Brief Introduction to Genomic informed Drug Target database

Merina Shrestha Zhonghao Bai Astrid Johanneson Hjelholtz
Tahereh Gholipour Viktor Milkevych- Palle Duun Rohde
Mads Fuglsang Kjølby Peter Sørensen

2023-05-10

Contents

We have developed scripts and workflows that efficiently generate functional marker data sets from publicly available resources. Functional marker information has been downloaded and processed from:

- Ensembl (link SNPs to genes and proteins)
- GO (gene ontology)
- STRING (protein-protein)
- STITCH (protein-chemical)
- Reactome (biological pathways)
- more ressources will be added .....

Processing included quality control, mapping to LD reference panel and creation of marker sets (e.g., markers linked to genes, proteins, pathways) used in marker set analyses and subsequently be used to help the biological interpretation of genome-wide association studies.

This includes screening functional marker sets (e.g. biological pathways, protein complexes, gene ontology terms) for association with complex diseases.

It is also possible to test specific biological hypothesis such as:

Genes, proteins, metabolites, pathways underlying T2DM are enriched for association signal with T2DM

Drugs used for treatment of T2DM are linked to genes, proteins, metabolites, pathways enriched for association signal with T2DM

Our workflow allows us to quickly process new functional marker sets and we will therefore continue to identify and process functional marker data relevant for T2DM and other complex disease.

The practical is based on the R package `gact` (Rohde et al.2022)). This package provides an infrastructure for working with large-scale genomic association data linked to different types of genomic features.

Load packages used

The most recent version of `gact` can be obtained from github:

```
library(devtools)
devtools::install_github("psoerensen/gact")
```

```
library(gact)
library(qgg)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(data.table)
```

Download and install GDT database

The function `gact()` dowload and install the GDT database:

```
# Set working for database
dbdir <- "C:/Users/au223366/Dropbox/Projects/balder/gdtdb"

# Download the database from repository
GAlist <- gact(version = "gact-0.0.1", dbdir = dbdir, task = "download")
```

Study and feature information in the database

```
# Information about studies in GDT database
head(as.data.frame(GAlist$study))
```

```
##      id      file trait   type  gender      n ncase ncontrol     neff
## 1 GWAS1 GWAS1.txt  T2DM binary    both 456236 55927   400309 49071.27
## 2 GWAS2 GWAS2.txt   CAD binary    both 184305 60801   123504 40743.15
## 3 GWAS3 GWAS3.txt  T2DM binary    both 933970 80154   853816 73275.12
## 4 GWAS4 GWAS4.txt  T2DM binary   males 425662 41846   383816 37732.20
## 5 GWAS5 GWAS5.txt  T2DM binary females 464379 30053   434326 28108.07
## 6 GWAS6 GWAS6.txt  T2DM binary    both 898130 74124   824006 68006.44
##       reference                                        source
## 1 PMID:30297969 Mahajan.NatGenet2018b.T2D-noUKBB.European.txt
## 2 PMID:26343387                    CARDIoGRAMplusC4D.txt.gz
## 3 PMID:35551307                  DIAMANTE-EUR.sumstat.txt.gz
## 4 PMID:30297969   Mahajan.NatGenet2018b.T2D.MALE.European.txt
## 5 PMID:30297969 Mahajan.NatGenet2018b.T2D.FEMALE.European.txt
## 6 PMID:30297969        Mahajan.NatGenet2018b.T2D.European.txt
```

```
# Information about features in GDT database
GAlist$features
```

```
## [1] "Markers"           "Genes"             "Proteins"
## [4] "GO"                "Pathways"          "ProteinComplexes"
## [7] "ChemicalComplexes"
```

Extract data from GDT database

The function `getStatDB()` extract data from the database:

```
# Extract marker summary statistics GWAS1
stat <- getMarkerStatDB(GAlist = GAlist, studyID = "GWAS1")
```

```
## Extracting data from study: GWAS1
```

```
head(stat)
```

```
##         rsids chr    pos ea nea    eaf       b   seb    p        n
## 1 rs554127336   1 565469  T   C 0.0020  0.1200 0.210 0.58 49071.27
## 2   rs2000096   1 567867  G   A 0.0000 -0.5200 0.630 0.41 49071.27
## 3  rs12238997   1 693731  G   A 0.1300 -0.0088 0.017 0.60 49071.27
## 4  rs72631875   1 705882  A   G 0.0630  0.0110 0.037 0.76 49071.27
## 5  rs55727773   1 706368  A   G 0.5000  0.0140 0.015 0.37 49071.27
## 6   rs4565649   1 713092  A   G 0.0038  0.1200 0.230 0.61 49071.27
```

