# nQTL instruction

## A、 Data preparation：

**1、 Require data:**

SNP data matrix, gene expression matrix, gene symbol list, PPI net.

**2、 Data format:**

SNP data matrix and gene expression matrix should have equal number of columns representing the same samples; gene symbol list should be corresponding to the feature/row id of gene expression matrix, e.g. the first list is feature id, the second list is related gene symbol; PPI net contains list of gene-pairs, each row represent one gene-pair including two gene symbols.

## B、Code usage：

**1、 Load R packages:**

```
Library(MatrixEQTL)
library(R.matlab)
```

**2、 Parameter setting:**

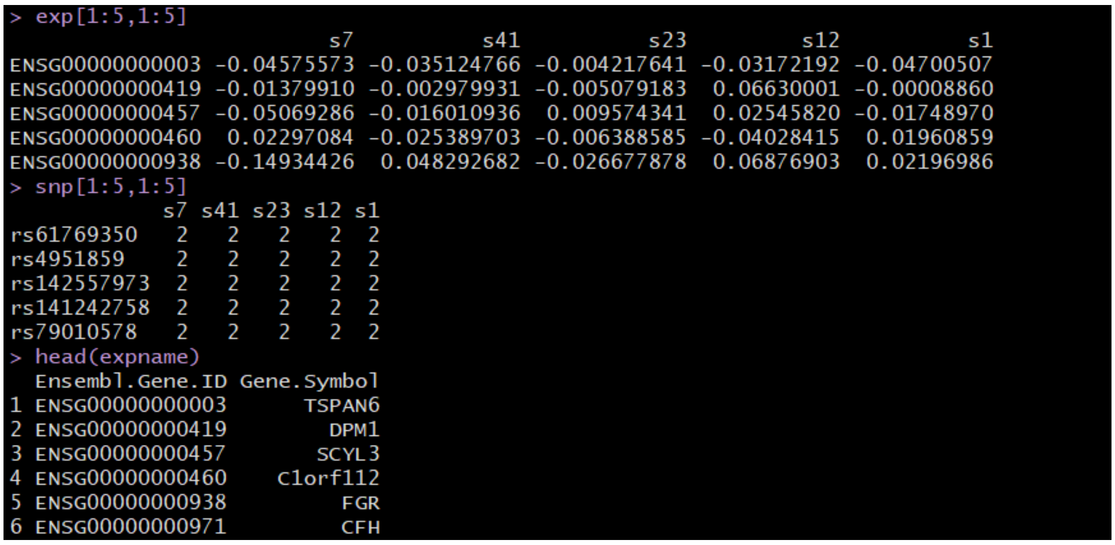
When SNP data is continuous, the modelLINEAR is used; and when SNP data isdiscrete, the modelANOVA is sued.

```
useModel1= modelANOVA;
useModel2= modelLINEAR;
pvOutputThreshold1 = 1e-100;
pvOutputThreshold2 = 1e-45;
```

**3、 Read data input：**

```
exp<-read.csv(file = " /example/exp.csv",row.names = 1)
snp<-read.csv(file=" /example/snp.csv",row.names = 1)
expname<-read.csv(file = " /example/genename.csv")
ppi<-read.csv("/expample/String_9606_v10.5_s900_new.csv",header= F)
```

