

```

#####
#####
## 2. Statistical methods for testing one pair of gene
#####
#####
library("GeneGeneInter")
data("gene.pair", package = "GeneGeneInter")
summary(gene.pair)

head(gene.pair$Y)

gene.pair$G1
gene.pair$G2

#####
## 2.1. Principal Component Analysis (PCA)
#####
PCA.test(Y = gene.pair$Y, G1 = gene.pair$G1, G2 = gene.pair$G2, threshold
= 0.7, method = "GenFreq")
PCA.test(Y = gene.pair$Y, G1 = gene.pair$G1, G2 = gene.pair$G2, threshold
= 0.7, method = "Std")

#####
## 2.2. Canonical Correlation Analysis (CCA)
#####
set.seed(1234)
CCA.test(Y = gene.pair$Y, G1 = gene.pair$G1, G2 = gene.pair$G2, n.boot =
500)

#####
## 2.3. Kernel Canonical Correlation Analysis (KCCA)
#####
set.seed(1234)
KCCA.test(Y = gene.pair$Y, G1 = gene.pair$G1, G2 = gene.pair$G2, kernel =
"rbfdot", sigma = 0.05, n.boot = 500)

set.seed(1234)
KCCA.test(Y = gene.pair$Y, G1 = gene.pair$G1, G2 = gene.pair$G2, kernel =
"besseldot", sigma = 0.05, degree = 1, order = 1, n.boot = 100)

#####
## 2.4. Partial Least Square Path Modeling (PLSPM)
#####
set.seed(1234)
PLSPM.test(Y = gene.pair$Y, G1 = gene.pair$G1, G2 = gene.pair$G2, n.perm
= 1000)

#####
## 2.5. Composite Linkage Disequilibrium (CLD)
#####
set.seed(1234)
CLD.test(Y = gene.pair$Y, G1 = gene.pair$G1, G2 = gene.pair$G2, n.perm =
2000)

#####
## 2.6. Gene-Based Information Gain Method (GBIGM)
#####
set.seed(1234)

```

```
GBIGM.test(Y = gene.pair$Y, G1 = gene.pair$G1, G2 = gene.pair$G2, n.perm
= 2000)
```

```
#####
```

```
## 2.7. From SNP-SNP interaction to Gene-Gene interaction testing
```

```
#####
```

```
## minP
```

```
set.seed(1234)
```

```
minP.test(Y = gene.pair$Y, G1 = gene.pair$G1, G2 = gene.pair$G2)
```

```
resp <- system.file(file.path("extdata", "response.txt"), package =
"GeneGeneInter")
```

```
Y <- read.csv(resp, header = FALSE, stringsAsFactors = TRUE)
```

```
load(system.file(file.path("extdata", "dataImputed.Rdata"), package =
"GeneGeneInter"))
```

```
PCSK6 <- selectSnps(data.imputed$snpX, data.imputed$genes.info,
"PCSK6")$snpX
```

```
TXNDC5 <- selectSnps(data.imputed$snpX, data.imputed$genes.info,
"TXNDC5")$snpX
```

```
ncol(PCSK6)
```

```
ncol(TXNDC5)
```

```
set.seed(1234)
```

```
minP.test(Y = Y, G1 = PCSK6, G2 = TXNDC5)
```

```
## GATES
```

```
set.seed(1234)
```

```
gates.test(Y = gene.pair$Y, G1 = gene.pair$G1, G2 = gene.pair$G2, me.est
= "ChevNy")
```

```
set.seed(1234)
```

```
gates.test(Y = gene.pair$Y, G1 = gene.pair$G1, G2 = gene.pair$G2, alpha =
0.05, me.est = "Keff")
```

```
set.seed(1234)
```

```
gates.test(Y = gene.pair$Y, G1 = gene.pair$G1, G2 = gene.pair$G2, me.est
= "LiJi")
```

```
set.seed(1234)
```

```
gates.test(Y = gene.pair$Y, G1 = gene.pair$G1, G2 = gene.pair$G2, me.est
= "Galwey")
```

```
## tTS and tProd
```

```
set.seed(1234)
```

```
tTS.test(Y = gene.pair$Y, G1 = gene.pair$G1, G2 = gene.pair$G2, tau =
0.5, n.sim = 10000)
```

```
set.seed(1234)
```

```
tProd.test(Y = gene.pair$Y, G1 = gene.pair$G1, G2 = gene.pair$G2, tau =
0.05, n.sim = 1000)
```

```
#####
```

```
#####
```

```
## 3. Analysis of a set of genes
```

```
#####
```

```
#####
```

```
#####
```

```
## 3.1. Genotype and phenotype importation
```

```

#####
## Genotype importation
ped <- system.file(file.path("extdata", "example.ped"), package =
"GeneGeneInterR")
info <- system.file(file.path("extdata", "example.info"), package =
"GeneGeneInterR")
posi <- system.file(file.path("extdata", "example.txt"), package =
"GeneGeneInterR")
data <- importFile(file = ped, snps = info, pos = posi, pos.sep = "")

summary(data)

data$snpX

summary(data$genes.info)

## Case-control status importation
Y <- read.table(system.file(file.path("extdata", "response.txt"), package =
"GeneGeneInterR"), sep = ";")

Y <- data$status

#####
## 3.2. Genotype data manipulation
#####
## Data filtering
data <- snpMatrixScour(data$snpX, genes.info = data$genes.info, min.maf =
0.05, min.eq = 1e-3, call.rate = 0.9)

data$snpX

selec <- selectSnps(data$snpX, data$genes.info, select = 1:10)
selec <- selectSnps(data$snpX, data$genes.info, c("DNAH9", "TXNDC5"))
selec <- selectSnps(data$snpX, data$genes.info,
c("15:101342000:101490000"))

## Genotype imputation
sum(is.na(data$snpX))

data <- imputeSnpMatrix(data$snpX, data$genes.info)

sum(is.na(data$snpX))

#####
## 3.3. Statistical analysis
#####
GGI.res <- GGI(Y = Y, snpX = data$snpX, genes.info = data$genes.info,
method = "PCA")
class(GGI.res)
names(GGI.res)

round(GGI.res$p.value[1:4, 1:4], digits = 4)

summary(GGI.res)

#####
## 3.4. Visualization
#####

```

```
## Heatmap-like visualization

#### Figure 1
plot(GGI.res)
plot(GGI.res, genes = c("CA1", "Gc", "PADI1"))

#### Figure 2
plot(GGI.res, col = c("black", "cyan", "white"), colbar.width = 0.25,
      title = "Interaction between 17 genes", hclust.order = TRUE, use.log
= TRUE,
      threshold = NULL, NA.col = "#D3D3D3", draw.pvals = FALSE, draw.names
= TRUE, interact = FALSE)

#### Figure 3
plot(GGI.res, col = c("black", "cyan", "white"), colbar.width = 0.05,
      title = "Interaction between 17 genes", hclust.order = TRUE, use.log
= FALSE,
      threshold = 0.05, NA.col = "#D3D3D3", draw.pvals = FALSE, draw.names
= TRUE, interact = FALSE)

## Network-like visualization
#### Figure 4
set.seed(1234)
plot(GGI.res, method = "network")

#### Figure 5
set.seed(1234)
plot(GGI.res, method = "network",
      genes = c("bub3", "CDSN", "Gc", "GLRX", "PADI1", "PADI2", "PADI4",
"PADI6", "PRKD3", "PSORS1C1", "SERPINA1", "SORBS1"),
      threshold = 0.05, plot.nointer = FALSE)
```