Advanced Multivariate Statistical Methods for Metabolomic Data Analysis

Dabao Zhang, Min Zhang

June 24, 2018

Outline

- R and RStudio
- Principal Component Analysis (PCA)
- Partial Least Squares (PLS)
- Seemingly Unrelated Regression (SUR)
- Penalized Orthogonal-Components Regression (POCRE)
- Random Forest

R and RStudio

- R is a free software environment for statistical computing and graphics
  - An independent implementation of the S language
  - Available from `https://cran.r-project.org`
- RStudio provides a free and open-source integrated development environment (IDE) for R
  - Available to run on the desktop (Windows, Mac, and Linux) or in a browser connected to RStudio Server
  - Available from `https://www.rstudio.com/products/rstudio/`

R Packages

- Statistical tools available in R as packages
- Each package bundles together codes, data, and documentation to share
- There are over 10,000 R packages available in the Comprehensive R Archive Network (CRAN)
- Packages we are going to use today:
  - stat: `? "stats-package"`
  - pls: `? pcr` or `? plsr`
  - systemfit:
  - pocre: available from us
  - randomForest:

Data Structure in R

- Common: vectors of character, numeric, logical, factor; list, matrix, array
- Most popular data structure: `data.frame`
- Data frame is a list of variables of the same length

```
mbd <- read.table("metabdata.csv",header=T,sep=",")
head(mbd[,1:7]) # metabolites' abundance in mbd[,7:30]
```

```
##   Diagnosis Age Gender   BMI Smoking Alcohol   Formate
## 1    Polyps  48      M 22.00     Yes      No 10.284198
## 2    Polyps  50      M 32.80      No      No  2.190488
## 3    Polyps  53      F 23.34      No     Yes 12.954623
## 4    Polyps  53      F 22.30      No      No  4.533463
## 5    Polyps  55      M 24.50      No     Yes  8.096929
## 6    Polyps  55      M 33.00      No      No 12.643753
```

```
summary(mbd[,1:4])
```

```
##    Diagnosis        Age          Gender       BMI        
##  Healthy:58   Min.   :48.00   F:49   Min.   :18.30  
##  Polyps :44   1st Qu.:56.00   M:53   1st Qu.:24.25  
##               Median :61.00          Median :27.04  
##               Mean   :60.85          Mean   :28.40  
##               3rd Qu.:66.00          3rd Qu.:32.35  
##               Max.   :72.00          Max.   :47.93
```

```
summary(mbd[,5:8])
```

```
##  Smoking  Alcohol     Formate          Histidine       
##  No :55   No :30   Min.   : 1.017   Min.   :  5.985  
##  Yes:47   Yes:72   1st Qu.: 4.867   1st Qu.: 84.514  
##                    Median : 7.489   Median :105.250  
##                    Mean   : 8.279   Mean   :101.397  
##                    3rd Qu.:10.797   3rd Qu.:121.240
```