The input data format is as follows：

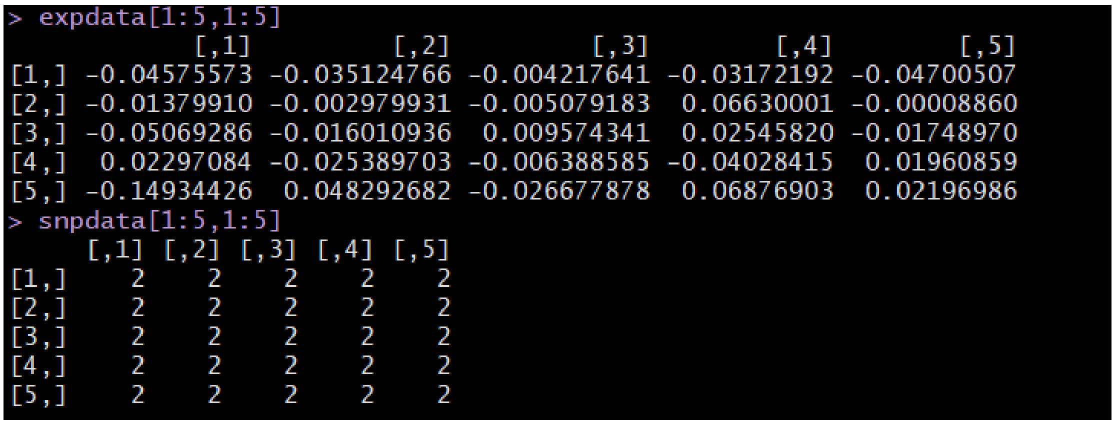


```
> exp[1:5,1:5]
                         s7          s41          s23         s12          s1
ENSG00000000003 -0.04575573 -0.035124766 -0.004217641 -0.03172192 -0.04700507
ENSG00000000419 -0.01379910 -0.002979931 -0.005079183  0.06630001 -0.00008860
ENSG00000000457 -0.05069286 -0.016010936  0.009574341  0.02545820 -0.01748970
ENSG00000000460  0.02297084 -0.025389703 -0.006388585 -0.04028415  0.01960859
ENSG00000000938 -0.14934426  0.048292682 -0.026677878  0.06876903  0.02196986
> snp[1:5,1:5]
            s7 s41 s23 s12 s1
rs61769350   2   2   2   2  2
rs4951859    2   2   2   2  2
rs142557973  2   2   2   2  2
rs141242758  2   2   2   2  2
rs79010578   2   2   2   2  2
> head(expname)
  Ensembl.Gene.ID Gene.Symbol
1 ENSG00000000003      TSPAN6
2 ENSG00000000419        DPM1
3 ENSG00000000457       SCYL3
4 ENSG00000000460    C1orf112
5 ENSG00000000938         FGR
6 ENSG00000000971         CFH
```

**4、 Data format adaption：**

```
expdata<-apply(exp,2,as.numeric)
colnames(expdata)<-NULL
snpdata<-apply(snp,2,as.numeric)
colnames(snpdata)<-NULL
```

The intra data format is as follows:



```
> expdata[1:5,1:5]
            [,1]         [,2]         [,3]        [,4]        [,5]
[1,] -0.04575573 -0.035124766 -0.004217641 -0.03172192 -0.04700507
[2,] -0.01379910 -0.002979931 -0.005079183  0.06630001 -0.00008860
[3,] -0.05069286 -0.016010936  0.009574341  0.02545820 -0.01748970
[4,]  0.02297084 -0.025389703 -0.006388585 -0.04028415  0.01960859
[5,] -0.14934426  0.048292682 -0.026677878  0.06876903  0.02196986
> snpdata[1:5,1:5]
     [,1] [,2] [,3] [,4] [,5]
[1,]    2    2    2    2    2
[2,]    2    2    2    2    2
[3,]    2    2    2    2    2
[4,]    2    2    2    2    2
[5,]    2    2    2    2    2
```

**5、 Edge co-expression calculation：**

```
exp$mean<-apply(exp, 1, mean)
myfuntion1<-function(ppi,expname){
if (length(which(expname[2]==as.character(ppi[1]))[1])==0){
            x<-0
    }else
            x<-which(expname[2]==as.character(ppi[1]))[1]
}
x<-apply(ppi, 1, myfuntion1,expname)
   myfuntion2<-function(ppi,expname){
if (length(which(expname[2]==as.character(ppi[2]))[1])==0) {
            x<-0
   }else
          x<-which(expname[2]==as.character(ppi[2]))[1]
}
    y<-apply(ppi,1, myfuntion2,expname)
    expsiteppi<-data.frame(cbind(x,y))
    expsiteppi<-na.omit(expsiteppi)
     save(expsiteppi,file="/example/expsiteppi.RData")

    c<-dim(exp)[2]
    pair<-data.frame()
    for (s in 1:(dim(exp)[2]-1)){
            for (i in 1:dim(expsiteppi)[1]){
               pair[i,s]<-(exp[expsiteppi[i,1],s]-exp[expsiteppi[i,1],c])*
                          (exp[expsiteppi[i,2],1]-exp[expsiteppi[i,2],c])
```

```
        }
}
save(pair,file="/example/pair.RData")
```

