Protocol for Partial Least Squares Regression Analysis Using the PLS R package

Nisha G. Sosale

02-24-2019

Using the PLS package to run PLSR Analysis in R

This is a summary of my method for using the R package PLS to perform PLSR analysis. The PLS pacakge implements Principal Component Regression and Partial Least Squares Regression in R. For more details on using the PLS package see the paper describing the package https://www.jstatsoft.org/article/view/v018i02 (https://www.jstatsoft.org/article/view/v018i02) and the associated vignette https://cran.r-project.org /web/packages/pls/vignettes/pls-manual.pdf (https://cran.r-project.org/web/packages/pls/vignettes/pls-manual.pdf).

Dataset

The dataset I'll be using to illustrate the utility of the package is Luminex, Cell Death and Cell Cycle data for the **glioblastoma SF188 cell line** in response to drug treatment. This dataset contains **8 conditions**: SF188 cells with or without a SPRY2 knockdown treated with two classes of drugs, DNA damaging agents or Receptor Tyrosine Kinase inhibitors (RTKi). The two DNA damaging agents examined here are carboplatin (carbo) and temozolomide (temo). Temozolomide was tested at two concentrations, 400 uM or 1600 uM. The RTKis examined here are a combination treatment of an EGFR inhibitor (Gefitinib or G) and a Met inhibitor (PHA665752 or P).

Load Packages

```
library(pls)
library (readxl)
library(NMF)
```

Import X and Y Matrix

The **X matrix** contains cell signal data obtained from a quantitative Luminex assay of phosphoprotein levels in each of the 8 conditions. This data has been Z-scored, or mean centered and variance scaled.

```
X0=read.table("Xmatrix.csv",sep=",",header=T)
X1=X0[,2:ncol(X0)]
rownames(X1)=X0$names
X1[,1:6]
```

```
##                   CREB_0h CREB_0.5h CREB_3h CREB_6h CREB_21h CREB_24h
## control_GP          -1.22     -0.21   -1.25   -0.85    -0.75    -1.54
## control_carbo       -1.22      1.87   -0.15    0.20    -0.09    -0.42
## control_temo_1600   -1.22      1.93   -0.15   -0.08    -0.13    -0.07
## control_temo_400    -1.22     -0.60    0.03   -0.83    -0.15    -0.23
## SPRY2kd_GP          -0.75     -0.69   -0.72   -0.66    -0.34    -0.75
## SPRY2kd_carbo       -0.75      2.97    0.73    0.80     0.61     0.21
## SPRY2kd_temo_1600   -0.75      3.08    0.09   -0.42     0.19     0.07
## SPRY2kd_temo_400    -0.75      0.75    1.12   -0.84    -0.41    -0.35
```

The **Y matrix** contains cell phenotype data for the 8 conditions. This is cell death data obtained from a flow cytometry To-Pro-3 permeability assay and cell cycle data from a flow cytometry based assay of DNA content. This data has been Z-scored, or mean centered and variance scaled.

