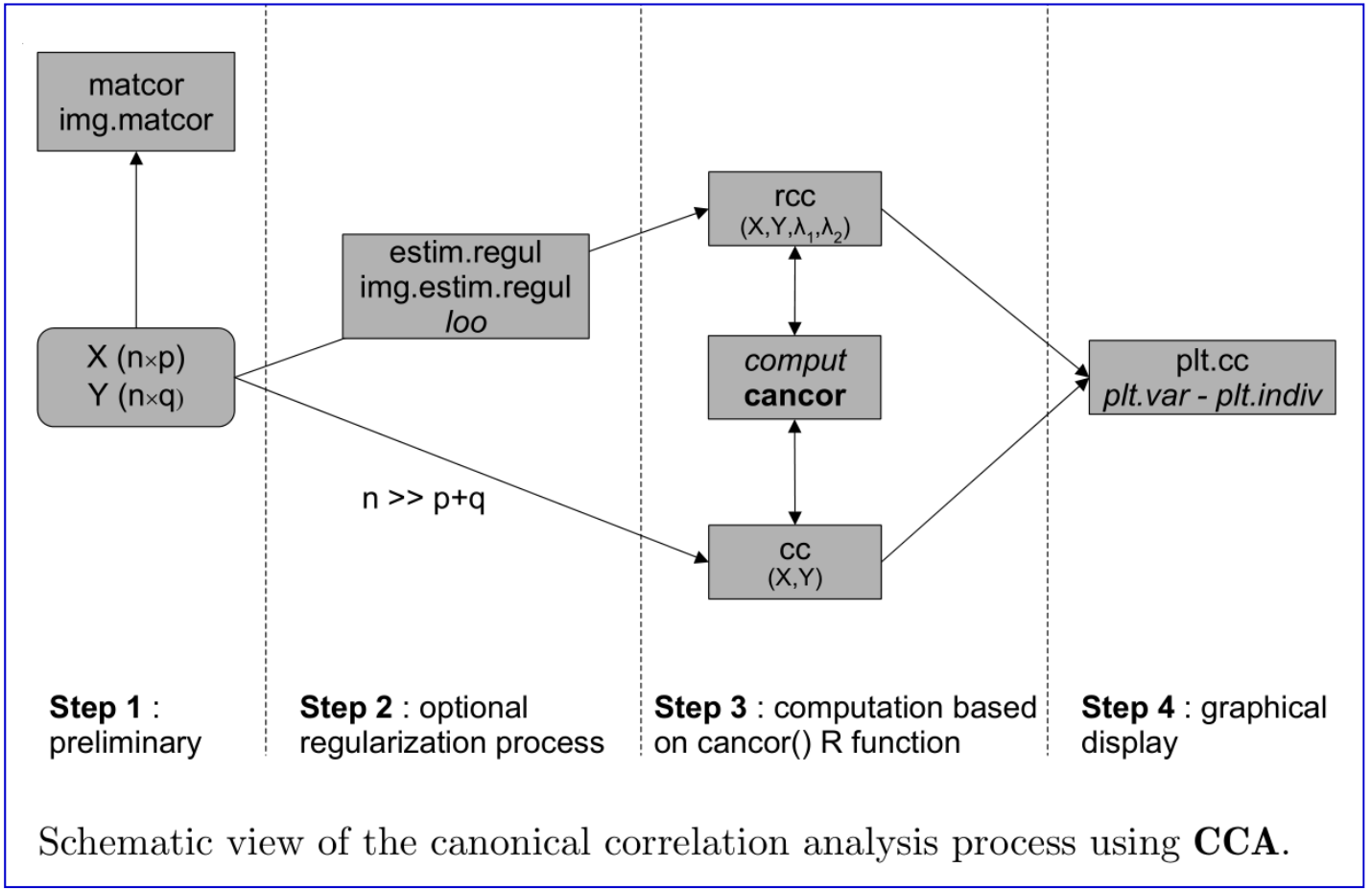
$$\hat{\alpha} = (\hat{\alpha_1}, \hat{\alpha_2}) = \arg\max_{\alpha_1, \alpha_2} CV(\alpha_1, \alpha_2).$$

Ignacio González, Sébastien Déjean, Pascal G. P. Martin, Alain Baccini, 2008, CCA: An R Package to Extend Canonical Correlation Analysis, Journal of Statistical Software, Vol 23, Issue 12.

- Note that $\hat{\alpha_1}$ and $\hat{\alpha_2}$ are chosen with respect to the first canonical variates and are then fixed for higher order canonical variates.