**6、 nQTL calculation:**

```
    snps1 = SlicedData$new( snpdata );
    gene1 = SlicedData$new( pair );
    cvrt1 = SlicedData$new( );
    snps1$ResliceCombined(500);
    gene1$ResliceCombined(500);
    filename = tempfile();
    errorCovariance = numeric();
    meh = Matrix_eQTL_main( snps = snps1, gene = gene1, cvrt = cvrt1, output_file_name = filename, pvOutputThreshold = pvOutputThreshold1, useModel = useModel1, errorCovariance = errorCovariance, verbose = TRUE, pvalue.hist = 100);
    unlink( filename );
    # png(filename = "histogram.png", width = 650, height = 650)
    plot(meh, col="grey")
    # dev.off();

    # a Q-Q plot
    meq = Matrix_eQTL_main( snps = snps1, gene = gene1, cvrt = cvrt1, output_file_name = filename, pvOutputThreshold = pvOutputThreshold2, useModel = useModel1, errorCovariance = errorCovariance,    verbose = TRUE, pvalue.hist = "qqplot");
    unlink( filename );
    # png(filename = "QQplot.png", width = 650, height = 650)
    plot(meq, pch = 16, cex = 0.7)
    # dev.off();
```

## C、Demo experiment

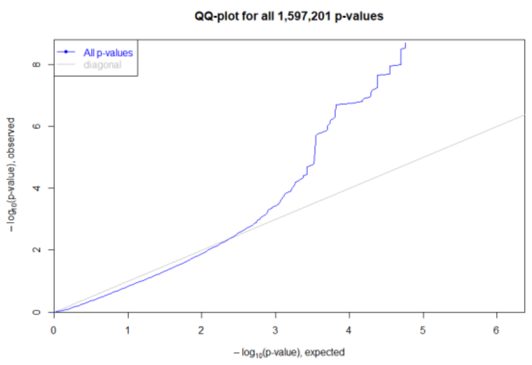
**1、Demo data：**

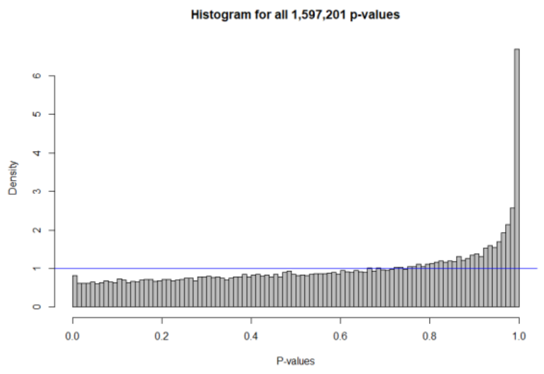
Size of SNP data: 1999*10

Size of expression data: 799*10

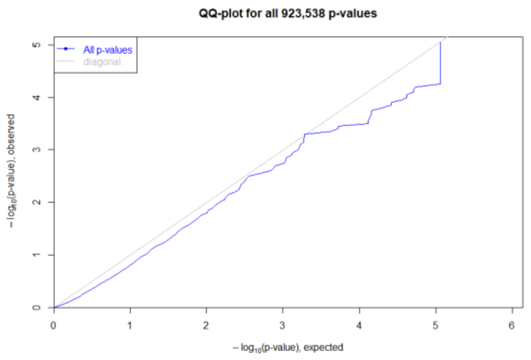
Size of co-expression data: 462*10 （ppi>900）