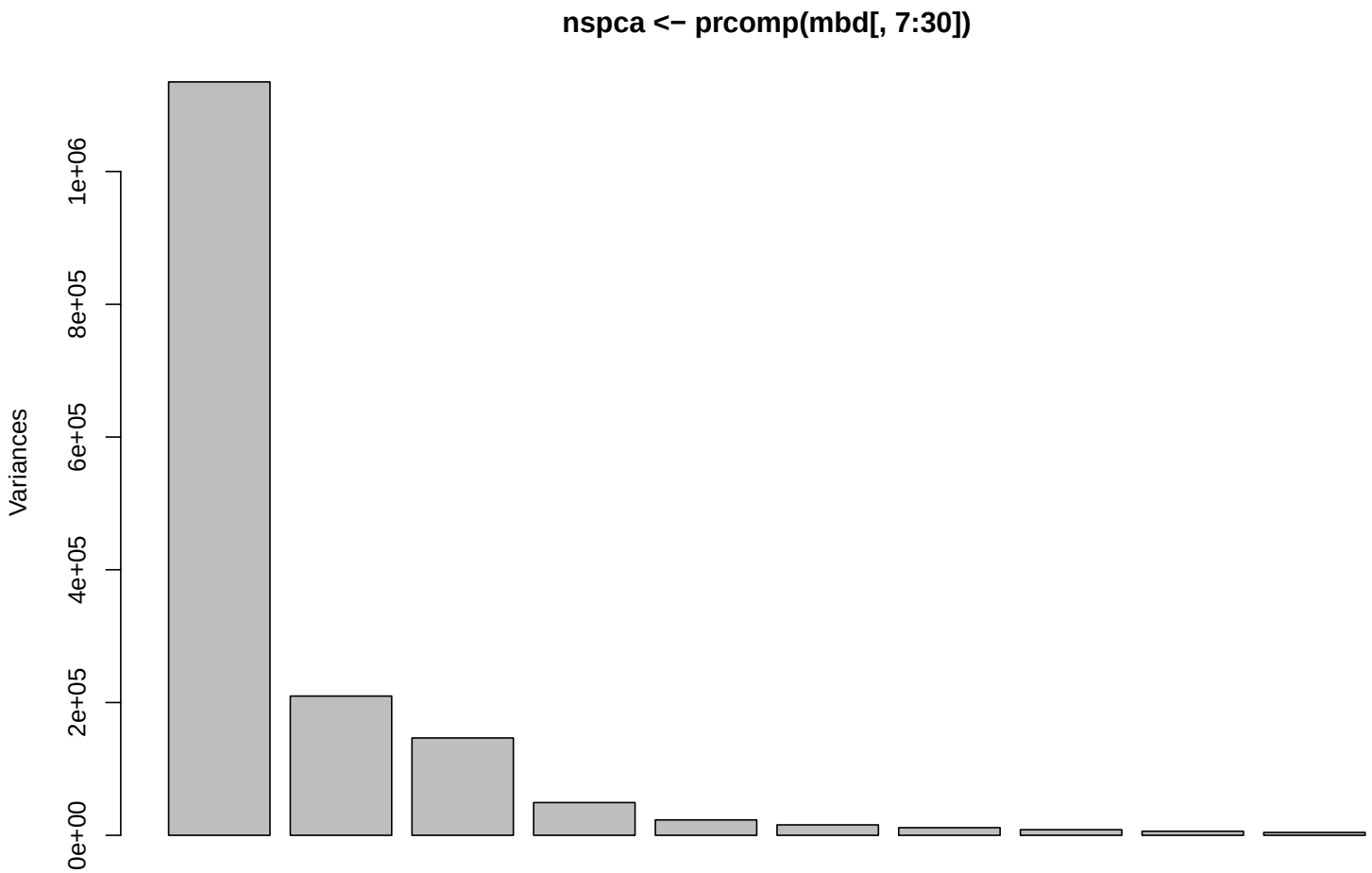
Principal Component Analysis (PCA)

- PCA is an unsupervised dimension reduction approach to construct principal components
  - First Principal Component: The direction which has the largest variation
  - Second Principal Component: The direction which has the second largest variation
  - ……
- Function in `statS` package: prcomp(x, retx=T, center=T, scale.=F)