Two tuning parameters in the regularized CCA,
so the CV is performed on a 2D surface.
(1) Directly searching for a maximum on the 2D parameter surface. or
(2) A relatively small grid of reasonable values for α_1 and α_2.

CCA: An R Package to Extend Canonical Correlation Analysis



Schematic view of the canonical correlation analysis process using **CCA**.

Ignacio González, Sébastien Déjean, Pascal G. P. Martin, Alain Baccini, 2008, CCA: An R Package to Extend Canonical Correlation Analysis, Journal of Statistical Software, Vol 23, Issue 12.

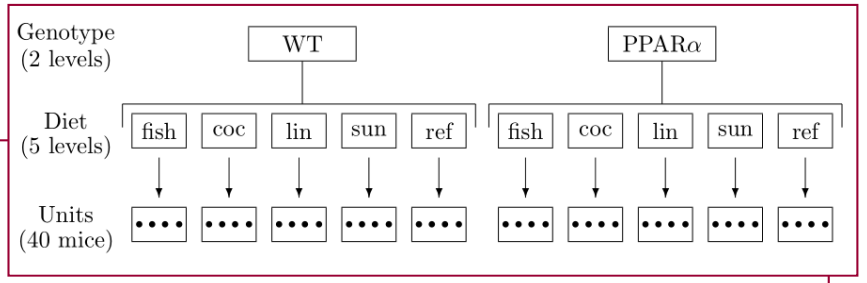
nutrimouse {CCA}: a nutrition study in the mouse

Two sets of variables were measured on 40 mice:

- gene (40 x 120): expressions of 120 genes measured in liver cells potentially involved in nutritional problems.
- lipid (40 x 21): concentrations of 21 hepatic fatty acids (肝脂肪酸).

The 40 mice were distributed in a 2-factors experimental design (4 replicates):

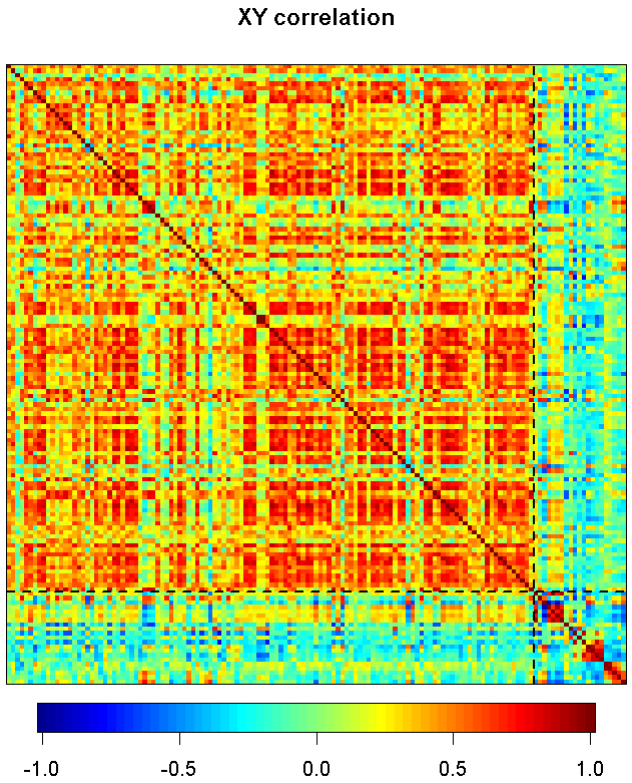
- genotype (2-levels factor): wild-type and PPARalpha -/-
- diet (5-levels factor): Oils used