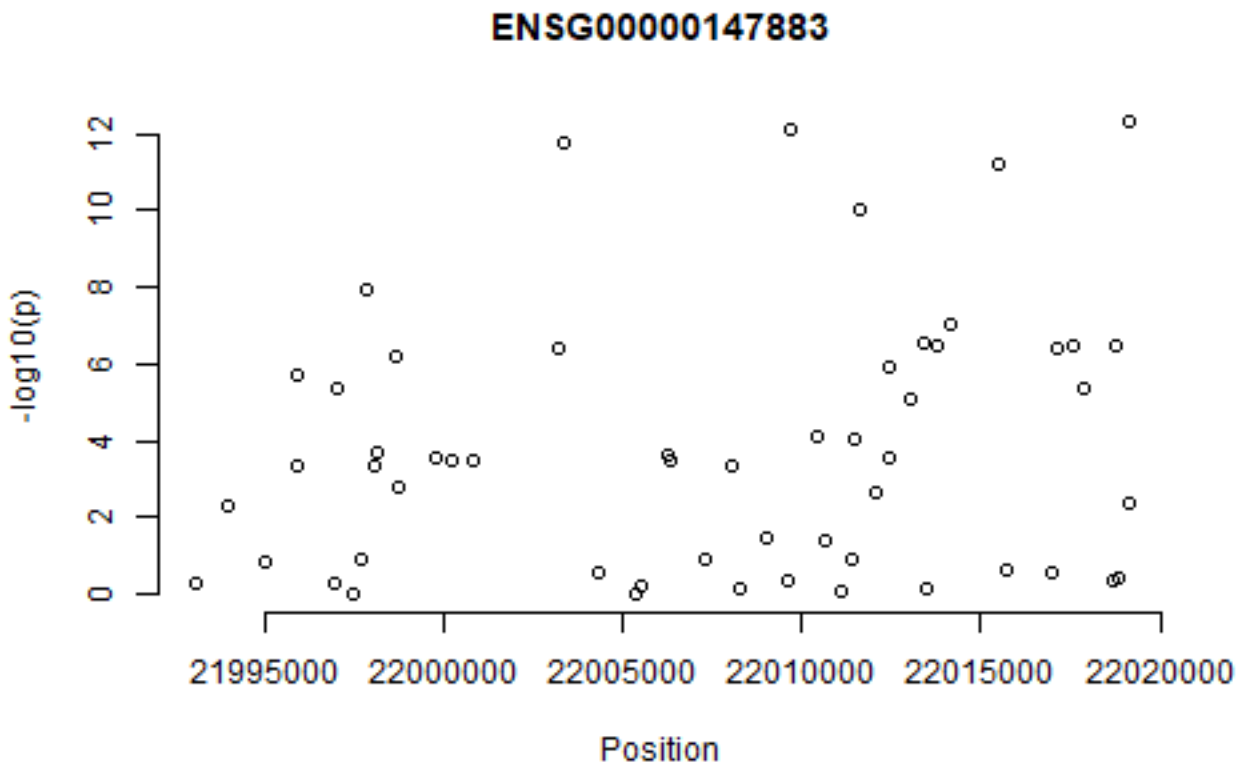
```
rownames(stat) <- stat$rsids

# Extract marker sets for gene ENSG00000147883
sets <- getSetsDB(GAlist = GAlist, feature = "Genes", featureID = "ENSG00000147883")
# sets is a list
str(sets)
```

```
## List of 1
##  $ ENSG00000147883: chr [1:62] "rs3731191" "rs2811711" "rs142723833" "rs55797833" ...
```

```
# convert sets into a vector
rsids <- sets[[1]]

# Plot -log10 p-values for selected gene
plot(y = -log10(stat[stat$rsids %in% rsids, "p"]), x = stat[stat$rsids %in%
    rsids, "pos"], ylab = "-log10(p)", xlab = "Position", frame.plot = FALSE,
    main = "ENSG00000147883")
```