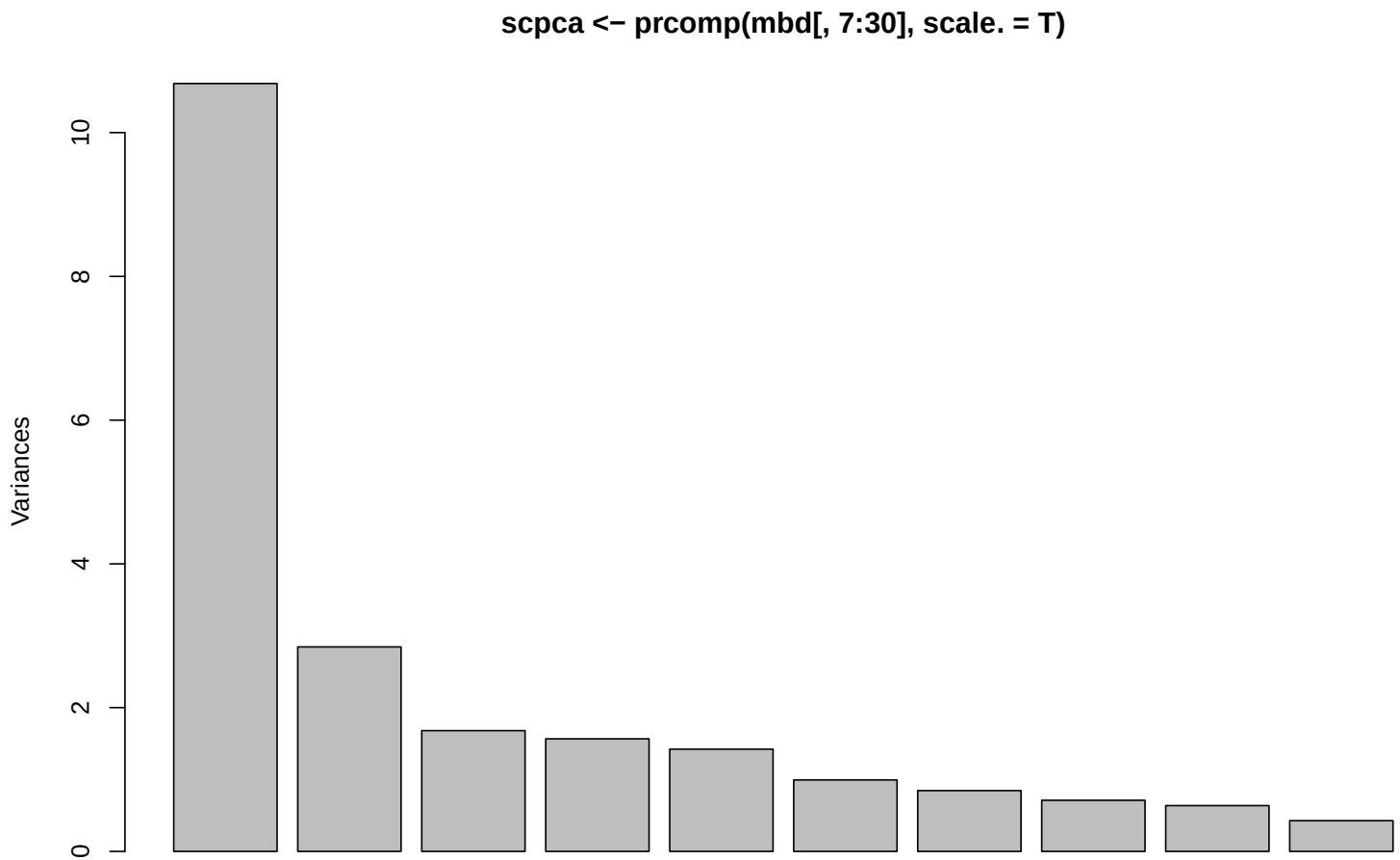
```
? prcomp
```

- As different variables may vary at significantly different scales, scaling is preferred, i.e., set `scale.=T`

```
plot(nspca<-prcomp(mbd[,7:30]))
```



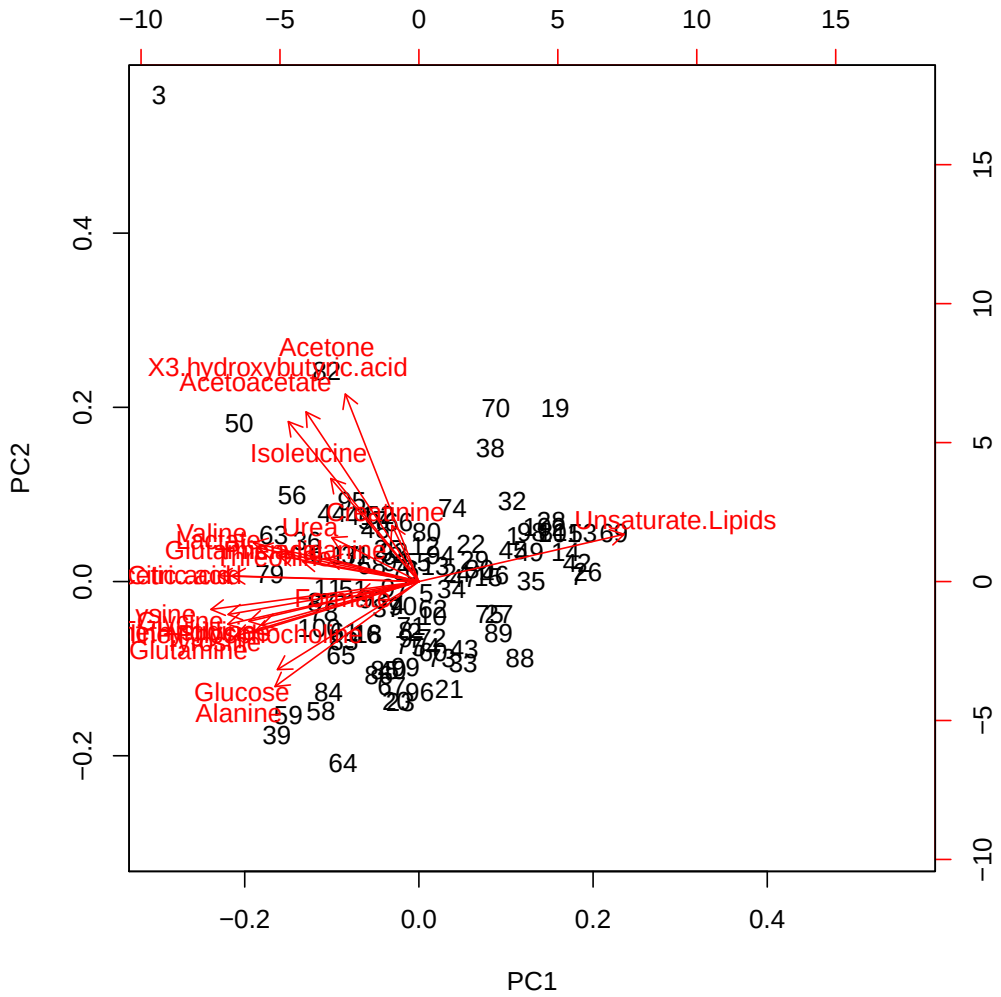
```
plot(scpca<-prcomp(mbd[,7:30],scale.=T))
```



- Checking with `summary(scpca)`, we observe that the first two PCs account for 44.51% and 11.86% of the total variation, respectively.
- The first five PCs account for over 75% of the total variation.
- We can choose the number of components based on how much of the total variation can be accounted for.
- Question: What do the PCs imply?