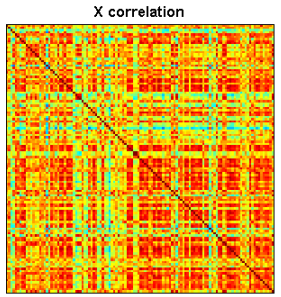


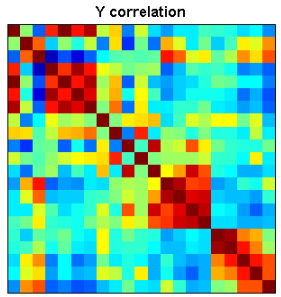

```
> library(CCA)
> data(nutrimouse)
> str(nutrimouse)
List of 4
 $ gene    :'data.frame':        40 obs. of  120 variables:
  ..$ X36b4    : num [1:40] -0.42 -0.44 -0.48 -0.45 -0.42 -0.43 -0.53 -0.49 -0.36 -0.5 ...
...
  ..$ Tpbeta   : num [1:40] -1.11 -1.09 -1.14 -1.04 -1.2 -1.05 -1 -1.16 -0.91 -1.07 ...
  .. [list output truncated]
 $ lipid   :'data.frame':        40 obs. of  21 variables:
  ..$ C14.0    : num [1:40] 0.34 0.38 0.36 0.22 0.37 1.7 0.35 0.34 0.22 1.38 ...
...
  ..$ C22.6n.3: num [1:40] 10.39 2.61 2.51 14.99 6.69 ...
 $ diet    : Factor w/ 5 levels "coc","fish","lin",..: 3 5 5 2 4 1 3 3 2 1 ...
 $ genotype: Factor w/ 2 levels "wt","ppar": 1 1 1 1 1 1 1 1 1 1 ...
>
> table(nutrimouse$genotype, nutrimouse$diet)

      coc fish lin ref sun
  wt    4    4   4   4   4
  ppar  4    4   4   4   4
```

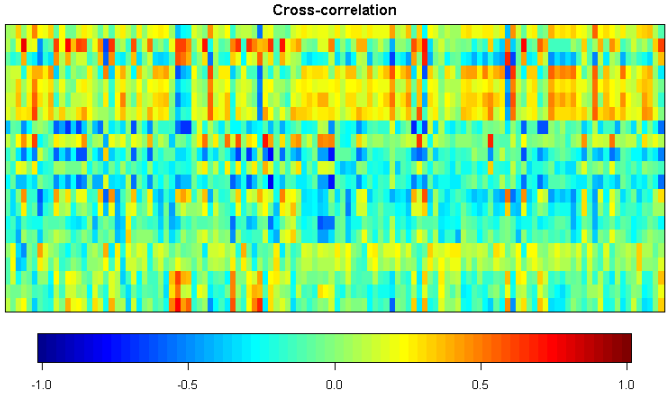
Visualizing the correlation matrices

```
> # convert the data into the matrix format before performing CCA.
> X <- as.matrix(nutrimouse$gene)
> Y <- as.matrix(nutrimouse$lipid)
> correl <- matcor(X, Y)
> img.matcor(correl)
> img.matcor(correl, type = 2)
> str(correl)
List of 3
 $ Xcor : num [1:120, 1:120] 1 0.328 0.107 0.262 0.491 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:120]
  .. ..$ : chr [1:120]
 $ Ycor : num [1:21, 1
  ..- attr(*, "dimname
  .. ..$ : chr [1:21]
  .. ..$ : chr [1:21]
 $ XYcor: num [1:141,
  ..- attr(*, "dimname
  .. ..$ : chr [1:141]
  .. ..$ : chr [1:141]
```