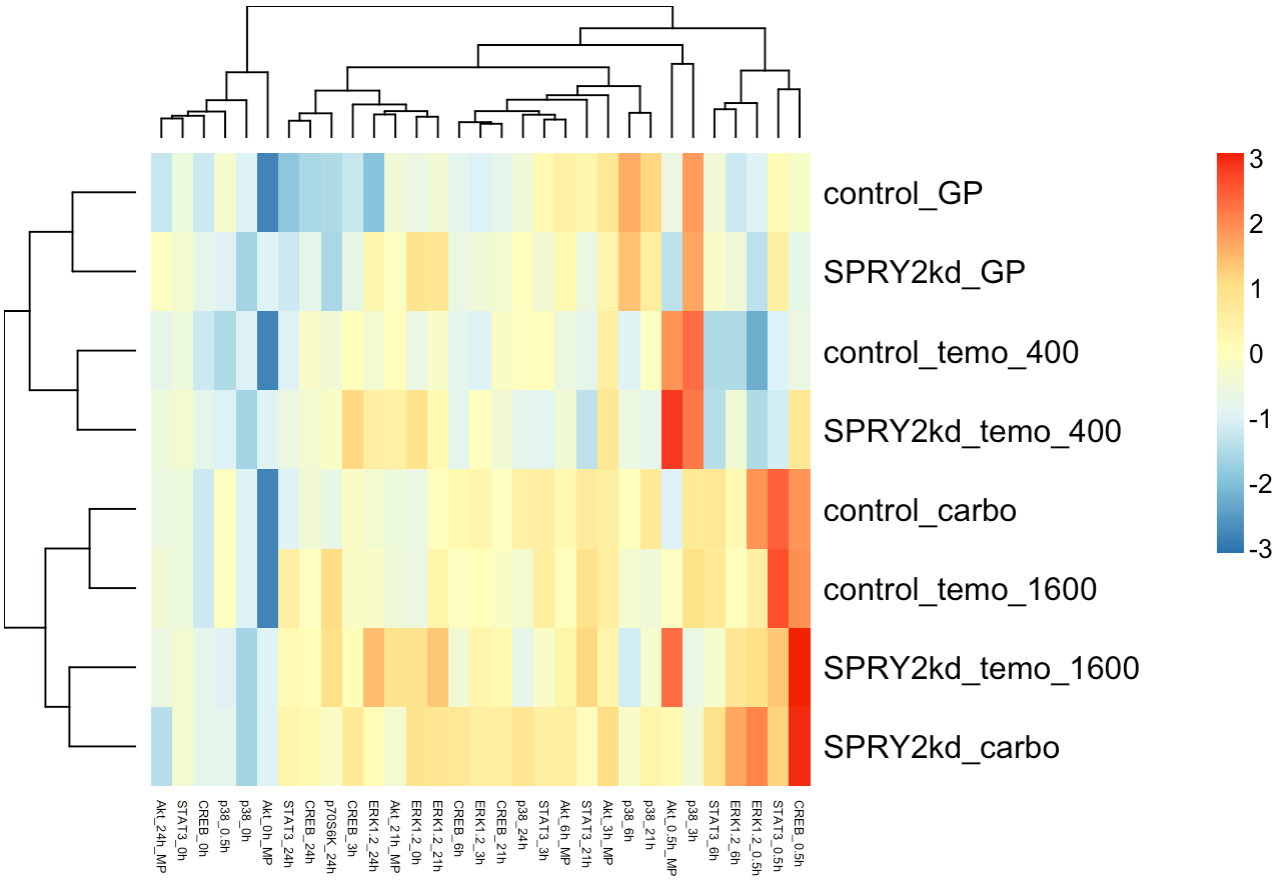
```
Y0=read.table("Ymatrix.csv", sep=",",header=T)
Y1=Y0[,2:ncol(Y0)]
rownames(Y1)=Y0$names
Y1
```

```
##                   Death    G1     S    G2
## control_GP        -0.84  1.69 -1.09 -1.27
## control_carbo     -0.02 -0.80  1.62 -0.19
## control_temo_1600 -0.62 -0.97  0.06  1.14
## control_temo_400  -0.08  0.34 -0.28 -0.21
## SPRY2kd_GP         1.88  1.16 -0.42 -1.11
## SPRY2kd_carbo      1.11 -0.49  1.43 -0.44
## SPRY2kd_temo_1600 -0.48 -0.92 -0.51  1.49
## SPRY2kd_temo_400  -0.95 -0.01 -0.80  0.59
```

