

# Introduction to the gact R package

From integrative genomics to polygenic risk scoring

Palle Duun Rohde

[palledr@hst.aau.dk](mailto:palledr@hst.aau.dk)

Genomic Medicine

Aalborg University

Peter Sørensen

[pso@qgg.au.dk](mailto:pso@qgg.au.dk)

Center for Quantitative Genetics and Genomics

Aarhus University



<https://pdrohde.github.io/>



## Section 1

### Setting up *gact* the first time

# Install *gact*

Install *gact* from our GitHub (<https://psoerensen.github.io/gact/>)

```
# Prepare gact
# only do once!
library(gact)

GAlist <- gact(version="hsa.0.0.1", dbdir="..../gact", task="download")

# Download 1000G reference data
GAlist <- downloadDB(GAlist=GAlist, what="1000G")

saveRDS(GAlist, file="..../gact/hsa.0.0.1/GAlist_hsa.0.0.1.rds")

> names(GAlist)
[1] "version"      "dbdir"        "dirs"         "features"     "markerfiles"
[6] "rsids"        "cptra"        "gsetsfiles"  "gseafiles"   "study"
[11] "studies"      "studyfiles"  "gwasfiles"    "gsets"       "targets"
[16] "drug2atc"    "atc"
```

hsa.0.0.2
> download
> drugdb
> gbayes
> glist
> gsea
> gsets
> gstat
> gtex
> gwas
> ldsc
> marker
> script
> vep

# Included annotations

List of 10

```
$ ENSG00000223972: chr [1:18] "rs575272151" "rs544419019" "rs540538026" "rs62635286" ...
$ ENSG00000227232: chr [1:19] "rs575272151" "rs544419019" "rs540538026" "rs62635286" ...
$ ENSG00000243485: chr [1:19] "rs575272151" "rs544419019" "rs540538026" "rs62635286" ...
$ ENSG00000237613: chr [1:19] "rs575272151" "rs544419019" "rs540538026" "rs62635286" ...
$ ENSG00000268020: chr [1:29] "rs564023708" "rs533090414" "rs806731" "rs542415070" ...
$ ENSG00000186092: chr [1:35] "rs542415070" "rs559500163" "rs528344458" "rs551668143" ...
$ ENSG00000241670: chr [1:6] "rs201347561" "rs144169752" "rs112455420" "rs141415251" ...
$ ENSG00000237094: chr [1:3] "rs576317820" "rs541569731" "rs182870673"
$ ENSG00000235249: chr [1:2] "rs541569731" "rs182870673"
$ ENSG00000185097: chr [1:8] "rs561532399" "rs113167131" "rs111824286" "rs369953380" ...
```

List of 10

```
$ R-HSA-1059683: chr [1:141] "ENSG00000096968" "ENSG00000096968" "ENSG00000105397" "ENSG00000105397" ...
$ R-HSA-109704 : chr [1:249] "ENSG0000033327" "ENSG0000051382" "ENSG0000066468" "ENSG0000068078" ...
$ R-HSA-110056 : chr [1:74] "ENSG0000096968" "ENSG00000102882" "ENSG00000105397" "ENSG00000134352" ...
$ R-HSA-110312 : chr [1:108] "ENSG0000009413" "ENSG00000035928" "ENSG0000049541" "ENSG00000106399" ...
$ R-HSA-110314 : chr [1:190] "ENSG00000035928" "ENSG0000049541" "ENSG0000062822" "ENSG0000070950" ...
$ R-HSA-110320 : chr [1:125] "ENSG00000010072" "ENSG00000035928" "ENSG0000049541" "ENSG0000070010" ...
$ R-HSA-110328 : chr [1:303] "ENSG00000065057" "ENSG00000076248" "ENSG0000092330" "ENSG00000102977" ...
$ R-HSA-110329 : chr [1:303] "ENSG00000065057" "ENSG00000076248" "ENSG0000092330" "ENSG00000102977" ...
$ R-HSA-110330 : chr [1:497] "ENSG00000092330" "ENSG00000092330" "ENSG00000102977" "ENSG00000102977" ...
$ R-HSA-110331 : chr [1:497] "ENSG00000092330" "ENSG00000092330" "ENSG00000102977" "ENSG00000102977" ...
```

# Ingest GWAS summary data

```
GAlist <- updateStatDB(GAlist = GAlist,
                      stat = stat,                                # R-object
                      source = "...",                            # markers file name
                      trait = "...",                             # name of trait
                      type = "...",                             # quantitative or binary
                      gender = "...",                           # male, female, or both
                      ancestry = "...",                          # ancestry
                      build = "...",                            # genome build, GRCh37 or GRCh38
                      reference = "...",                         # PMID
                      n = NA,                                   # total sample size
                      ncase = NA,                               # if binary, how many cases
                      ncontrol = NA,                            # if binary, how many controls
                      comments = "...")                         # comments
)
```