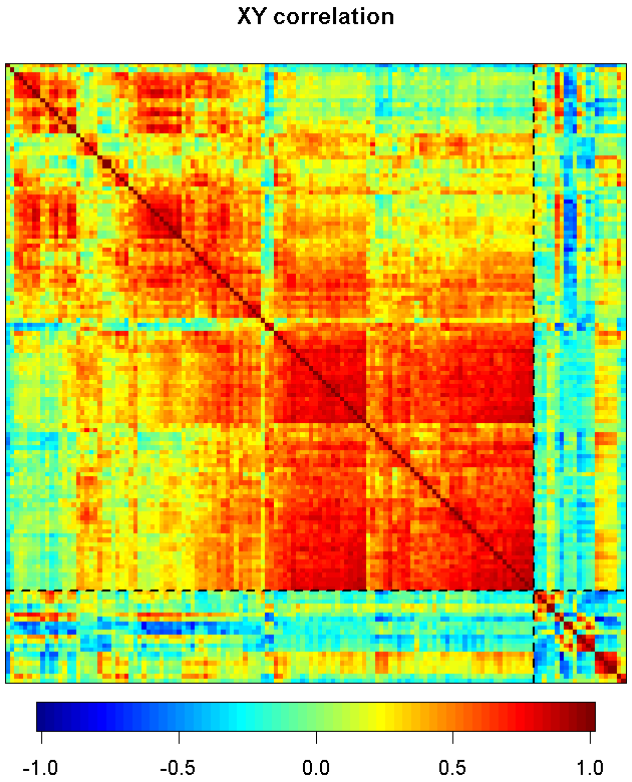


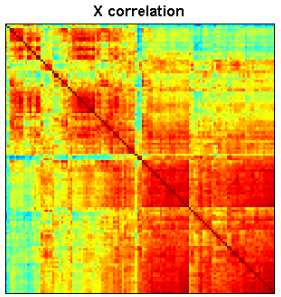


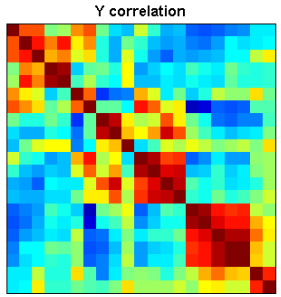


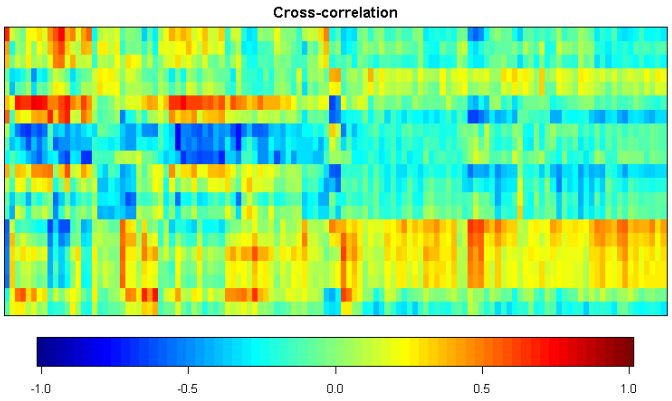
Visualizing the correlation matrices

```
> x.hm <- heatmap(correl$Xcor)
> y.hm <- heatmap(correl$Ycor)
> ordering <- c(x.hm$rowInd, y.hm$rowInd + 120)
> correl.2 <- list(Xcor=correl$Xcor[x.hm$rowInd, x.hm$rowInd],
+                  Ycor=correl$Ycor[y.hm$rowInd, y.hm$rowInd],
+                  XYcor=correl$XYcor[ordering, ordering])
> img.matcor(correl.2)
> img.matcor(correl.2, type = 2)
```