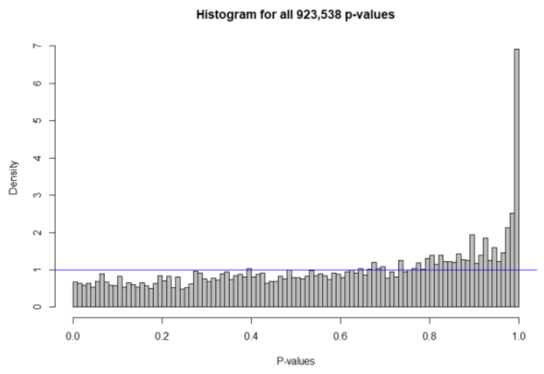
**2、eQTL outcomes：**





### 3、nQTL outcomes：



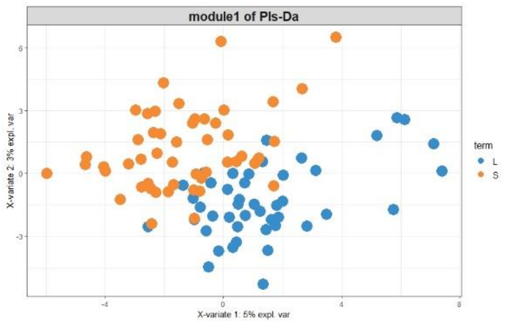


## D、List of software used in down-stream analysis of nQTL framework

### 1、PLSDA

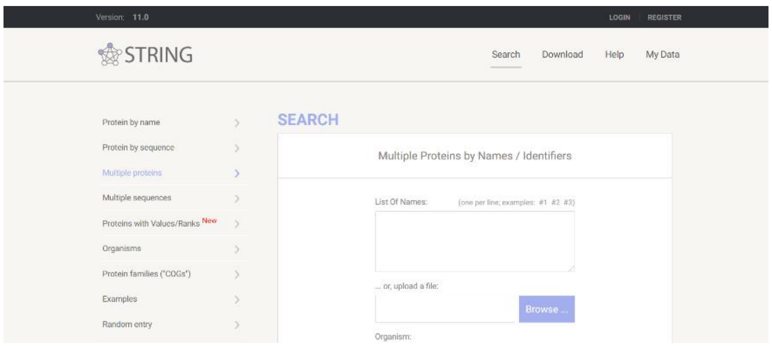
```
library(mixOmics)
#set working path(setwd(working path))
exp<-read.csv("./expdata.csv",row.names = 1)
#Delete outliers
exp<-exp[,-24]
module<-read.csv("./modulegene.csv")
sampletype<-read.csv("./sampletype.csv")
sampletype<-sampletype[sampletype$X.Sample_title%in%colnames(exp),]
exp<-as.data.frame(t(exp))
exp$term<-sampletype$term
exp<-exp[order(exp$term),]
a <- dim(exp)[2]
exp<-as.data.frame(t(exp[,-a]))

exp$term<-sampletype$type.cancer
exp<-exp[order(exp$term),]
exp<-as.data.frame(t(exp[,-a]))

data<-exp[rownames(exp)%in%module$cluster1,]
data<-apply(data,2,as.numeric)
XXt<-t(scale(data))
```

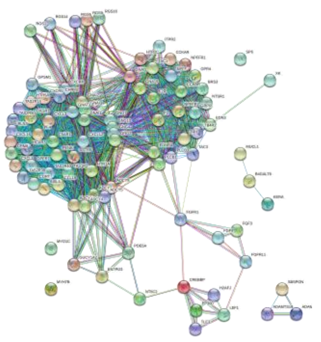
```
YY <- c(rep(c("adenocarcinoma","large cell carcinoma","squamous cell carcinoma "),c(50,20,26)))
YY<-c(rep(c("L","S"),c(45,51)))#names(datat)
plsda.datatm <-plsda(XXt, YY, ncomp = 2)
n <- dim(exp)[1]
plsda.datatm$names$sample<-c(1:n)
plotIndiv(plsda.datatm,ind.names = F,pch = 16, plot.ellipse = TRUE,title="module1 of Pls-Da" ,legend = T,legend.title = "term",cex = 6,add.legend =TRUE,style="ggplot2")
a<-vip(plsda.datatm)
```



