SUMSTATS

Description of GWAS summary statistics

The repository shows how to provide input for PW-pipeline, FM-pipeline including GCTA, among others.

The summary table

Briefly, the format has the following columns,

Column	Name	Description
1	SNP	RSid
2	A1	Effect allele
3	A2	Other allele
4	freqA1	A1 frequency
5	beta	effect estimate
6	se	standard error of effect
7	P	P-value
8	N	sample size
9*	chr	chromosome
10*	pos	position

* see next section.

SNP information

The chromosomes and positions can be obtained from https://genome.ucsc.edu/, which should be helpful for GWAS summary statistics either using chromosomal positions from different builds or without these at all.

```
wget http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/snp150.txt.gz
gunzip -c snp150.txt.gz | \
awk '{split($2,a,"_");sub(/chr/,"",a[1]);print a[1],$4,$5}' | \
sort -k3,3 > snp150.txt
```

where it first obtains build 37 positions, sorts them by RSid into the file `snp150.txt`. More flexiblly, we can do as in TWAS-pipeline on refGene by selecting appropriate columns,

```
# from the MySQL database
mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A -D hg19 -e 'select *
from snp150' > snp150.txt
# into a MySQL database
gunzip -c snp150.txt.gz > snp150.txt
(
  wget -qO-
http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/snp150.sql
  echo load data local infile 'snp150.txt' into table snp150;
) > snp150.sql
```

while snp150.sql is amended. The option -A on the MysQL command line makes it faster and -D specifies database. More generally, it is possible to use MySQL interactively,

```
mysql --user=genome --host=genome-mysql.cse.ucsc.edu -A <<END
show databases;
use hg19;
show tables;
describe snp150;
select * from snp150 LIMIT 1,30;
select chrom, chromStart, chromEnd, strand, name, refNCBI, observed from
snp150 WHERE name="rs548419688";
END
```

which illustrate some useful commands.

Lastly, it may be useful to generate a rsid – snpid (chromosome:position_allele1_allele2 such that allele1 < allele2) linkage file. Assuming that our download is in compressed format, this can be achieved as follows,

```
gunzip -c snp150.txt.gz | \
cut -f2,4,5,10 | \
awk '/^chr[0-9]$|^chr[0-9][0-9]$|chrX|chrY/{if(!index($4,"-"))
{split($4,a,"/"); print $1 ":" $2 "_" a[1] "_" a[2], $3}}' | \
sort -k1,1 | \
gzip -f > snp150.snpid_rsid.gz
```

Besides standard chromosomal positions, hg38 reference genome assembly also has other categories[1],

- **Random contigs** (e.g., chrY_KI270740v1_random). Unlocalized sequences that are known to originate from specific chromosomes, but whose exact location within the chromosomes is not known (e.g., chr9_KI270720v1_random).
- **ChrUn** (e.g., chrUn_GL000218v1). Unplaced sequences that are known to originate from the human genome, but which cannot be confidently placed on a specific chromosome.
- **EBV** (e.g., -). Epstein-Barr virus sequence, representing the genome of Human herpes virus 4 type 1, the cause of infectious mononucleosis. This disease results in fever, sore throat, and enlarged cervical lymph nodes. About 98% of adults have been exposed to the virus by the age of 40 years. Since the virus remains latent in the body after infection, it is very common to find EBV sequences when performing human genome sequencing.
- **HLA** (e.g., HLA-C*01:08). Human leukocyte antigen (HLA) sequences representing selected alleles of the HLA A, B, C, DQA1, DQB1, and DRB1 loci. The HLA loci encode the major histocompatibility complex (MHC) proteins and are highly variable in the population.
- **Decoy sequences** (e.g., chrUn_KN707606v1_decoy). A major motivation for including the " sequences in the reference genome is that if a sample actually contains a genomic segment that is not in the reference assembly, aligners may spend a lot of CPU time trying to find a good match, or worse, aligners may wrongly assign the reads with a low mapping quality to segments with similar sequences in the reference genome. If the segment matches to one of the decoy contigs and the decoy is included in the reference, then the segment will quickly be assigned to the decoy and prevent the aligner from uselessly searching for other

matches. Thus, the decoy sequences are a pragmatic solution to this, contain EBV and human sequences that in effect " reads that would otherwise map with low quality to other regions in the reference and lead to unnecessary computation and avoid false variant calls related to false mappings.

- **Alternate contigs** (e.g., chr3_KI270778v1_alt). Alternative sequence paths in regions with complex structural variation in the form of additional locus sequences.

It might be worthwhile to check for options with the sumstats as defined in ldsc, https://github.com/bulik/ldsc, and particularly its munge_sumstats.py utility.

lz.sh is a script which extracts information on SNP and their positions from LocusZoom 1.4 database.

Examples

BMI

We take data reprted by Locke, et al. (2015) as example which requires build 37 positions from UCSC described above.

```
# GWAS summary statistics
wget
http://portals.broadinstitute.org/collaboration/giant/images/1/15/SNP_gwas_mc
_merge_nogc.tbl.uniq.gz
gunzip -c SNP_gwas_mc_merge_nogc.tbl.uniq.gz |
awk 'NR>1' | \
sort -k1,1 | \
join -11 -23 - snp150.txt | \
awk '($9!="X" && $9!="Y" && $9!="Un")' | \
gzip -f > bmi.tsv.gz
```

where file containing the GWAS summary statistics is downloaded, its header dropped, sorted and positional information added leading to a compressed file named bmi.tsv.gz. We also filter out nonautosomal SNPs.