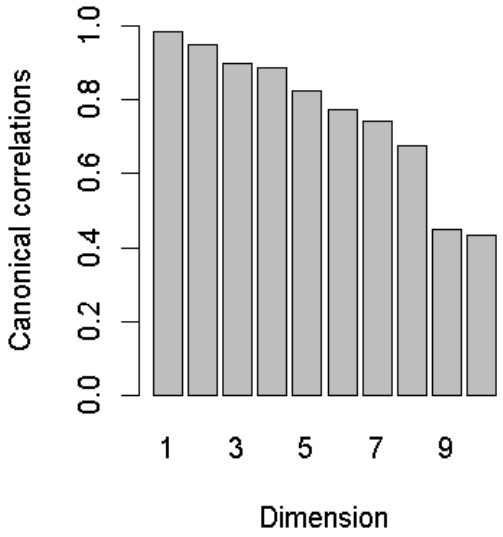


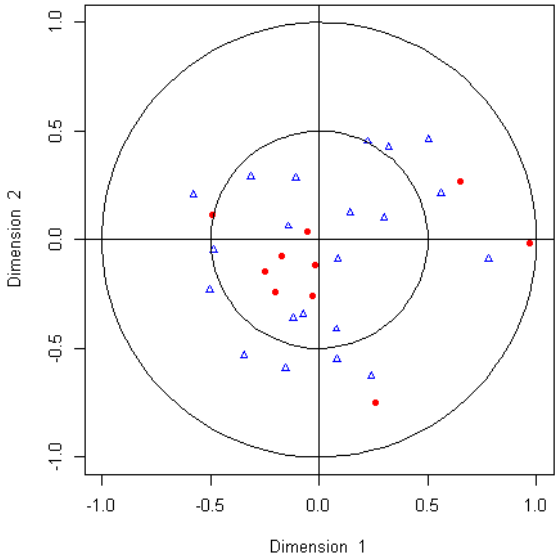


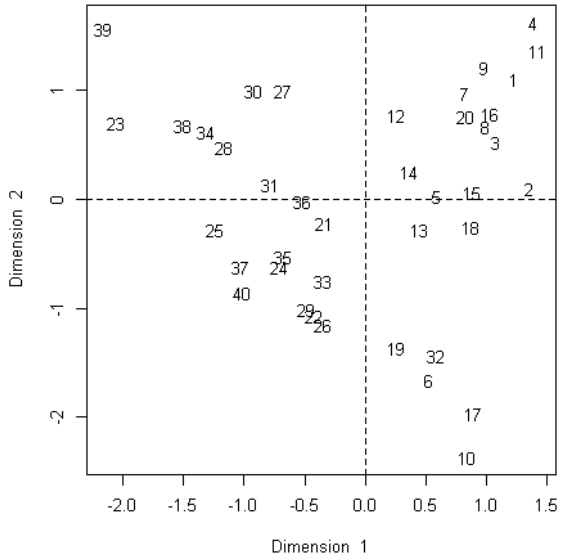


Classical CCA

```
> X.subset <- as.matrix(nutrimouse$gene[, sample(1:120, size = 10)])
> my.cca <- cc(X.subset, Y)
> barplot(my.cca$cor, xlab="Dimension",
+         ylab="Canonical correlations",
+         names.arg=1:10, ylim=c(0,1))
> plt.cc(my.cca)
> names(my.cca)
[1] "cor"    "names"  "xcoef"  "ycoef"  "scores"
> my.cca$cor
 [1] 0.9851407 0.9494411 0.8996597 0.8882256 0.8228657 0.7720856 0.7430616 0.6739105 0.4497435
[10] 0.4339681
```





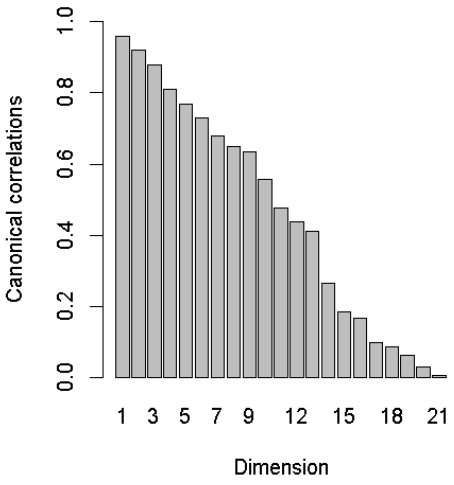


Regularized CCA

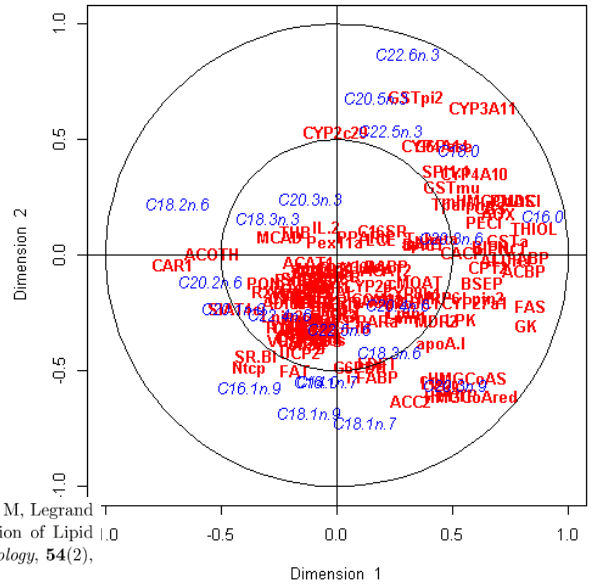
```
> regul.par <- estim.regul(X, Y, plt = TRUE,
+                          grid1 = seq(0.0001, 0.01, l=5), # l=50
+                          grid2 = seq(0, 0.1, l=5)) # l=50
  lambda1 =  0.01
  lambda2 =  0.075
 CV-score =  0.884716
> my.rcca <- rcc(X, Y, regul.par$lambda1, regul.par$lambda2)
> names(my.rcca)
[1] "cor"    "names"  "xcoef"  "ycoef"  "scores"
> barplot(my.rcca$cor, xlab = "Dimension",
+         ylab = "Canonical correlations", names.arg = 1:21, ylim = c(0,1))
> plt.cc(my.rcca, var.label = TRUE,
+        ind.names = paste(nutrimouse$genotype, nutrimouse$diet, sep = "."))
```