# Ingest GWAS summary data, example

```
# Load GWAS data
fname_stat <- "./Mahajan.NatGenet2018b.T2D-noUKBB.European.zip"
stat <- fread(fname_stat, data.table = FALSE)

# Modify columns according to required format
stat <- stat[, c("SNP", "Chr", "Pos", "EA", "NEA", "Beta", "SE", "Pvalue")]
colnames(stat) <- c("marker", "chr", "pos", "ea", "nea", "b", "seb", "p")

# Update database
GAlist <- updateStatDB(GAlist = GAlist,
                        stat = stat,
                        source = "Mahajan.NatGenet2018b.T2D-noUKBB.European.zip",
                        trait = "T2D",
                        type = "binary",
                        gender = "both",
                        ancestry = "EUR",
                        build = "GRCh37",
                        reference = "PMID:30297969",
                        n = 456236,
                        ncase = 55927,
                        ncontrol = 400309,
                        comments = "Exclude UK biobank",
                        writeStatDB = TRUE)

# Save updated database
saveRDS(GAlist, file = "./gact/hsa.0.0.1/GAlist_hsa.0.0.1.rds", compress = FALSE)
```

# Collecting information and performing QC

Collecting information on external summary statistics

Perform quality control of external summary statistics

Map markers based on cpca

Number of markers in stat mapped to marker ids in GAlist: 9060453

Number of markers in stat not mapped to marker ids in GAlist: 12448245

Number of markers in stat also found in bimfiles: 9060453

Number of effect alleles aligned with first allele in bimfiles: 5056308

Number of effect alleles not aligned with first allele in bimfiles: 4004145

Number of markers excluded by large difference between MAF difference: 621

Writing processed summary statistics to internal file: GWAS1.txt

# gstat

## gstat

GWAS\_information.csv

GWAS1.txt.gz

GWAS2.txt.gz

GWAS3.txt.gz

GWAS4.txt.gz

id	file	trait	type	gender	n	ncase	ncontrol	neff	reference	source	ancestry	build	comments	
GWAS1	GWAS1.txt	T2DM	binary	both	456236	55927	400309	49071.27329	PMID:30297969	Mahajan.NatGenet2018b.T2D-noUKBB.Eur	EUR	GRCh37	Exclude UK biobank	
GWAS2	GWAS2.txt	CAD	binary	both	184305	60801	123504	40743.1524	PMID:26343387	CARDIoGRAMplusC4D.txt.gz		EUR	GRCh37	Exclude UK biobank
GWAS3	GWAS3.txt	T2DM	binary	both	933970	80154	853816	73275.12411	PMID:35551307	DIAMANTE-EUR.sumstat.txt.gz		EUR	GRCh37	Include UK biobank
GWAS4	GWAS4.txt	T2DM	binary	males	425662	41846	383816	37732.20146	PMID:30297969	Mahajan.NatGenet2018b.T2D.MALE.Europ	EUR	GRCh37	Include UK biobank	

## Section 2

# Example analyses

# Estimating genetic parameters - 1

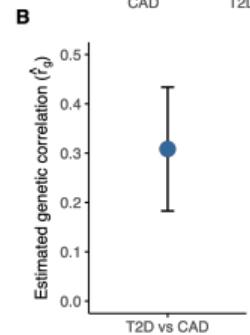
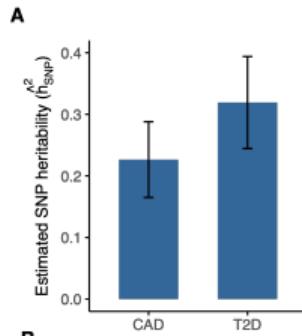
```
# Load GAlist
GAlist <- readRDS(file="./gact/hsa.0.0.1/GAlist_hsa.0.0.1.rds")

# Select GWAS study IDs
studyIDs <- c("GWAS1", "GWAS2")

# Get GWAS summary statistics for studyIDs (e.g. z and n) from gact database
stat <- getMarkerStat(GAlist=GAlist, studyID=studyIDs)

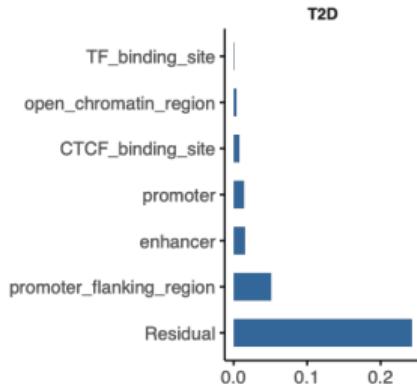
# Get ldsc scores matched to the ancestry of GWAS data
ldscscores <- getLDscoresDB(GAlist=GAlist, ancestry="EUR", version="1000G")

# Estimate heritability and genetic correlation using ldsc
fit.h2 <- ldsc(z=stat$z, n=stat$n, ldscscores=ldscscores, what="h2", SE.h2=T)
fit.rg <- ldsc(z=stat$z, n=stat$n, ldscscores=ldscscores, what="rg", SE.rg=T)
```



