유전적 위험도 측정 가이드라인(v1.0)

(Guideline for polygenic risk score)

유전체연구기술개발과

2021. 12. 16

목 차

1 유전적 위험도 개요

□ 유전적 위험도란?

○ 유전적 위험도(polygenic risk score, 이하 PRS)는 유전변이와 그의 유전적 효과를 이용해 계산된 개인의 질환에 대한 유전적 위험도임

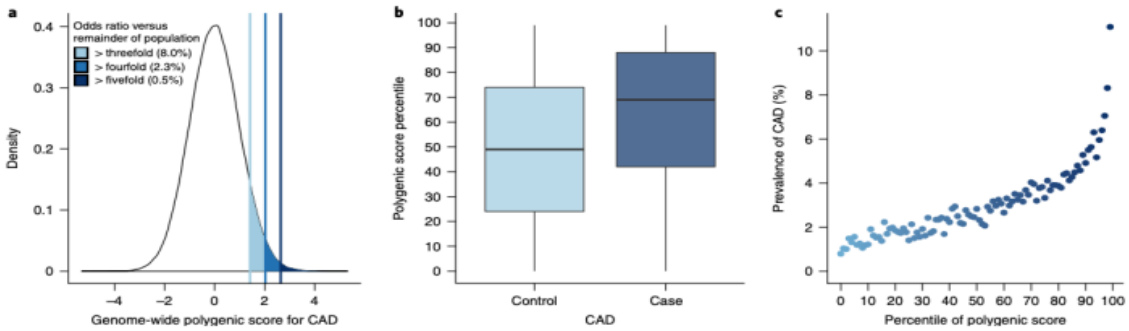
$$PRS_i = \sum_{j=1}^{m} \beta_j G_{ij} \quad (\beta : effect\ size,\ G: genotype,\ i: individual,\ j: trait\text{-}associated\ SNP)$$

- 기존 연구 결과에 따르면 상위 5%의 유전적 고위험군은 일반인 대비 3배 이상의 질환 발생 가능성이 높다는 것이 보고됨

Table 3 | Prevalence and clinical impact of a high GPS

High GPS definition	Reference group	Odds ratio	95% CI	*P* value
CAD				
Top 20% of distribution	Remaining 80%	2.55	2.43–2.67	$<1\times10^{-300}$
Top 10% of distribution	Remaining 90%	2.89	2.74–3.05	$<1\times10^{-300}$
Top 5% of distribution	Remaining 95%	3.34	3.12–3.58	6.5×10^{-264}
Top 1% of distribution	Remaining 99%	4.83	4.25–5.46	1.0×10^{-132}
Top 0.5% of distribution	Remaining 99.5%	5.17	4.34–6.12	7.9×10^{-78}
Atrial fibrillation				
Top 20% of distribution	Remaining 80%	2.43	2.29–2.59	2.1×10^{-177}
Top 10% of distribution	Remaining 90%	2.74	2.55–2.94	7.0×10^{-169}
Top 5% of distribution	Remaining 95%	3.22	2.95–3.51	1.1×10^{-152}
Top 1% of distribution	Remaining 99%	4.63	3.96–5.39	2.9×10^{-84}
Top 0.5% of distribution	Remaining 99.5%	5.23	4.24–6.39	3.5×10^{-56}

<그림 1. PRS를 분석 결과 (Khera AV et al. Nat Genet. 2018, Table 3 중 CAD, AF)>



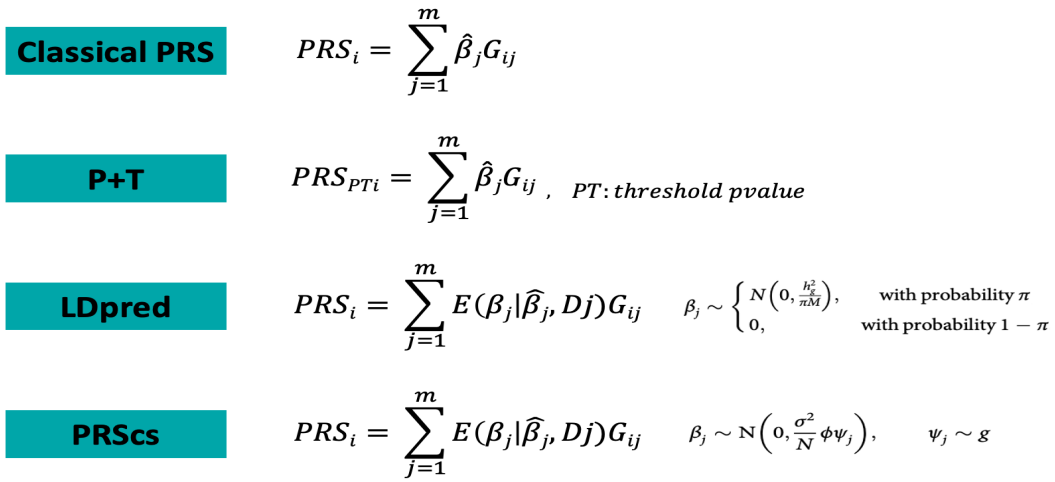
<그림 2. PRS 분포에 따른 심혈관질환 유병률 변화 (Khera AV et al. Nat Genet. 2018)>