The list of 97 SNPs can be extracted as follows,

```
R --no-save <<END
library(openxlsx)
xlsx <- "https://www.nature.com/nature/journal/v518/n7538/extref/nature14177-
s2.xlsx"
snps <- read.xlsx(xlsx, sheet = 4, colNames=FALSE, skipEmptyRows = FALSE,
cols = 1, rows = 5:101)
snplist <- sort(as.vector(snps[,1]))
write.table(snplist, file="97.snps", row.names=FALSE, col.names=FALSE,
quote=FALSE)
END
```

The list is SNPs is contained in 97.snps.

As described elsewhere, we are rather tempted to use a distance-based approach for independent signals,

```
(
  awk -vOFS="\t" 'BEGIN{print "MarkerName", "A1", "A2", "freq", "Effect",
"StdErr", "P.value", "N", "Chrom", "End"}'
  zcat bmi.tsv.gz | \
  sed 's/ /\t/g' | \
  awk '(NR>1 && $7<=2.4e-7)' | \
  sort -k9,9n -k10,10n
) > bmi.dat
R --no-save -q < bmi.R > bmi.out
grep  -f 97.snps bmi.out | \
wc -l
```

with `bmi.R` as follows,

```
options(echo=FALSE)
bmi <- read.delim("bmi.dat",as.is=TRUE)
require(reshape)
require(gap)
chrs <- with(bmi,unique(Chrom))
for(chr in chrs)
{
#  print(chr)
  ps <-
subset(bmi[c("Chrom","End","MarkerName","Effect","StdErr","P.value")],Chrom==
chr)
  row.names(ps) <- 1:nrow(ps)
  sentinels(ps,chr,1)
}
```

and the output is bmi.out. One may wonder the overlap between the two, and the answer is 69.

T2D

The data was reported by Scott, et al. (2017),

```
R -q --no-save <<END

library(openxlsx)
library(dplyr)

xlsx <-
"http://diabetes.diabetesjournals.org/highwire/filestream/79037/field_highwir
e_adjunct_files/1/DB161253SupplementaryData2.xlsx"

# Supplementary Table 3. Results for established, novel and additional
distinct signals from the main analysis.
ST3 <- read.xlsx(xlsx, sheet = 3, colNames=TRUE, skipEmptyRows = FALSE, cols
= 1:20, rows = 2:130) %>%
       rename(P="p-value.in.stage.1") %>% within(
       {
          beta=log(OR)
          L <- as.numeric(substr(CI,1,4))
          U <- as.numeric(substr(CI,6,9))
          se=abs(log(L)-log(U))/3.92
       }) %>% select(
          SNP=rsid,
          A1=EA,
          A2=NEA,
          freqA1=EAF,
          beta,
          se,
          P,
          N=Sample.size,
          chr=Chr,
          pos=Position_b37
       )
write.table(ST3, file="ST3", row.names=FALSE, col.names=FALSE, quote=FALSE)

# Supplementary Table 4. BMI-unadjusted association analysis model
ST4 <- read.xlsx(xlsx, sheet = 4, colNames=TRUE, skipEmptyRows = FALSE, cols
= 1:12, rows = 3:132) %>% rename(
          "CI"="CI.95%",
          "P"="P-value") %>% within(
       {
          beta=log(OR)
          L <- as.numeric(substr(CI,1,4))
          U <- as.numeric(substr(CI,6,9))
          se=abs(log(L)-log(U))/3.92
          P=2*(1-pnorm(abs(beta/se)))
       }) %>% select(
          SNP=rsid,
          A1=allele1,
          A2=allele2,
          freqA1=freq1,
          beta,
          se,
          P,
          N,
          chr,
          pos=position_b37
       )
write.table(ST4, file="ST4", row.names=FALSE, col.names=FALSE, quote=FALSE)

END
```

where we generate data based on the paper's supplementary tables ST3 and ST4; the former is in line with the paper (by specifying _db=depict and p_threshold=0.00001 when calling PW-pipeline).

Plasma proteins

The data was reported by Sun, et al. (2018),

```
require(openxlsx)
xlsx <- "https://static-content.springer.com/esm/art%3A10.1038%2Fs41586-018-
0175-2/MediaObjects/41586_2018_175_MOESM4_ESM.xlsx"
# Supplementary Table 4
ST4 <- read.xlsx(xlsx, sheet=4, colNames=TRUE, skipEmptyRows=FALSE,
cols=c(6:8,11:13,23:25), rows=6:1986)
names(ST4) <- c("SNP","Chr","Pos","A1","A2","EAF","b","se","p")
write.table(ST4, file="plamsprotein", row.names=FALSE, col.names=FALSE,
quote=FALSE)
```

and Supplementary Table 4 is fetched here.

Related resource

A repository with specific focus on download of sumstats is as follows,

https://github.com/mikegloudemans/gwas-download

References

GIANT (Genetic Investigation of ANthropometric Traits) data

Locke AE, et al. (2015) Genetic studies of body mass index yield new insights for obesity biology. *Nature* 518(7538):197-206. doi: 10.1038/nature14177

DIAGRAM (DIAbetes Genetics Replication And Meta-analysis) data

Scott R, et al. (2017) An Expanded Genome-Wide Association Study of Type 2 Diabetes in Europeans. *Diabetes* 66:2888–2902.

Protein website at Cardiovascular Epidemiology Unit (CEU)

Sun BB, et al. (2018) Genomic atlas of the human plasma proteome. *Nature* 558: 73-79.

[1] Robinson PN, Piro RM, Jager K (2018). Computational Exome and Genome Analysis, CRC.