# Estimating genetic parameters - 2

```
# Partitioned h2 across regulatory categories
sets <- getMarkerSets(GAlist=GAlist, feature="Regulatory Categories")
fit <- ldsc(z=stat$z, n=stat$n, ldscores=ldscores, sets=sets, what="h2",
            method="bayesC", residual=TRUE)
```



# Gene-level association (VEGAS)

```
# Load Glist with information on 1000G matched to the ancestry of GWAS data
GAlist <- readRDS(file=".~/hsa.0.0.1/GAlist_hsa.0.0.1.rds")
Glist <- readRDS(file.path(GAlist$dirs["marker"],"Glist_1000G_eur_filtered.rds"))

# Extract gene-marker sets (include markers 40kb/10kb upstream/downstream)
markerSets <- getMarkerSets(GAlist = GAlist, feature = "Genesplus")

# Select study1
studyID <- "GWAS1"

# Get GWAS summary statistics from gact database
stat <- getMarkerStat(GAlist=GAlist, studyID=studyID)

# Check and align summary statistics based on marker information in Glist
stat <- checkStat(Glist=Glist, stat=stat)

# Gene analysis using VEGAS
res <- vegas(Glist=Glist, sets=markerSets, stat=stat, verbose=TRUE)
```

EnsemblID	Chr	m	X2	z	p
ENSG00000237491	1	26	12.16659	-0.6649650	0.7375987
ENSG00000177757	1	33	12.92801	-0.6622818	0.7355755
ENSG00000225880	1	37	13.57057	-0.6788080	0.7480649
ENSG00000230368	1	50	22.41127	-0.7223630	0.7812910

# Gene-set enrichment analysis (MAGMA)

```
# Get gene sets
sets <- getFeatureSets(GAlist = GAlist, feature="Pathways", minsets=25)

# Get VEGAS results
z <- getVEGAS(GAlist,studyID="GWAS1")

# Run standard MAGMA
fitST.m <- magma(stat=z, sets=sets, method="marginal")
fitST.j <- magma(stat=z, sets=sets, method="joint")

# Run Bayesian MAGMA
fitST.b <- magma(stat=z, sets=sets, method="bayesC", pi=0.01, nit=10000, nburn=1000)

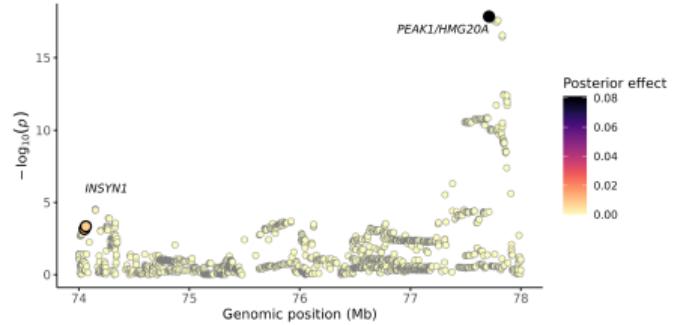
# Run Bayesian multi-trait MAGMA
z <- getVEGAS(GAlist,studyID=c("GWAS1","GWAS2"))
fitMT.b <- magma(stat=z, sets=sets, method="bayesC", pi=0.01, nit=10000, nburn=1000)
```

# Statistical Fine Mapping

```
res <- vector(22, mode="list")

for(CHR in 1:22){
  ldscores <- Glist$ldscores[[CHR]]
  ldscores <- ldscores[names(ldscores) %in% stat$rsids]
  sets <- createLDsets(ldscores=ldscores,
                        maxsize=3000, msize=100, verbose=TRUE)

  fit <- gmap(Glist=Glist, stat=stat, sets=sets,
              method="bayesR", algorithm="mcmc-eigen",
              nit=10000, nburn=1000, cs_threshold=0.5, cs_r2=0.5,
              eigen_threshold=c(0.99, 0.97, 0.95, 0.9), verbose=TRUE)
  res[[CHR]] <- fit
}
```