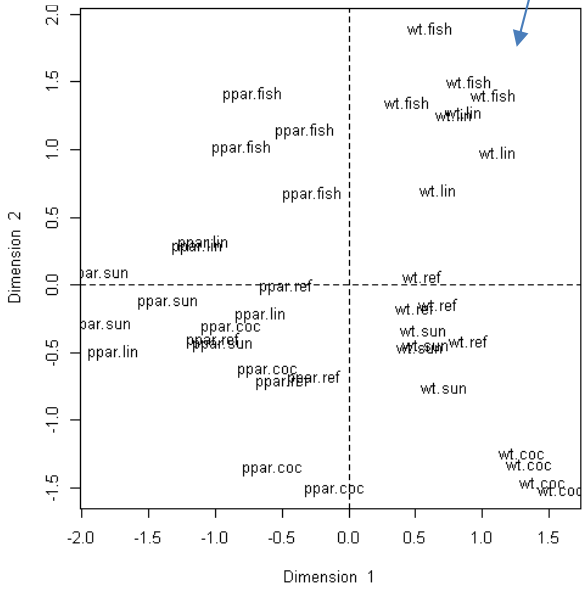
```
> names(my.rcca$scores)
[1] "xscores"       "yscores"       "corr.X.xscores"
[4] "corr.Y.xscores" "corr.X.yscores" "corr.Y.yscores"
```

my.rcca$scores$xscores[,1:2]



Variables (on the left) and units (on the right) representations on the plane defined by the first two canonical variates.





Martin PG, Guillou H, Lasserre F, Déjean S, Lan A, Pascussi JM, SanCristobal M, Legrand P, Besse P, Pineau T (2007). "Novel Aspects of PPARα-mediated Regulation of Lipid and Xenobiotic Metabolism Revealed through a Nutrigenomic Study." *Hepatology*, **54**(2), 767–777.

plt.cc

```
> plt.cc
function (res, d1 = 1, d2 = 2, int = 0.5, type = "b", ind.names = NULL,
    var.label = FALSE, Xnames = NULL, Ynames = NULL)
{
    par(mfrow = c(1, 1), pty = "s")
    if (type == "v")
        plt.var(res, d1, d2, int, var.label, Xnames, Ynames)
    if (type == "i")
        plt.indiv(res, d1, d2, ind.names)
    if (type == "b") {
        def.par <- par(no.readonly = TRUE)
        layout(matrix(c(0, 0, 1, 2, 0, 0), ncol = 2, nrow = 3,
            byrow = TRUE), widths = 1, heights = c(0.1, 1, 0.1))
        par(pty = "s", mar = c(4, 4.5, 0, 1))
        plt.var(res, d1, d2, int, var.label, Xnames, Ynames)
        plt.indiv(res, d1, d2, ind.names)
        par(def.par)
    }
}
<environment: namespace:CCA>
```

```
> plt.var
function (res, d1, d2, int = 0.5, var.label = FALSE, Xnames = NULL,
    Ynames = NULL)
{
    if (!var.label) {
        plot(0, type = "n", xlim = c(-1, 1), ylim = c(-1, 1),
            xlab = paste("Dimension ", d1), ylab = paste("Dimension ",
                d2))
        points(res$scores$corr.X.xscores[, d1], res$scores$corr.X.xscores[,
            d2], pch = 20, cex = 1.2, col = "red")
        points(res$scores$corr.Y.xscores[, d1], res$scores$corr.Y.xscores[,
            d2], pch = 24, cex = 0.7, col = "blue")
    }
    else {
        if (is.null(Xnames))
            Xnames = res$names$Xnames
        if (is.null(Ynames))
            Ynames = res$names$Ynames
        plot(0, type = "n", xlim = c(-1, 1), ylim = c(-1, 1),
            xlab = paste("Dimension ", d1), ylab = paste("Dimension ",
                d2))
        text(res$scores$corr.X.xscores[, d1], res$scores$corr.X.xscores[,
            d2], Xnames, col = "red", font = 2)
        text(res$scores$corr.Y.xscores[, d1], res$scores$corr.Y.xscores[,
            d2], Ynames, col = "blue", font = 3)
    }
    abline(v = 0, h = 0)
    lines(cos(seq(0, 2 * pi, l = 100)), sin(seq(0, 2 * pi, l = 100)))
    lines(int * cos(seq(0, 2 * pi, l = 100)), int * sin(seq(0,
        2 * pi, l = 100)))
}
<environment: namespace:CCA>
```

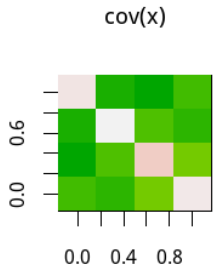
```
> plt.indiv
function (res, d1, d2, ind.names = NULL)
{
    if (is.null(ind.names))
        ind.names = res$names$ind.names
    if (is.null(ind.names))
        ind.names = 1:nrow(res$scores$xscores)
    plot(res$scores$xscores[, d1], res$scores$xscores[, d2],
        type = "n", main = "", xlab = paste("Dimension ", d1),
        ylab = paste("Dimension ", d2))
    text(res$scores$xscores[, d1], res$scores$xscores[, d2],
        ind.names)
    abline(v = 0, h = 0, lty = 2)
}
<environment: namespace:CCA>
```

