Practical session: Reconstruction of protein-protein interaction and metabolic network

Jean-Philippe Vert

In this session you will

- Implement a method for the prediction of undirected edges of a networks, with the TPPK kernel
- Use it to predict protein-protein interactions (PPI) and missing edges in metabolic pathways

1 Data preparation

1.1 Download networks and genomic data

Download two networks from the yeast (PPI network and metabolic pathway), as well as several kernel matrices to represent different informations about the genes, from the course webpage. For example, from a command window, type:

```
wget http://cbio.ensmp.fr/~jvert/svn/tutorials/practical/netinference/ppimetabo.tar.gz
tar xvzf ppimetabo.tar.gz
```

This will create two folders, named respectively `metabolic` and `interaction`. Each contains several text files: one to describe the network, and several to describe kernels.

1.2 Read the data in R and reduce the network

For sake of simplicity, we will focus on single network (metabolic network), to be predicted from a single kernel (which integrates the information of all other kernels). We read the corresponding files, and create a dataset with all known interactions (positive examples), and the same number of unknown interactions (negative examples).

```
# Load network
admat <- as.matrix(read.table('metabolic/gold_admat.txt'))
n <- dim(admat)[1]

# Load data (kernel)
K <- as.matrix(read.table('metabolic/Kmat_intg.txt'))

# We extract the list of interactions
xpos <- which(admat>0,arr.ind=TRUE)
xpos <- xpos[xpos[,1]>xpos[,2],]
npos <- dim(xpos)[1]

# Generate the same number of negative pairs
ineg <- sample(seq(admat)[-which(admat!=0)] , npos)
```

```
xneg <- cbind((ineg-1)%%n+1 , (ineg-1)%/%n+1)

# Create the full dataset
x <- rbind(xpos,xneg)
y <- c(rep(1,npos),rep(-1,npos))
```

2 Implement the TPPK kernel

We implement a method to predict the label of a pair of genes (interaction or non-interaction), from a training of pairs with known labels. This can be formulated as a classical supervised binary classification problem, which we will solve with a SVM.

However, each example is a *pair* of genes, and we just have a kernel for *individual* genes. To overcome this problem, we create a kernel for pairs from a kernel for individuals with the TPPK formulation:

$$K_{TPPK}((a,b),(c,d)) = K(a,c)K(b,d) + K(a,d)K(b,c)\,.$$

In `R`, we represent a pair of genes by a vector of the form `c(i,j)`, where `i` and `j` are integers between 1 and n (the total number of genes). Then for a given kernel matrix for individuals `K`, we need a kernel function `Kpair` that takes two such vectors `x=c(i,j)` and `u=c(k,l)` as input, and output the TPPK kernel, i.e.:

```
Kpair(x,u) = K[i,k]*K[j,l] + K[i,l]*K[j,k]
```

Question 1 Implement in `R` a function `tppk <- function(Kindiv=matrix())` such that `Kpair <- tppk(K)` is the TPPK kernel associated to `K`.

Question 2 Check that it works by computing a simple Gram matrix with the command `kernelMatrix`

3 Prediction of metabolic interactions

Question 3 Assess the ability of a SVM with the TPPK kernel to correctly predict metabolic reactions. Run a cross-validation procdure, or a single train/test split.

Question 4 If time allows run the experiments on different networks and different input kernels.