- There are culsters of metabolites shown in `biplot` of the first two PCs

```
biplot(scpca)
```



Principal Components Regression

- When some response variables like clinical traits are interested, we may regressing Y against principal components of predictors, instead of regressing directly against the original predictors
  - Avoid collinearity between predictors!
  - Avoid overfitting due to a large number of predictors!
- Function in `pls` package: pcr(y~x, scale=FALSE, validation=c("none", "CV", "LOO"))

```
? pcr
```

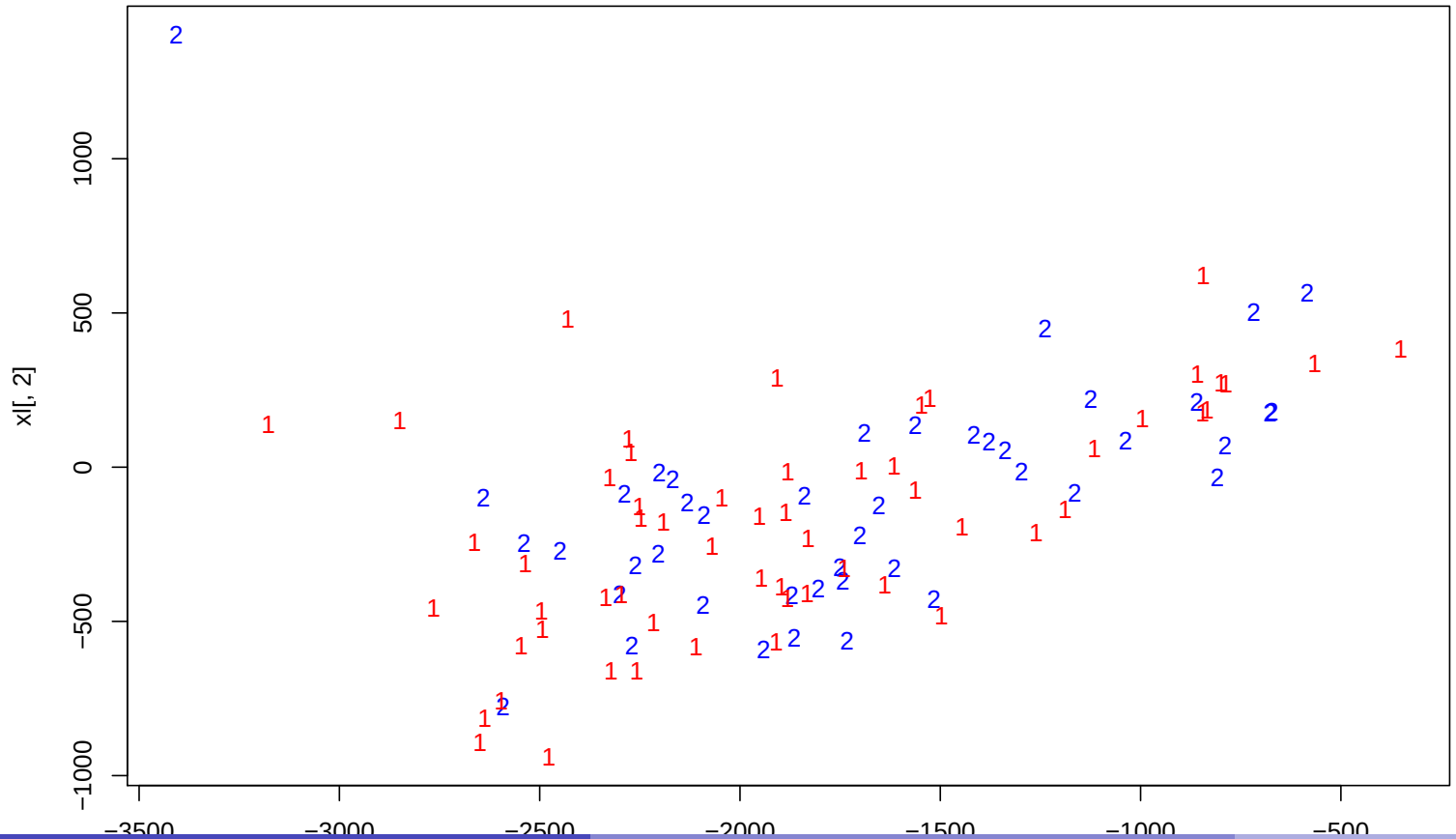
- As different variables may vary at significantly different scales, scaling is preferred, i.e., set `scale=T`

```r
require(pls,warn.conflicts=F,quietly=T)
idiag <- as.integer(mbd$Diagnosis) # 1~Healthy, 2~Polyps
lmpcr <- pcr(idiag~as.matrix(mbd[,7:30]),5)
summary(lmpcr)
```

```
## Data:    X dimension: 102 24
##  Y dimension: 102 1
## Fit method: svdpc
## Number of components considered: 5
## TRAINING: % variance explained
##         1 comps  2 comps  3 comps  4 comps  5 comps
## X        69.927   82.850   91.884   94.922   96.351
## idiag     1.931    1.942    2.125    2.216    2.753
```