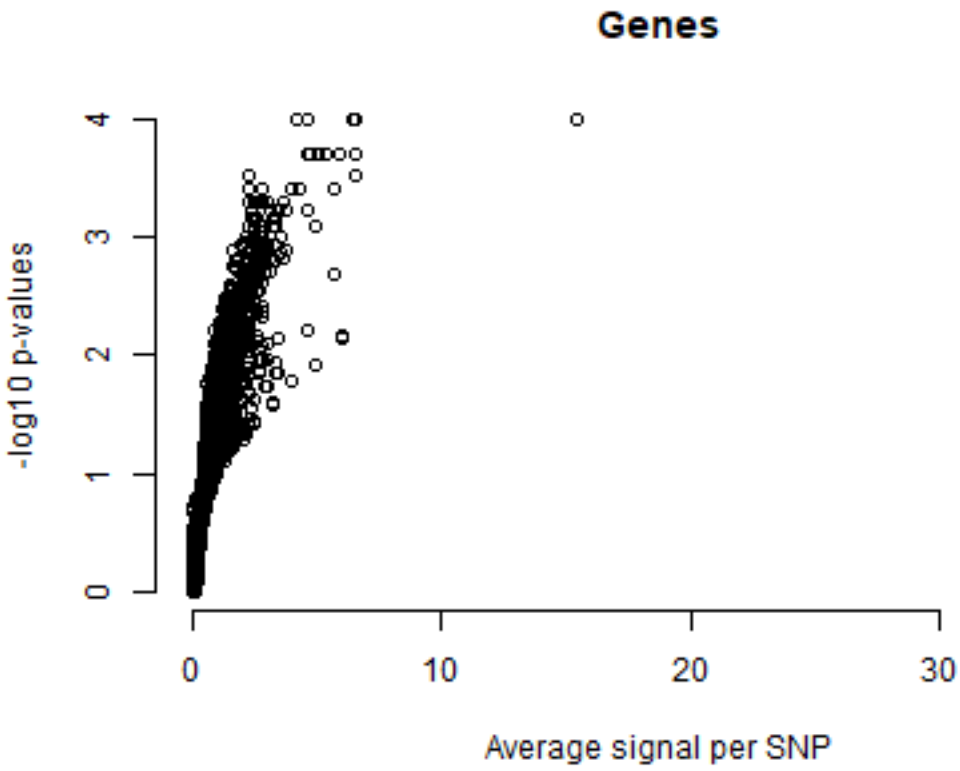

```
# Extract data for GWAS1 for genomic feature Genes
stat <- getStatDB(GAlist = GAlist, studyID = "GWAS1", feature = "Genes",
    format = "data.frame")
```

```
## Extract statistics based p-value threshold: 0.95
```

```
head(stat)
```

```
##                 Ensembl Gene ID   Symbol  m       stat      p
## ENSG00000121410 ENSG00000121410     A1BG 25 10.7469606 0.2364
## ENSG00000175899 ENSG00000175899      A2M 63 20.1668143 0.3656
## ENSG00000256069 ENSG00000256069    A2MP1 36  0.7684015 0.9592
## ENSG00000171428 ENSG00000171428     NAT1 60 14.5923539 0.5179
## ENSG00000156006 ENSG00000156006     NAT2 50 17.6043867 0.3116
## ENSG00000196136 ENSG00000196136 SERPINA3 16  3.8040525 0.4615
```

```
# Plot results
plot(x = stat$stat/stat$m, y = -log10(stat$p), ylab = "-log10 p-values",
    xlab = "Average signal per SNP", frame.plot = FALSE, main = "Genes")
```