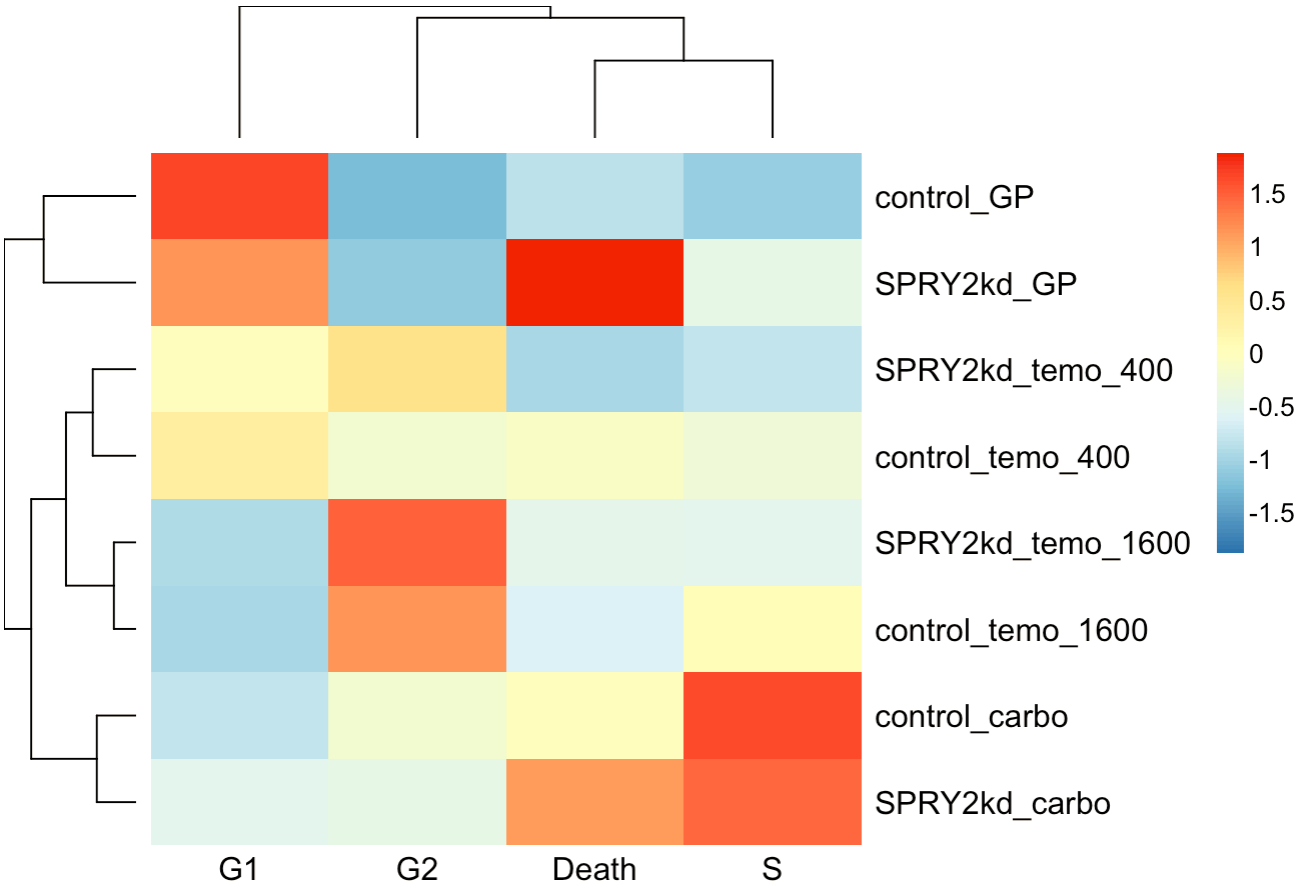
Heatmaps of X and Y Data

Before beginning the PLSR analysis, I use the aheatmap function from the NMF package to visualize the data. For more details on using the NMF package see the vignette that describes the package https://cran.r-project.org/web/packages/NMF/vignettes/NMF-vignette.pdf (https://cran.r-project.org/web/packages/NMF/vignettes/NMF-vignette.pdf).

```
aheatmap((X1),breaks=c(0),revC=T)
```




```
aheatmap((Y1),breaks=c(0),revC=T)
```



PLSR using pls package

create data frame of X and y matrix

```
X <- data.matrix((X1))
y <- data.matrix((Y1))
mydata <- data.frame(yy=I(y),xx=I(X))
```

run PLSR

```
PLSR_matrix <- plsr(yy~xx, scale=F, data=mydata, method="kernelpls")
```

look at summary of PLSR results from PLS package

```
summary(PLSR_matrix)
```

```
## Data:    X dimension: 8 31
##  Y dimension: 8 4
## Fit method: kernelpls
## Number of components considered: 7
## TRAINING: % variance explained
##         1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X      47.68049    75.49    88.24    92.67    96.67    98.45      100
## Death   0.00124    28.40    73.86    77.43    92.81    99.98      100
## G1     79.78206    85.66    89.06    93.32    97.08    99.00      100
## S      36.31232    60.00    60.02    95.96    96.07    96.17      100
## G2     42.42777    84.36    89.96    93.29    97.84    99.97      100
```