- Although the first five PCs account for over 95% of the variation in metabolites, they only account for less than 3% of the total variation in the clinical trait

```r
xl <- as.matrix(mbd[,7:30])%*%scpca$rotation
idiag <- as.integer(mbd$Diagnosis) # 1~Healthy, 2~Polyps
plot(xl[,1],xl[,2],type = "n")
text(xl[,1],xl[,2],labels=idiag,col=c('red','blue')[idiag])
```



```r
bmipcr <- pcr(mbd$BMI~as.matrix(mbd[,7:30]),5)
summary(bmipcr)
```

```
## Data:    X dimension: 102 24
##  Y dimension: 102 1
## Fit method: svdpc
## Number of components considered: 5
## TRAINING: % variance explained
##           1 comps  2 comps  3 comps  4 comps  5 comps
## X           69.93    82.85    91.88    94.92    96.35
## mbd$BMI     23.71    24.77    24.87    24.88    25.09
```

- Although the first five PCs account for over 95% of the variation in metabolites, they only account for about 25% of the total variation in BMI
- Therefore, the leading principal components may not contribute significantly to explaining Y

Partial Least Squares (PLS)

- PLS is a supervised dimension reduction approach to construct principal components
  - principal components are constructed to be the most correlated to the response variable (like clinical traits)
  - also works for multiple responses (e.g., multiple clinical traits), and builds a latent model
- PLS has all the advantages that PCA has
  - Avoid collinearity!
  - Avoid overfitting!
- Function in `pls` package: `plsr(Y~X, ncomp, scale=F, validation=c("none","CV","LOO"))`

```
? plsr
```

```
plsres <- plsr(idiag~as.matrix(mbd[,7:30]),5,scale=T)
summary(plsres)
```

```
## Data:    X dimension: 102 24
##  Y dimension: 102 1
## Fit method: kernelpls
## Number of components considered: 5
## TRAINING: % variance explained
##          1 comps  2 comps  3 comps  4 comps  5 comps
## X         43.273   52.666    58.69    62.99    66.68
## idiag      2.802    7.867    13.07    16.08    17.75
```

- The first five components account for about 2/3 of total variation in metabolites, and about 18% of total variation in the clinical trait, significantly improved from PCR.

```
plsres <- plsr(mbd$BMI~as.matrix(mbd[,7:30]),5,scale=T)
summary(plsres)
```

```
## Data:    X dimension: 102 24
##  Y dimension: 102 1
## Fit method: kernelpls
## Number of components considered: 5
## TRAINING: % variance explained
##           1 comps  2 comps  3 comps  4 comps  5 comps
## X           44.24    50.35    59.07    64.03    68.21
## mbd$BMI     20.84    32.31    35.79    39.22    41.44
```

- The first five components account for almost 70% of total variation in metabolites, and also over 40% of total variation in BMI.

- Question: How many components to choose? How to choose?
  - Based on R^2
  - Cross-validation
  - Significance test?
  - Predictibility?

- PLS can also construct the same set of principal components for multiple traits (i.e., multiple Y)

```
dbpres <- plsr(as.matrix(mbd[,c(12,4)])~as.matrix(mbd[,7:11])
               +as.matrix(mbd[,13:30]),5)
summary(dbpres)
```

```
## Data:    X dimension: 102 23
##  Y dimension: 102 2
## Fit method: kernelpls
## Number of components considered: 5
## TRAINING: % variance explained
##                    1 comps  2 comps  3 comps  4 comps  5 co
## X                    63.66    73.63    77.82    88.81    94
## Unsaturate.Lipids    70.97    94.48    96.50    96.93    97
## BMI                  23.97    23.99    24.00    24.20    24
```