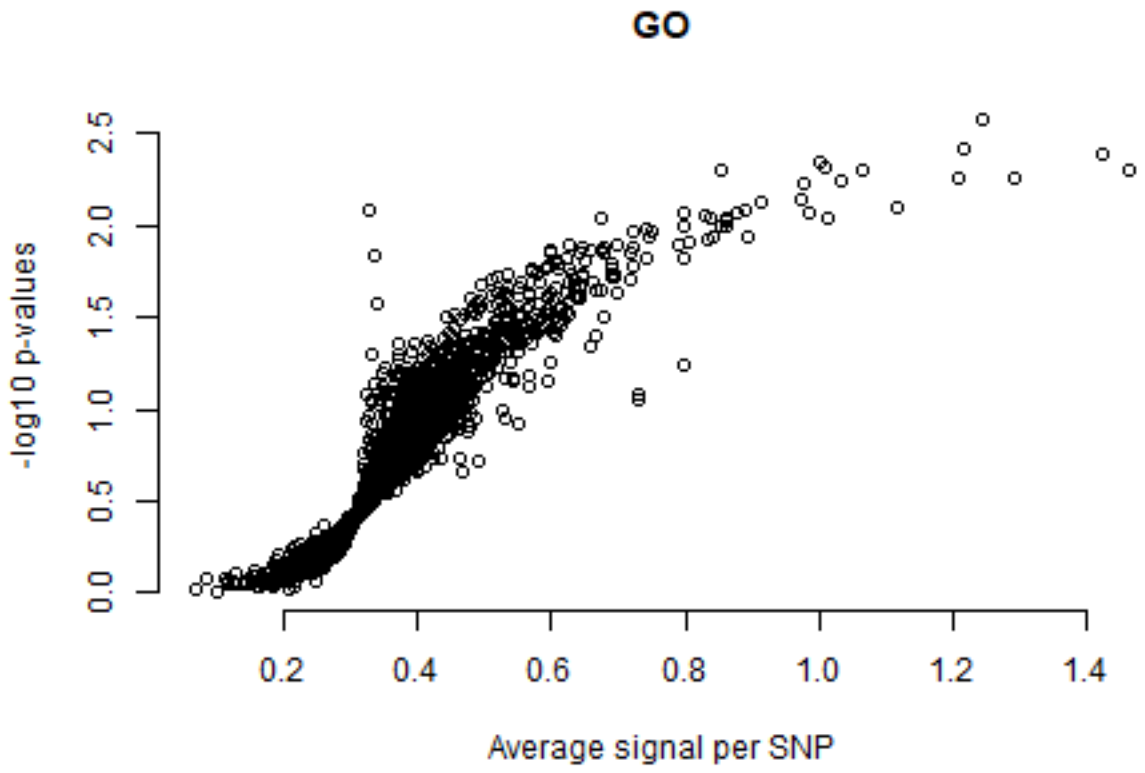

```
# Extract data for GWAS1 for genomic feature Gene Ontology
# (GO) where output format is a data frame
stat <- getStatDB(GAlist = GAlist, studyID = "GWAS1", feature = "GO")
```

```
## Extract statistics based p-value threshold: 0.95
```

```
str(stat)
```

```
## List of 3
##  $ m   : Named int [1:4624] 805 765 685 339 906 3213 377 1332 421 2010 ...
##   ..- attr(*, "names")= chr [1:4624] "GO:0000002" "GO:0000012" "GO:0000027" "GO:0000028" ...
##  $ stat: Named num [1:4624] 320 248 222 135 210 ...
##   ..- attr(*, "names")= chr [1:4624] "GO:0000002" "GO:0000012" "GO:0000027" "GO:0000028" ...
##  $ p   : Named num [1:4624] 0.145 0.326 0.327 0.176 0.703 ...
##   ..- attr(*, "names")= chr [1:4624] "GO:0000002" "GO:0000012" "GO:0000027" "GO:0000028" ...
```

```
plot(x = stat$stat/stat$m, y = -log10(stat$p), ylab = "-log10 p-values",
    xlab = "Average signal per SNP", frame.plot = FALSE, main = "GO")
```




```
# Extract data from T2D for genomic feature Gene Ontology
# (GO) where output format is list
stat <- getStatDB(GAlist = GAlist, feature = "GO")
```

```
## Extract statistics based p-value threshold: 0.95
```