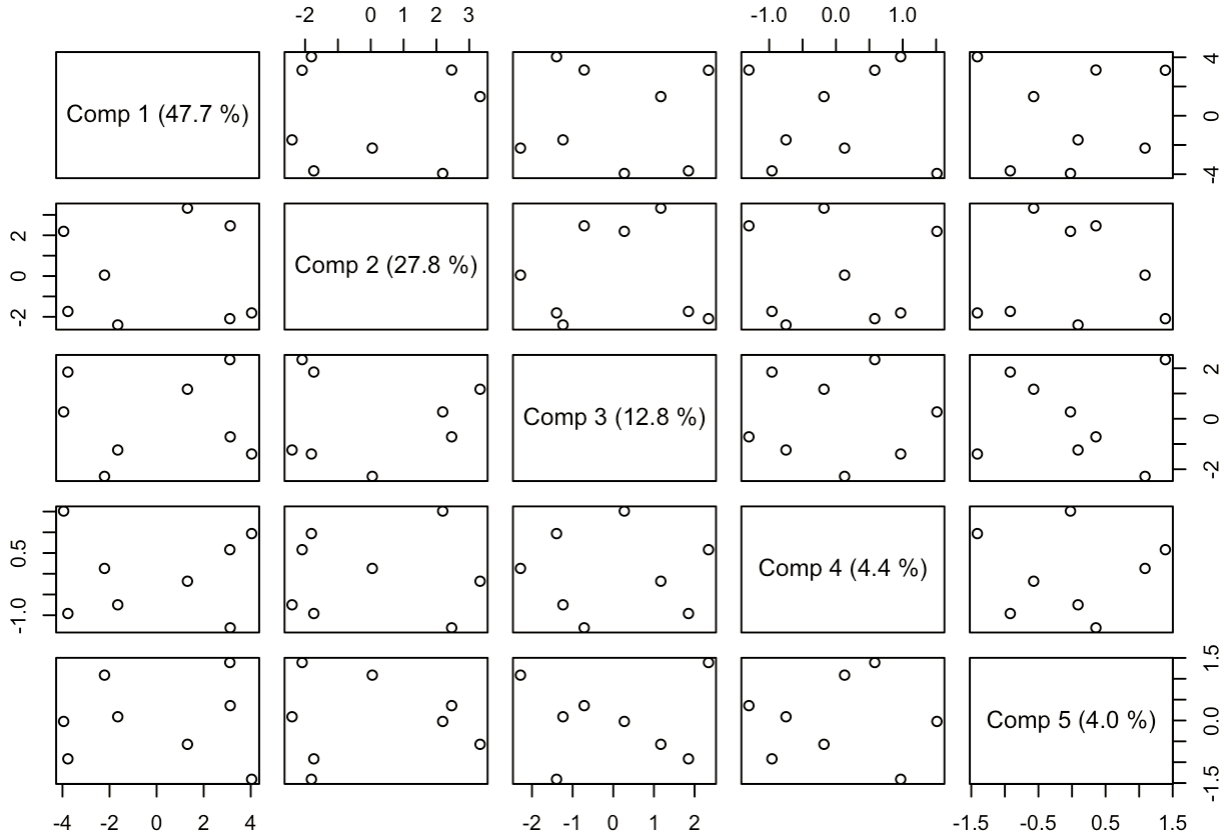
output scores and loadings

```
sc <- scores(PLSR_matrix)
ld <- loadings(PLSR_matrix)
Ysc <-Yscores(PLSR_matrix)
Yld <- Yloadings(PLSR_matrix)

write.table(sc, file="scores.txt",col.names=T,row.names=T,sep="\t", quote=F)
write.table(ld, file="loadings.txt",col.names=T,row.names=T,sep="\t", quote=F)
write.table(Ysc, file="Yscores.txt",col.names=T,row.names=T,sep="\t", quote=F)
write.table(Yld, file="Yloadings.txt",col.names=T,row.names=T,sep="\t", quote=F)
```

Plot scores

```
plot(PLSR_matrix, plottype = "scores", comps = 1:5)
```


Comp 1 (47.7 %)
Comp 2 (27.8 %)
Comp 3 (12.8 %)
Comp 4 (4.4 %)
Comp 5 (4.0 %)