- The first five PCs account for almost 95% of total variation in metabolites (excluding the unsaturated lipids), and also account over 97% of total variation in the unsaturated lipids and almost one-quarter of total variation in BMI.

```
mbd$idiag <- as.integer(mbd$Diagnosis)
mbd$igender <- as.integer(mbd$Gender)
mbd$ismoke <- as.integer(mbd$Smoking)
mbd$ialco <- as.integer(mbd$Alcohol)
mbres <- plsr(as.matrix(mbd[,7:30])~idiag+igender+ismoke
              +ialco+I(as.matrix(mbd[,c(2,4)])),3,data=mbd)
```

```
summary(mbres)
```

```
## Data:    X dimension: 102 6
##  Y dimension: 102 24
## Fit method: kernelpls
## Number of components considered: 3
## TRAINING: % variance explained
##                          1 comps  2 comps  3 comps
## X                        46.1429  98.8788   99.211
## Formate                   9.2535   9.8472    9.905
## Histidine                20.2880  20.5142   20.817
## Phenylalanine             1.3291   1.3539    2.928
## Tyrosine                  8.9094   9.1081    9.391
## Urea                      1.5972   1.5984    2.329
## Unsaturate.Lipids        19.3383  19.3432   23.564
## Glucose                   9.3427  10.6421   13.480
## Threonine                 5.0259   5.0352    5.744
## Lactate                   3.1344   3.7664    8.050
```

Seemingly Unrelated Regression (SUR)

- Simultaneous modeling multiple traits (Y), both PLS and SUR allow each trait to borrow information from other traits
- PLS assumes a latent-variable model. That is, every trait is affected by the same set of latent varibles (PCs).
- Unlike PLS, SUR allows each trait has its unique linear model.
- Function in `systemfit` package: `systemfit(formula,method = "OLS",...)`

```
require(systemfit,warn.conflicts=F,quietly=T)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
eqAcetoac <- Acetoacetate~Age+BMI+idiag+igender+ismoke+ialco
eqX3 <- X3.hydroxybutyric.acid~Age+BMI+idiag+igender+ismoke+ia
system <- list(Acetoac=eqAcetoac,X3=eqX3)
sres <- systemfit(system,method="SUR",data=mbd)
summary(sres)
```

```
##
## systemfit results
## method: SUR
##
##          N  DF      SSR   detRCov   OLS-R2 McElroy-R2
## system 204 190 11482707 173336955 0.092359   0.074933
##
##           N DF      SSR       MSE     RMSE       R2   Adj R
## Acetoac 102 95   651239   6855.14  82.7958 0.082152 0.02418
## X3      102 95 10831468 114015.45 337.6617 0.092965 0.03567
##
## The covariance matrix of the residuals used for estimation
```

```
coefficients(summary(sres))
```

```
##                                Estimate Std. Error     t value
## Acetoac_(Intercept)  354.2969326  90.578853  3.91147516 1.7
## Acetoac_Age           -1.8018247   1.313164 -1.37212509 1.7
## Acetoac_BMI           -2.5804107   1.395189 -1.84950625 6.7
## Acetoac_idiag          5.9346055  17.519627  0.33874041 7.3
## Acetoac_igender      -12.8776547  17.167888 -0.75010127 4.5
## Acetoac_ismoke         0.6527157  17.356886  0.03760557 9.7
## Acetoac_ialco         26.6352316  18.345641  1.45185616 1.4
## X3_(Intercept)      1577.2358868 369.403012  4.26968875 4.6
## X3_Age                -6.8926237   5.355406 -1.28704025 2.0
## X3_BMI               -12.8266555   5.689926 -2.25427439 2.6
## X3_idiag              48.6615663  71.449381  0.68106352 4.9
## X3_igender           -91.9532559  70.014904 -1.31333832 1.9
## X3_ismoke            -28.4934840  70.785685 -0.40253173 6.8
## X3_ialco              51.5094591  74.818070  0.68846281 4.9
```

High-Dimensional Data & Big Data

- Challenge due to High-Dimensional Data:
  - A large number of available covariates
  - A relative small number of them are correlated to y
- Example:
  - Number of metabolites may be much larger than the sample size!
- With all metabolites included to study a clinical trait, principal components may be dominated by variation of unrelated metabolites
  - The importance of related metabolites may be significantly perturbed
  - Reults in low predictibility

```
library(POCRE); data(simdata)
yy <- simdata[1:50,1]
xx <- as.matrix(simdata[1:50,2:1001])
xxs <- as.matrix(xx[,1:200])
axpls <- plsr(yy~xx,5) # Using all x
sxpls <- plsr(yy~xxs,5) # Using selected x
xxn <- simdata[51:100,2:1001] # new x
xxsn <- as.matrix(xxn[,1:200]) # new x
ya <- predict(axpls,xxn)
ys <- predict(sxpls,xxsn)
sum((simdata[51:100,1]-ya[,,5])^2) #prediction error when usi
```

```
## [1] 34609.83
```

```
sum((simdata[51:100,1]-ys[,,5])^2) #prediction error when usi
```

```
## [1] 1148.673
```