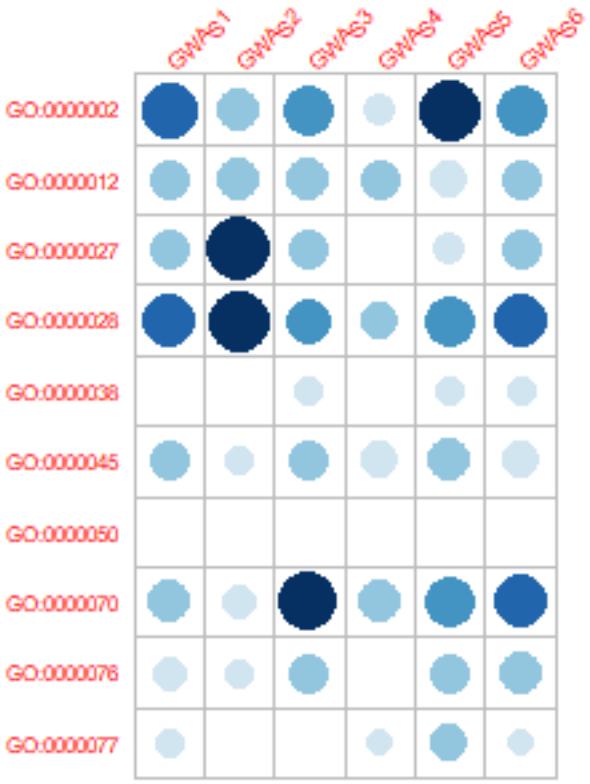
```
str(stat)
```

```
## List of 3
##  $ m   : Named int [1:4624] 805 765 685 339 906 3213 377 1332 421 2010 ...
##   ..- attr(*, "names")= chr [1:4624] "GO:0000002" "GO:0000012" "GO:0000027" "GO:0000028" ...
##  $ stat: num [1:4624, 1:6] 320 248 222 135 210 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4624] "GO:0000002" "GO:0000012" "GO:0000027" "GO:0000028" ...
##   .. ..$ : chr [1:6] "GWAS1" "GWAS2" "GWAS3" "GWAS4" ...
##  $ p   : num [1:4624, 1:6] 0.145 0.326 0.327 0.176 0.703 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4624] "GO:0000002" "GO:0000012" "GO:0000027" "GO:0000028" ...
##   .. ..$ : chr [1:6] "GWAS1" "GWAS2" "GWAS3" "GWAS4" ...
```

```
library(corrplot)

# Plot results for genomic feature Gene Ontology (GO)
colbar <- colorRampPalette(c("#FFFFFF", "#D1E5F0", "#92C5DE",
```

```
    "#4393C3", "#2166AC", "#053061"))
corrplot(-log10(stat$p[1:10, ]), is.corr = FALSE, tl.cex = 0.7,
    tl.srt = 45, col = colbar(6), cl.pos = "n", mar = c(1, 1,
        1, 1))
```




```
# Extract results from gsea of Reactome pathways
stat <- getStatDB(GAlist = GAlist, feature = "Genes", studyID = "GWAS1")
```

```
## Extract statistics based p-value threshold: 0.95
```

```
str(stat)
```

```
## List of 3
##  $ m   : Named int [1:26119] 25 63 36 60 50 16 33 30 20 7 ...
##   ..- attr(*, "names")= chr [1:26119] "ENSG00000121410" "ENSG00000175899" "ENSG00000256069" "ENSG0000
##  $ stat: Named num [1:26119] 10.747 20.167 0.768 14.592 17.604 ...
##   ..- attr(*, "names")= chr [1:26119] "ENSG00000121410" "ENSG00000175899" "ENSG00000256069" "ENSG0000
##  $ p   : Named num [1:26119] 0.236 0.366 0.959 0.518 0.312 ...
##   ..- attr(*, "names")= chr [1:26119] "ENSG00000121410" "ENSG00000175899" "ENSG00000256069" "ENSG0000
```

```
# Select the top pathways for GWAS1
featureID <- names(stat$p[stat$p < 0.01])
stat <- getStatDB(GAlist = GAlist, feature = "Genes", studyID = "GWAS1",
    featureID = featureID)
```

```
## Extract statistics based p-value threshold: 0.95
```

```
str(stat)
```

```
## List of 3
##  $ m   : Named int [1:26119] 25 63 36 60 50 16 33 30 20 7 ...
##   ..- attr(*, "names")= chr [1:26119] "ENSG00000121410" "ENSG00000175899" "ENSG00000256069" "ENSG0000
##  $ stat: Named num [1:503] 80.7 80.7 80.7 80.7 80.7 ...
##   ..- attr(*, "names")= chr [1:503] "ENSG00000204305" "ENSG00000231268" "ENSG00000229058" "ENSG000002
##  $ p   : Named num [1:503] 0.0024 0.0021 0.0027 0.0026 0.0028 0.0019 0.0012 0.0093 0.0047 0.0072 ...
##   ..- attr(*, "names")= chr [1:503] "ENSG00000204305" "ENSG00000231268" "ENSG00000229058" "ENSG000002
```

```
# Get marker stat for GWAS1 and GWAS2
stat <- getMarkerStatDB(GAlist = GAlist, studyID = c("GWAS1",
    "GWAS2"))
```

```
## Extracting data from study: GWAS1

## Extracting data from study: GWAS2
```

```
str(stat)
```