**2、PPI network and module**

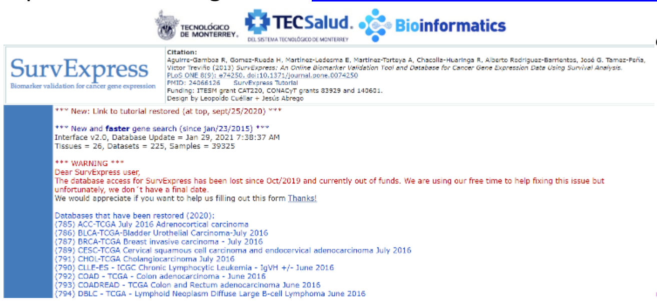
Input the module genes to [https://string-db.org/cgi/input?sessionId=bUDUQJ46kRyH&input_page_active_form=multiple_identifiers](https://string-db.org/cgi/input?sessionId=bUDUQJ46kRyH&input_page_active_form=multiple_identifiers)





**3、Survival analysis**

Input the module genes to [http://bioinformatica.mty.itesm.mx:8080/Biomatec/SurvivaX.jsp](http://bioinformatica.mty.itesm.mx:8080/Biomatec/SurvivaX.jsp)



TECNOLÓGICO DE MONTERREY.
TECSalud.
DEL SISTEMA TECNOLÓGICO DE MONTERREY
Bioinformatics
SurvExpress
Biomarker validation for cancer gene expression

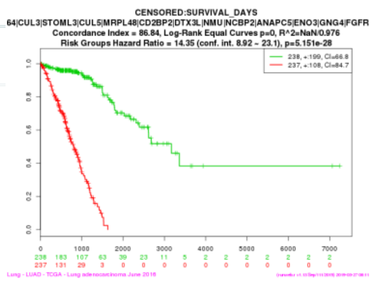
Citation:
Aguirre-Gamboa R, Gomez-Rueda H, Martinez-Ledesma E, Martinez-Torteya A, Chacolla-Huaringa R, Alberto Rodriguez-Barrientos, José G. Tamez-Peña, Victor Treviño (2013) *SurvExpress: An Online Biomarker Validation Tool and Database for Cancer Gene Expression Data Using Survival Analysis.*
PLoS ONE 8(9): e74250. doi:10.1371/journal.pone.0074250
PMID: 24066126 SurvExpress Tutorial
Funding: ITESM grant CAT220, CONACyT grants 83929 and 140601.
Design by Leopoldo Cuéllar + Jesús Abrego

*** New: Link to tutorial restored (at top, sept/25/2020) ***

*** New and **faster** gene search (since jan/23/2015) ***
Interface v2.0, Database Update = Jan 29, 2021 7:38:37 AM
Tissues = 26, Datasets = 225, Samples = 39325

*** WARNING ***
Dear SurvExpress user,
The database access for SurvExpress has been lost since Oct/2019 and currently out of funds. We are using our free time to help fixing this issue but unfortunately, we don´t have a final date.
We would appreciate if you want to help us filling out this form Thanks!