- The simulated Y is affected by twenty true predictors: $X_1, \cdots, X_{10}, X_{101}, \cdots, X_{110}$
- When we apply PLS to all 1,000 predictors, the prediction error is over 34,000.
- When we apply PLS to 200 predictors including the true twenty, the prediction error is dramatically decreased under 1,150.
- Indeed, the correlation of predicted values to the true values is also significantly increased from 0.15 to 0.99.

```
cor(simdata[51:100,1],cbind(ya[,,5],ys[,,5]))
```

```
##              [,1]      [,2]
## [1,] 0.1523343 0.9858016
```

- Therefore, it is crucial to selecte important predictors to build up high-dimensional models, even for supervised dimension reduction.

Penalized Orthogonal-Components Regression (POCRE)

- POCRE is a supervised dimension reduction method for high-dimensional data
- POCRE simultaneously selects important variables and constructs principal components of selected variables
- Like PLS, POCRE constructs principal components which are the most correlated to Y
- Like PLS, POCRE also works for multiple Y, and builds a latent model
- Advantage:
  - Avoid collinearity!
  - Avoid overfitting!
  - Select important variables
- The R package `POCRE` is available: POCRE_0.1.0.tar

```
install.packages("POCRE_0.1.0.tar")
```

- Major functions available in POCRE:
  - `pocrescreen` – Screen for a pre-specified number of predictors based on supervised dimension reduction
  - `pocre` – Build linear regression model based on supervised dimension reduction with a pre-specified tuning parameter
  - `pocrepath` – Build linear regression model for a series of tuning parameters
  - `selectmodel` – Select the optimal tuning parameter and the corresponding model based on information criteria, including EBIC, BIC, AIC, AICc.
  - `cvpocre` – Choose the optimal tuning parameter via cross-validation
  - `sipocre` – Evaluate the significance of predictions identified by POCRE using the multiple splitting method.

pocrescreen(inY, inX, maxvar=nrow(inX), maxcmp=5, inEIdx=NULL, ...)

- It screens the variables and stores the selected predictors and their indices.

```
xx <- scale(as.matrix(simdata[,-1]))
yy <- scale(as.matrix(simdata[,1]))
psres <- pocrescreen(yy,xx,maxvar=50,maxcmp=10)
```

```
##  Screening variables ..........
```

```
psX <- psres$retX; psXIdx <- psres$retSIdx
rbind(psXIdx[1:10],psXIdx[11:20])
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    2    3    4    5    6    7    9   10    11
## [2,]   13   15  101  102  104  105  106  107  108   110
```

pocre(inY, inX, inTP=1, covidx=NA, maxvar=dim(inX)[1]/2, maxcmp=10, ...)

- Build linear regression model based on supervised dimension reduction with a pre-specified tuning parameter (inTP)
  - The tuning parameter should be positive and usually around one, implying that the correlation among high-dimensional data may bias down or up the variance estimate.

```
tres <- pocre(yy,xx)$retRes  # inTP=1 by default
tXIdx <- which(abs(tres$beta)>1e-6)
tXIdx
```

- POCRE identifies $X_1, \cdots, X_{15}, X_{101}, \cdots, X_{117}$, including all true predictors

pocrepath(inY, inX, covidx=NA, XId=NA, maxvar=dim(inX)[1]/2, maxcmp=10, delta=0.1, ...)

- Run POCRE by automatically scanning a series of tuning parameter values around one

```
ppres <- pocrepath(yy,xx,delta=0.01)     # Scan tuning paramete
```

- By default, pocrepath() starts at inTP=1 and increase inTP by delta consecutively until it will identify no predictor because of too large inTP; Then it will decrease at inTP=1-delta until it identifies too many predictors.
  - It will return the results for each scanned tuning parameter inTP (also known as lambda in the below).
  - The function `selectmodel` can be applied to pocrepath() results to select the optimal tuning parameter.

- `selectmodel(inRes)` select the optimal tuningparameter based on some information criteria, such as EBIC, BIC, AIC, and AICc.

```
optres <- selectmodel(inRes=ppres)
```

- Several functions are available in POCRE package to plot the results of pocrepath() and help select the tuning parameter (inTP=lambda).
  - plotbetanzbeta() provides the plot of lambda vs. beta and number of nonzero-beta for the results from pocrepath().
  - plotbetarsq() provides the plot of lambda vs. beta and R^2 for the results from pocrepath().
  - plotrsqnzbeta() provides the plot of lambda vs. R^2 and number of nonzero-beta.

```
plotbetanzbeta(ppres)
plotbetarsq(ppres)
plotrsqnzbeta(ppres)
```

- As shown in the above figures, we may choose an appropri.ate tuning parameter (inTP=lambda) based on R^2 and the number of identified predictors.