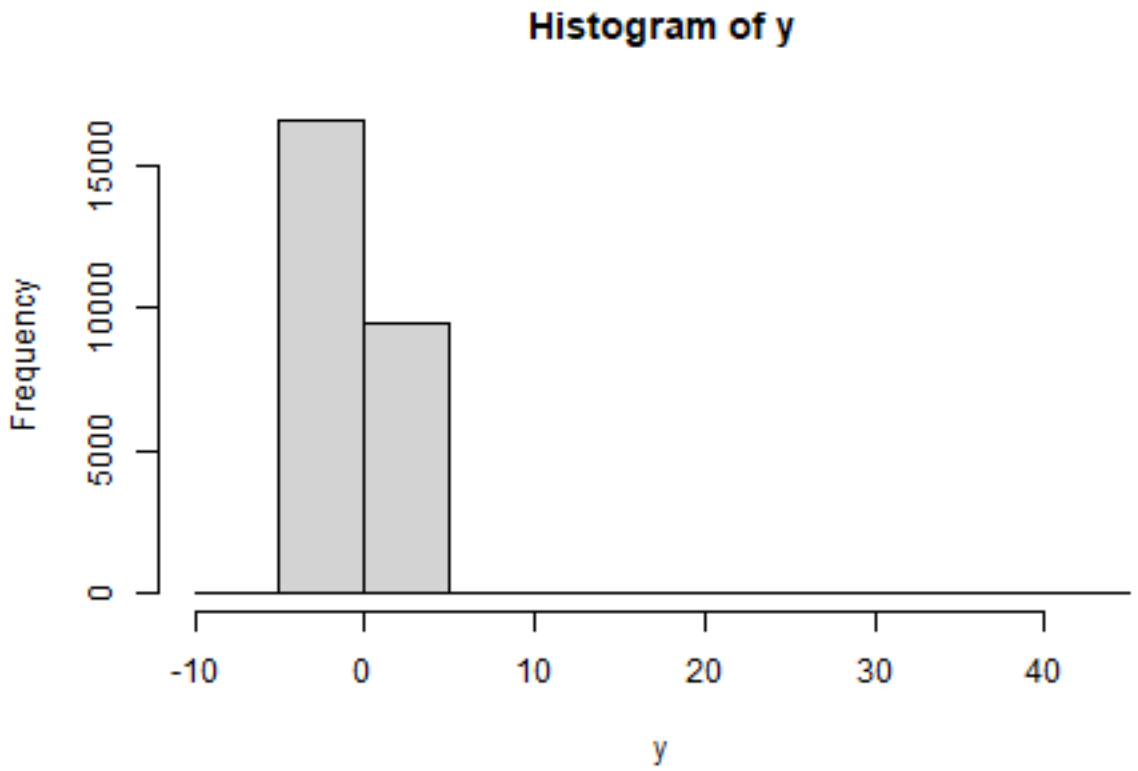
```
## List of 5
##  $ b  : num [1:7239752, 1:2] -0.0077 -0.012 -0.0053 -0.0053 -0.014 -0.0056 -0.0026 -0.0096 0.0041 -0
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:7239752] "rs61768174" "rs12562034" "rs60320384" "rs59066358" ...
##   .. ..$ : chr [1:2] "GWAS1" "GWAS2"
##   ..- attr(*, "na.action")= 'omit' Named num [1:370491] 1 2 3 7 9 14 15 34 35 36 ...
##   .. ..- attr(*, "names")= chr [1:370491] "rs143225517" "rs148989274" "rs7515915" "rs7518545" ...
##  $ seb: num [1:7239752, 1:2] 0.016 0.017 0.014 0.014 0.018 0.014 0.014 0.016 0.016 0.016 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:7239752] "rs61768174" "rs12562034" "rs60320384" "rs59066358" ...
##   .. ..$ : chr [1:2] "GWAS1" "GWAS2"
##   ..- attr(*, "na.action")= 'omit' Named num [1:370491] 1 2 3 7 9 14 15 34 35 36 ...
##   .. ..- attr(*, "names")= chr [1:370491] "rs143225517" "rs148989274" "rs7515915" "rs7518545" ...
##  $ z  : num [1:7239752, 1:2] -0.481 -0.706 -0.379 -0.379 -0.778 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:7239752] "rs61768174" "rs12562034" "rs60320384" "rs59066358" ...
##   .. ..$ : chr [1:2] "GWAS1" "GWAS2"
##   ..- attr(*, "na.action")= 'omit' Named num [1:370491] 1 2 3 7 9 14 15 34 35 36 ...
##   .. ..- attr(*, "names")= chr [1:370491] "rs143225517" "rs148989274" "rs7515915" "rs7518545" ...
##  $ p  : num [1:7239752, 1:2] 0.62 0.48 0.7 0.7 0.44 0.69 0.85 0.54 0.79 0.56 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:7239752] "rs61768174" "rs12562034" "rs60320384" "rs59066358" ...
##   .. ..$ : chr [1:2] "GWAS1" "GWAS2"
##   ..- attr(*, "na.action")= 'omit' Named num [1:370491] 1 2 3 7 9 14 15 34 35 36 ...
##   .. ..- attr(*, "names")= chr [1:370491] "rs143225517" "rs148989274" "rs7515915" "rs7518545" ...
##  $ n  : num [1:7239752, 1:2] 49071 49071 49071 49071 49071 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:7239752] "rs61768174" "rs12562034" "rs60320384" "rs59066358" ...
##   .. ..$ : chr [1:2] "GWAS1" "GWAS2"
##   ..- attr(*, "na.action")= 'omit' Named num [1:370491] 1 2 3 7 9 14 15 34 35 36 ...
##   .. ..- attr(*, "names")= chr [1:370491] "rs143225517" "rs148989274" "rs7515915" "rs7518545" ...
```

```
# Get gene-level association statistics (i.e. sum of )
stat <- getStatDB(GAlist = GAlist, feature = "Genes", threshold = 0.95,
    studyID = "GWAS1")
```

```
## Extract statistics based p-value threshold: 0.95
```

```
y <- scale(stat$stat)

# Adjust for gene length
y <- residuals(lm(y ~ stat$m))
hist(y)
```