# Polygenic scoring

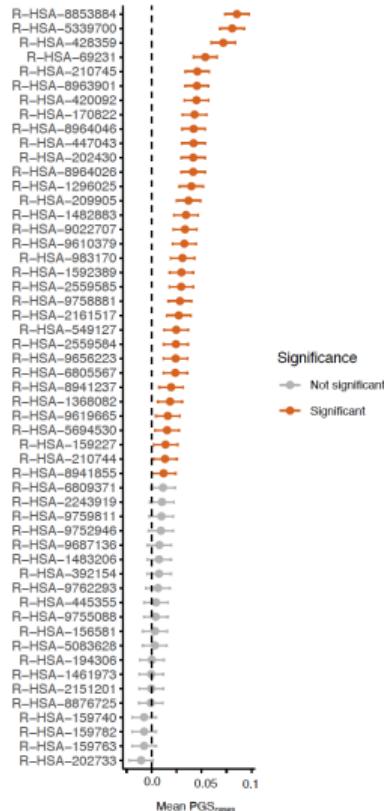
```
# Finemap based on 1kg
stat <- readRDS("./finemap_gwas1.rds")

# Load Glist for UKB
Glist <- readRDS("./Glist_eur_ukb.rds")

# Match GWAS effects with allele effects in UKB
stat <- checkStat(Glist=Glist, stat=stat,
                   excludeMAF=0.05,
                   excludeMAFDIFF=0.05,
                   excludeINFO=0.8,
                   excludeCGAT=TRUE,
                   excludeINDEL=TRUE,
                   excludeDUPS=TRUE,
                   excludeMHC=FALSE,
                   excludeMISS=0.05,
                   excludeHWE=1e-12)

# Compute PGS
pgs <- gscore(Glist=Glist, stat=stat)

# Pathway-specific scores
sets <- getMarkerSets(GAlist = GAlist, feature="Pathways")
pgs.pathway <- gscore(Glist=Glist, stat=stat, sets=sets)
```



# Visit our poster tomorrow

**A versatile data repository for GWAS summary statistics-based downstream genomic analysis of human complex traits**

Peter Sørensen<sup>1</sup> & Palle Duun Rohde<sup>2</sup>

<sup>1</sup> Centre for Quantitative Genetics and Genomics, Aarhus University, Denmark  
<sup>2</sup> Genomic Medicine, Department of Health Science and Technology, Aalborg University, Denmark

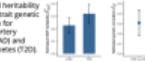
### BACKGROUND & MOTIVATION

- Background**
  - Downstream GWAS analyses often rely on fragmented and non-reproducible pipelines
  - Limited cross-study consistency and biological interpretation
- Challenge**
  - Inconsistent data formats and incomplete quality control (QC) procedures
  - Limited integration between association results and functional or biological annotations
- Solution - part (Genomic Association of Complex Traits)**
  - Open-source, modular R framework for standardized downstream GWAS
  - Provides harmonized data structures and reference resources
  - Integrates tools for fine-mapping, heritability estimation, gene set enrichment, and polygenic scoring
  - Enables reproducible and biologically informed interpretation of complex traits

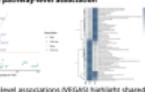


### DEMONSTRATION

Estimation of genetic parameters



Gene- and pathway-level association



Statistical fine mapping



Polygenic prediction



Impact and outlook

- Open-source R package for standardized downstream GWAS analyses
- Enables large-scale, reproducible, and biologically informed genomic analyses
- Facilitates integration of association results with biological and functional annotations
- Extends analysis beyond GWAS to additional molecular layers (transcriptome, epigenome, proteome) and cross-trait integration

**THE GACT FRAMEWORK**  
The gact package consists of three integrated modules that together provide a complete workflow for downstream GWAS analyses.

- 1 GWAS Summary Data Integration & QC**
  - Import, harmonize, and clean GWAS summary data
  - Standardize formats and metadata for downstream analyses
- 2 Annotation & Linking**
  - Map SNPs to genes, pathways, drug targets, and tissues
  - Integrate functional annotations and gene expression resources
- 3 Analysis**
  - Perform large-scale statistical genetic analyses, including:
    - Gene-level association and pathway enrichment
    - Fine-mapping and polygenic scoring
    - Estimate population genetic parameters from GWAS summary data

**Integration**  
Fully compatible with egg, enabling efficient large-scale statistical genetic analyses in R.  
(Rohde et al., Bioinformatics 2019 & 2020)



**Palle Duun Rohde**  
Associate Professor  
+45 8716 1616  
+45 2020 3000  

QR code: 

## Section 3

# Discussion - jigsaw

# Initiating the discussion