Databases that have been restored (2020):
(785) ACC-TCGA July 2016 Adrenocortical carcinoma
(786) BLCA-TCGA-Bladder Urothelial Carcinoma-July 2016
(787) BRCA-TCGA Breast invasive carcinoma - July 2016
(789) CESC-TCGA Cervical squamous cell carcinoma and endocervical adenocarcinoma July 2016
(791) CHOL-TCGA Cholangiocarcinoma July 2016
(790) CLLE-ES - ICGC Chronic Lymphocytic Leukemia - IgVH +/- June 2016
(792) COAD - TCGA - Colon adenocarcinoma - June 2016
(793) COADREAD - TCGA Colon and Rectum adenocarcinoma June 2016
(794) DBLC - TCGA - Lymphoid Neoplasm Diffuse Large B-cell Lymphoma June 2016



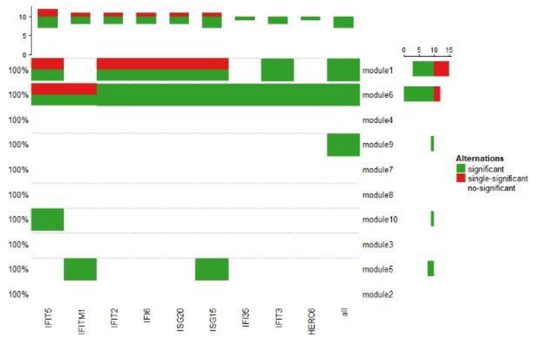
**4、CCA**

```
##set working path(setwd(working path))
```

```
library(vegan)
load("./pair.RData")
load("./exp.RData")
genename<-read.csv("./genename.csv")
modulepair<-read.csv("./geneorder.csv")
inectgene<-c('IFITM1')
#Select relevant factors :'ISG15',,'IFI35','IFIT5','IFIT3','ISG20','HERC6','IFIT2','IFI6','TNF'
inectgene<-genename[genename$Gene.Symbol%in%inectgene,]
inectexp<-exp[rownames(exp)%in%inectgene$Ensembl.Gene.ID,]
rownames(inectexp)<-inectgene[inectgene$Ensembl.Gene.ID%in%rownames(inectexp),2]
inectexp<-as.data.frame(t(inectexp))
module<-modulepair[modulepair$ClassGene=='Cluster9',]#change different module
modulepair<-as.data.frame(t(pair[rownames(pair)%in%module$X,]))

sumpair<-apply(modulepair, 1, sum)
modulepair<-modulepair[-(which(sumpair<0.05)),]
inectexp<-inectexp[rownames(inectexp)%in%rownames(modulepair),]

modulecca<-cca(modulepair,inectexp)
plot(modulecca)
permutest(modulecca,permu=999)
ef<-envfit(modulecca,inectexp,permu=999)
```

you need to record data into tables to plot

```
#heatmap
library(ComplexHeatmap)
data1<-read.csv("./cca.csv",header = T,stringsAsFactors = F,row.names = 1)
col = c( "significant" = "#33A02C", "single-significant" = "#E31A1C")
alter_fun = function(x, y, w, h, v) {
   n=sum(v)
   h=h*0.9
   grid.rect(x, y, w-unit(0.5, "mm"), h-unit(0.5, "mm"), gp = gpar(fill = "#CCCCCC", col = NA))
   if(v["single-significant"])   grid.rect(x, y - h*0.5 + 0.95:n/n*h, w*1, 1/n*h,   gp = gpar(fill = 
col[names(which(v))], col = NA), just = "top")
   if(v["significant"])       grid.rect(x, y - h*0.5 + 0.95:n/n*h, w*1, 1/n*h,   gp = gpar(fill = 
col[names(which(v))], col = NA), just = "top")

}
oncoPrint(data1, get_type = function(x) strsplit(x, ";")[[1]],
          alter_fun = alter_fun, col = col,   row_order = NULL,
          column_order = colnames(data1), show_column_names = TRUE,
          heatmap_legend_param = list(title = "Alternations",
                                      at = c("significant", "single-significant"),
labels = c("significant", "single-significant")))
```


module1
module6
module4
module9
module7
module8
module10
module3
module5
module2
IFIT5
IFITM1
IFIT2
IFI6
ISG20
ISG15
IFI35
IFIT3
HERC6
all
Alternations
🟩 significant
🟥 single-significant
no-significant