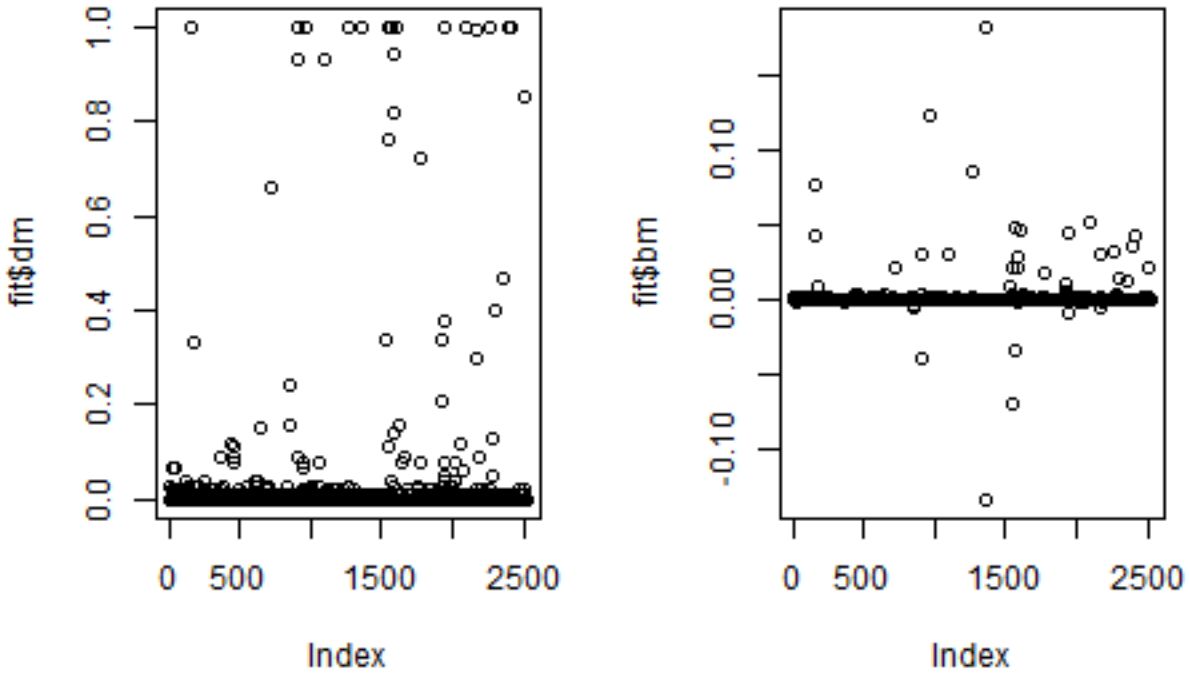



```
# Get design matrix for gene sets
W <- designMatrixDB(GAlist = GAlist, feature = "Pathways", rowFeatureID = names(y))
W <- scale(W)


# Fit BLR model
fit <- gbayes(y = y, W = W, method = "bayesC")
```

```
## Type of analysis performed: st-blr-individual-level-dense-ld
```

```
# Plot some BLR result
layout(matrix(1:2, ncol = 2))
plot(fit$dm)
plot(fit$bm)
```



Extract and write data from GDT database

The function `writeStat()` extract and write data from the database:

```
writeStatDB(GAlist = GAlist, feature = "GO", studyID = "GWAS1",
    file.csv = "go_GWAS1_gcta.csv")
writeStatDB(GAlist = GAlist, feature = "Pathways", studyID = "GWAS1",
    file.csv = "pathways_GWAS1_gcta.csv")
writeStatDB(GAlist = GAlist, feature = "ProteinComplexes", studyID = "GWAS1",
    file.csv = "proteincomplexes_GWAS1_gcta.csv")
writeStatDB(GAlist = GAlist, feature = "ChemicalComplexes", studyID = "GWAS1",
    file.csv = "chemicalcomplexes_GWAS1_gcta.csv")
writeStatDB(GAlist = GAlist, feature = "Genes", studyID = "GWAS1",
    file.csv = "genes_GWAS1_gcta.csv")
```