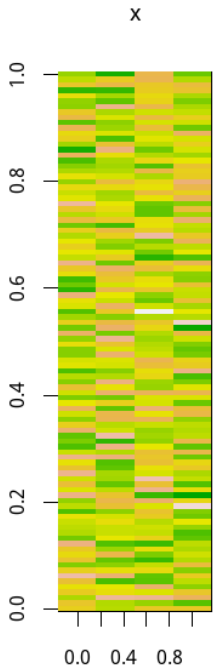
Check Data and its Covariance

```
> library("corpcor")
> options(digits=4)
>
> test <- function(x, s){
+   image(t(s)[,nrow(s):1], main="cov(x)", col=terrain.colors(100))
+   image(t(x)[,nrow(x):1], main="x", col=terrain.colors(100))
+   cat("is.positive.definite:", is.positive.definite(s), "\n")
+   cat("eigen:\n")
+   print(eigen(s))
+   cat("inverse:\n")
+   print(solve(s))
+ }
>
> layout(matrix(1:2, ncol=1), height=c(1,2))
> # set 1: regular data
> n <- 100
> p <- 4
> x1 <- matrix(rnorm(n*p), ncol=p)
> summary(x1)
       V1                 V2                 V3                 V4
 Min.   :-2.4964   Min.   :-2.5184   Min.   :-2.182   Min.   :-2.9044
 1st Qu.:-0.7516   1st Qu.:-0.5753   1st Qu.:-0.393   1st Qu.:-0.7751
 Median : 0.0941   Median : 0.0726   Median : 0.221   Median :-0.0412
 Mean   : 0.0382   Mean   : 0.1075   Mean   : 0.167   Mean   : 0.0181
 3rd Qu.: 0.6850   3rd Qu.: 0.9046   3rd Qu.: 0.770   3rd Qu.: 0.7272
 Max.   : 2.3993   Max.   : 2.3687   Max.   : 3.369   Max.   : 3.0475
```

Check Data and its Covariance

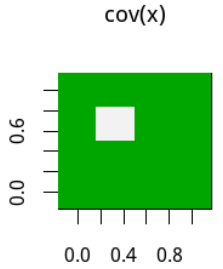
```
> s1 <- cov(x1)
> test(x1, s1)
is.positive.definite: TRUE
eigen:
$values
[1] 1.4157 1.2531 1.0118 0.7773

$vectors
        [,1]    [,2]    [,3]     [,4]
[1,]  0.6637  0.1010  0.4884  0.55746
[2,] -0.4384 -0.5896  0.6775  0.03517
[3,] -0.5707  0.3055 -0.1423  0.74879
[4,] -0.2040  0.7408  0.5313 -0.35681

inverse:
         [,1]     [,2]     [,3]     [,4]
[1,]  0.95483  0.09922  0.22535 -0.03536
[2,]  0.09922  0.86835 -0.02841  0.05419
[3,]  0.22535 -0.02841  1.04587 -0.15555
[4,] -0.03536  0.05419 -0.15555  0.91009
```