- POCRE can also fit a high-dimensional linear regression model for multiple traits (Y)
- Example: the data set `sim5ydata` in `POCRE` package has five response variables simulated from the same components.

```
data('sim5ydata')
dim(sim5ydata)
```

```
## [1]  100 1005
```

```
xx = as.matrix(sim5ydata[,-(1:5)])
yy = as.matrix(sim5ydata[,1:5])
```

- Similarly, we can first run `pocrepath()` to automatically scan a seires of possible tuning parameter values (inTP=lambda), and then use 'selectmodel()' to select the optimal tuning parameter based on some information criteria.

```
ppres <- pocrepath(yy, xx, delta=0.01)
optres <- selectmodel(inRes = ppres)
```

- Again, we can use the differnt functions to plot the results of `pocrepath()` and select an appropriate tuning parameter based on R^2 and/or the number of identified predictors.

```
plotbetanzbeta(ppres)
plotbetarsq(ppres)
plotrsqnzbeta(ppRes)
plotcomponents(ppres,inLambda=optres$lambda)
```

A Real High-Dimensional Data Set

- The objective of this research is to assess the effect of the miRNA on the protein expression in breast cancer.
  - Tumors from 283 primary breast cancer patients belonging to Oslo2 cohort were profiled for genome-wide miRNA expression using Agilent microarrays
  - A selected panel of 105 cancer-related proteins were profiled for protein expression using reverse-phase protein arrays as well.
- The miRNA expression data can be found on Gene Expression Omnibus (GEO) database with accession number GSE58210.

*The protein expression data can be found on the additional file 4 attached to the paper + Aure MR, Jernstrom S, Krohn M, Vollan HK, Due EU, Rodland E, Oslo Breast Cancer Research Consortium, Ram P, Lu Y, Mills GB, Sahlberg KK, Borresen-Dale A-L, Lingjerde OC, Kristensen VN (2015). Integrated analysis reveals microRNA networks coordinately expressed with key proteins in breast cancer. Genome Medicine, 7: 21.

```
protein<-read.xlsx('pe.xlsx',17,startRow = 2,
                   colNames = TRUE,rowName=T)[,-1]
protein<-t(protein)
miRNA<-fread('GSE58210_NormalizedData_withannotations.txt',
             header=T,fill=T)
miRNA<-as.data.frame(miRNA)
namemiRNA<-miRNA[,1]
miRNA<-miRNA[,-(1:7)]
rownames(miRNA)<-namemiRNA
miRNA<-t(miRNA)
ry<-protein
rx<-miRNA
dim(rx)
dim(ry)
```

- We first screen the variables with `pocrescreen()` and store the selected X variables and their indices

```
resultmiRNA <- pocrescreen(ry, rx, maxvar=100, maxcmp=5)
tmpX <- resultmiRNA$retX
tmpXIdx <- resultmiRNA$retSIdx
```

- We then fit the data with `pocrepath()`

```
ppResmiRNA <- pocrepath(ry, inX=tmpX, XId=tmpXIdx, delta=0.025
                        maxvar=50, maxcmp=10, pval=F)
```

- We then use `selectmodel()` to select the best model on the basis of a prespecified criterion (AIC by default)

```
optResmiRNA <- selectmodel(inRes=ppResmiRNA)
```

- We can also visualize the results by interactive plots

```
plotbetanzbeta(ppResmiRNA,XId=tmpXIdx)
plotbetarsq(ppResmiRNA,XId=tmpXIdx)
plotrsqnzbeta(ppResmiRNA)
```

- We can plot the principal components for the optimal model we select.

```
plotcomponents(ppResmiRNA,inLambda=optResmiRNA$lambda,
               XIdx=tmpXIdx,name=T)
```

Bibliography

- Zhang, D., Lin, Y., & Zhang, M. (2009). Penalized orthogonal-components regression for large p small n data. Electronic Journal of Statistics, 3: 781-796. doi:10.1214/09-ejs354
- Aure MR, Jernstrom S, Krohn M, Vollan HK, Due EU, Rodland E, Oslo Breast Cancer Research Consortium, Ram P, Lu Y, Mills GB, Sahlberg KK, Borresen-Dale A-L, Lingjerde OC, Kristensen VN (2015). Integrated analysis reveals microRNA networks coordinately expressed with key proteins in breast cancer. Genome Medicine, 7: 21.

Acknowledgement

- Collaborators:
  - Dan Raftery
  - Nagana Gowda
  - Marietta Harrison
- Graduate Students:
  - Chen Chen
  - Zhongli Jiang
  - Yanzhu Lin
  - Zeyu Zhang
- NIH Big Data to Knowledge (BD2K)
- National Cancer Institute
- National Science Foundation