Extract marker set data from GDT database

The marker sets in the database can be extracted using:

```
geneSets <- getSetsDB(GAlist = GAlist, feature = "Genes")
chemSets <- getSetsDB(GAlist = GAlist, feature = "ChemicalComplexes2Genes")
```

Extract data for chemical "CIDm00004091" from GDT database

The marker sets in the database can be extracted using:

```
chemStat <- getStatDB(GAlist = GAlist, studyID = "GWAS1", feature = "ChemicalComplexes",
    format = "data.frame")
```

```
## Extract statistics based p-value threshold: 0.95
```

```
chemStat["CIDm00004091", ]
```

```
##                Chemical ID    m     stat      p
## CIDm00004091 CIDm00004091 4981 1713.073 0.2529
```

```
genesStat <- getStatDB(GAlist = GAlist, studyID = "GWAS1", feature = "Genes",
    format = "data.frame")
```

```
## Extract statistics based p-value threshold: 0.95
```

```
sets <- getSetsDB(GAlist = GAlist, feature = "ChemicalComplexes2Genes",
    featureID = "CIDm00004091")
# sets is a list
str(sets)
```

```
## List of 1
##  $ CIDm00004091: chr [1:153] "ENSG00000050344" "ENSG00000065970" "ENSG00000100448" "ENSG00000103121"
```

```
ensgIDs <- sets[[1]]
# convert sets into a vector
ensgIDs <- sets[[1]]
# select rsids in stat object
ensgIDs <- ensgIDs[ensgIDs %in% rownames(genesStat)]

head(genesStat[ensgIDs, ])
```

```
##                 Ensembl Gene ID Symbol   m      stat      p
## ENSG00000050344 ENSG00000050344 NFE2L3  40 20.784308 0.1450
## ENSG00000065970 ENSG00000065970  FOXJ2  29  6.420827 0.5209
## ENSG00000100448 ENSG00000100448   CTSG  10  4.874574 0.2070
## ENSG00000103121 ENSG00000103121   CMC2 140 32.162940 0.6005
## ENSG00000104899 ENSG00000104899    AMH  21  9.017324 0.2377
## ENSG00000104918 ENSG00000104918   RETN  16  7.913851 0.1957
```

Process external GWAS summary statistic data and add to database

library(data.table)

```
# Load GAlist
GAlist <- readRDS(file = file.path(dbdir, "GAlist_gact-0.0.1.rds"))

# Add Meta-analysis of BMI in UK Biobank and GIANT data.
# Combined set of samples, max N = 697,734.
fname_stat <- "https://portals.broadinstitute.org/collaboration/giant/images/1/14/Bmi.giant-ukbb.meta-a
stat <- fread(fname_stat, data.table = FALSE)
stat <- stat[, c("SNP", "CHR", "POS", "Tested_Allele", "Other_Allele",
    "Freq_Tested_Allele", "BETA", "SE", "P")]
colnames(stat) <- c("marker", "chr", "pos", "ea", "nea", "eaf",
    "b", "seb", "p")
head(stat)

GAlist <- updateStatDB(GAlist = GAlist, stat = stat, source = "bmi.giant-ukbb.meta-analysis.combined.23
    trait = "BMI", type = "quantitative", gender = "both", ancestry = "White",
    build = "GRCh37", reference = "PMID:30239722", n = 505454,
    ncase = 0, ncontrol = 0, comments = "Meta-analysis", writeStatDB = TRUE,
    excludeMAFDIFF = 0.05)

# Save updated GAlist
saveRDS(GAlist, file = file.path(dbdir, "GAlist_gact-0.0.1.rds"))
```

Shiny Apps to use genomic informed drug target database

These app provides an interface for conducting a range of gene set enrichment analyses and visualizing the results in various formats.The interface includes a title panel and a sidebar layout with input controls for selecting the study, entering gene symbols, and specifying analysis parameters. The main panel displays the results of the analysis in several tabs, including data tables, a Manhattan plot, and information about available studies.

```
# Load GAlist
GAlist <- readRDS(file = file.path(dbdir, "GAlist_gact-0.0.1.rds"))

# Shiny App for Gene Set Enrichment Analysis of Gene
# Complexes
shinyAppsDB(GAlist = GAlist, what = "customGSEA")

# Shiny App for Gene Set Enrichment Analysis of Genomic
# Features
shinyAppsDB(GAlist = GAlist, what = "featureGSEA")

# Shiny App for Drug Target Analysis of a Specific Genomic
# Features
shinyAppsDB(GAlist = GAlist, what = "featureDRUGS")
```