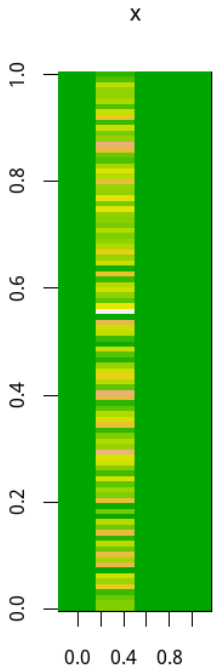




Check Data and its Covariance

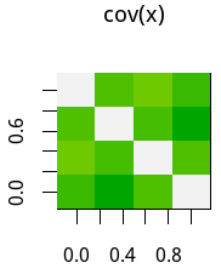
```
> # set 2: add some outlier variables
> x2 <- matrix(rnorm(n*p, sd=0.0001), ncol=p)
> id <- sample(1:p, floor(p/3))
> x2[, id] <- x2[, id] + abs(rnorm(n*length(id), m=10000, sd=5000))
> summary(x2)
       V1                    V2              V3                    V4
 Min.   :-0.00019325   Min.   :  318   Min.   :-0.0002198   Min.   :-0.00026808
 1st Qu.:-0.00004618   1st Qu.: 7273   1st Qu.:-0.0000667   1st Qu.:-0.00007213
 Median :-0.00000283   Median : 9900   Median : 0.0000107   Median : 0.00000007
 Mean   : 0.00000367   Mean   :10201   Mean   : 0.0000103   Mean   : 0.00000063
 3rd Qu.: 0.00005736   3rd Qu.:12617   3rd Qu.: 0.0000998   3rd Qu.: 0.00007487
 Max.   : 0.00021909   Max.   :27439   Max.   : 0.0003177   Max.   : 0.00027794
> s2 <- cov(x2)
> test(x2, s2)
is.positive.definite: FALSE
eigen:
$values
[1] 25301262.378715548664        0.000000012360        0.000000010255        0.00

$vectors
                [,1]             [,2]             [,3]             [,4]
[1,] -0.0000000006300 -0.2198484423160 -0.123788454460  0.9676482216971
[2,] -1.0000000000000 -0.0000000004066  0.000000004536 -0.0000000001631
[3,] -0.0000000002447 -0.9600601615329 -0.148515578779 -0.2371236156468
[4,]  0.0000000045072 -0.1730640015964  0.981131765566  0.0861934449310

inverse:
               [,1]           [,2]            [,3]          [,4]
[1,] 133713761.94777 -0.06915255563 -12572701.82036  2664291.0791
[2,]        -0.06915  0.00000004155        -0.02882        0.4378
[3,] -12572701.82036 -0.02881648352  84429162.38724 -3566776.2275
[4,]   2664291.07907  0.43778259944  -3566776.22748 97309029.4158
```

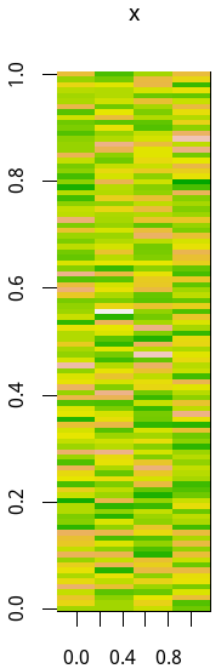




Check Data and its Covariance

```
> # set 3: do the scaling
> x3 <- scale(x2)
> s3 <- cov(x3)
> test(x3, s3)
is.positive.definite: TRUE
eigen:
$values
[1] 1.2270 1.1165 0.8797 0.7768

$vectors
         [,1]     [,2]     [,3]     [,4]
[1,]  0.21350  0.66557  0.71239  0.06273
[2,]  0.69471 -0.06814 -0.20495  0.68610
[3,]  0.04397  0.71997 -0.67057 -0.17333
[4,] -0.68547  0.18442 -0.02884  0.70377

inverse:
         [,1]     [,2]     [,3]     [,4]
[1,]  1.01590 -0.03032 -0.12020  0.02415
[2,] -0.03032  1.05120 -0.01590  0.22895
[3,] -0.12020 -0.01590  1.01571 -0.04068
[4,]  0.02415  0.22895 -0.04068  1.05192
```





Some Columns are Linear Dependent

```
> library(fields); library("corpcor"); library(CCA)
> cor.col <- two.colors(start="blue", middle="white",
                        end="red") # length=255
> par(mfrow=c(3,1))
> n <- 100
> p <- 4
> q <- 5
> set.seed(12345)
> X <- matrix(rnorm(n*p), ncol=p); rX <- cor(X)
> range.col <- floor((1+range(rX))*127+1)
> # wrong: image(t(cor(Y))[,q:1], main="cor(Y)", col=cor.col)
> image(t(rX)[,p:1], main="cor(X)", col=cor.col[range.col[1]: range.col[2]])
> is.positive.definite(cov(X))
[1] TRUE
> Y <- matrix(rnorm(n*q), ncol=q); rY <- cor(Y)
> range.col <- floor((1+range(rY))*127+1)
> image(t(rY)[,q:1], main="cor(Y)", col=cor.col[range.col[1]: range.col[2]])
> is.positive.definite(cov(Y))
[1] TRUE
> xy.cca <- cc(X, Y)
>
> X[,1] <- 3*X[,4] + 1; rX <- cor(X)
> range.col <- floor((1+range(rX))*127+1)
> image(t(rX)[,p:1], main="cor(X)", col=cor.col[range.col[1]: range.col[2]])
> xy.cca <- cc(X, Y)
Error in chol.default(Bmat) :
  the leading minor of order 4 is not positive definite
> is.positive.definite(cov(X))
[1] FALSE
```