□ PRS 계산 방법 개요

○ PRS는 구축 모형에 따라 다양한 소프트웨어를 활용하여 계산이 가능함

<표 1. Polygenic risk score 모형 종류>

No.	Method	# of marker	Software	Note
1	GRS	~ hundreds	plink	using validated marker
2	P+T (clumping + p-value)	~ millions	PRSice, Plink	using independent marker (no optimized)
3	LDpred	~ millions	LDpred, LDpred2	adjust LD
4	PRScs	~ millions	PRScs	adjust LD & Shrinkage factor z



<그림 3. PRS 기법에 따른 모형 (Ge T et al. Nat Commun. 2019)>

- 정교한 PRS 계산을 위해서 다음의 분석 과정을 따름

  1. 독립된 연구 집단에서 수행된 GWAS 결과 및 참조패널(reference panel)의 linkage disequilibrium(LD)를 활용한 모형 구축

     * GWAS: genome-wide association study

  2. 구축한 모형을 타겟(target) 유전체정보에 적용하여 연관성 분석, variance explained 분석 등 수행하여 최적화된 모형 선정 및 활용

- 독립된 GWAS 결과가 없거나 연구 집단의 특이성을 고려할 필요가 있는 경우 10-fold LOGO(leave on group out) 메타분석 활용함

2 분석을 위한 기본 정보 준비

□ PRS 분석을 위한 기본 정보 준비

○ 타겟 유전체정보: PRS를 계산하는 연구 대상 정보

- 유전체정보는 GWAS standard guideline에 따라 정도관리를 수행함
  * 한국인칩 홈페이지(https://koreanchip.org)에서 메뉴 항목에 '공지사항'->'학술관련'->'2019 통계유전학워크숍 발표자료 공유'의 자료 및 '홍보자료'->'백서' 참고
- 정제된 유전체정보를 imputation을 통해 imputed genotype 확보
  * 분석 방법은 상기 홈페이지에서 확인 가능하며, 분석 후 imputation quality score 0.8 이상의 고품질 유전체정보만 사용하는 것으로 추천함

○ PRS 분석을 위한 독립된 GWAS 분석 결과 준비

- 공개 웹사이트 등을 통해 summary statistics 자료를 확보하고 아래의 필수 정보 포함 여부를 확인함
  · chromosome, position, allele 정보 (effect, other allele), effect (beta), standard error, sample size, p-value
  · Position 정보 등은 PRS 소프트웨어를 활용시에 사용하는 reference panel과 타겟 유전체정보가 모두 동일한 genome build를 사용해야함 (예. hg19)
  · Strand에 따라 allele 정보가 바뀔 수 있으므로, 아래 웹사이트를 참고하여 Strand 정보를 확인하고 보정이 필요함
  * 참고: https://www.well.ox.ac.uk/~wrayner/strand/

<표 2. 주요 summary statistics 공개 웹사이트>

No.	Database	Website	Population	# of summary statistics
1	GWAS catalog	https://www.ebi.ac.uk/gwas/downloads/summary-statistics	Multi-ethnic	6,475
2	Global Biobank Engine	https://biobankengine.stanford.edu	Multi-ethnic	> 2,000
3	Biobank Japan	https://pheweb.jp/downloads	East Asian	224
4	T2D AMP portal	https://t2d.hugeamp.org	Multi-ethnic	292

3 GRS 분석 방법

□ GRS 분석이란?

○ GRS(Genetic Risk Score) 분석은 여러 연구를 통해 검증된 마커(validated marker)만을 사용하는 방법임

* 검증된 마커를 사용함으로써 안전성이 있으나 많은 연구가 수행되지 않은 표현형의 경우 적은 수의 마커로 인해 설명력이 낮은 단점이 있음

□ GRS 분석 방법

○ GRS 분석을 위한 준비

- plink 다운로드 및 설치 (https://www.cog-genomics.org/plink/)
- GRS 계산을 위한 검증된 마커 정보 준비

· 논문에서 보고 또는 연구자가 자체적으로 준비한 마커 정보

· 필수정보: Marker ID, effective allele(coded allele), effect (beta)

* Marker ID는 타겟 유전체정보와 동일 아이디가 필요. Allele 일치 여부 확인 필요

· 각 컬럼을 탭(tab)으로 구분하여 파일로 저장함

* 예) validated_marker_list.txt: Marker ID만 저장, validated_info.txt: 모든 정보

- GRS 계산을 위한 타겟 유전체정보 준비

· imputation된 데이터(plink binary format으로 변환 필요)에서 아래의 명령어를 참고하여 검증된 마커 정보만 추출함

* plink binary format 변환 및 옵션 등은 plink 홈페이지 참고

```
plink --bfile MYDATA --extract validated_marker_list.txt —make-bed --out MYDATA_PRS_target
```

○ Plink를 이용하여 GRS 계산

```
plink --bfile MYDATA_PRS_target --score validate_info.txt 1 2 3 header sum include-cnt double-dosage --out MYDATA_PRS_out
```

* --score 파일 <마커아이디 컬럼 번호> <effective allele 컬럼 번호> <effect 컬럼 번호> header (header가 있는 경우)

4 Clumping 분석 방법

□ Clumping 분석이란?

○ Summary statistics를 이용하여 p-value와 LD 정보를 이용하여 PRS 분석을 위한 마커를 선별하는 방법

□ Clumping 분석 방법

○ Clumping 분석을 위한 준비

- plink 다운로드 및 설치

· 참고: https://www.cog-genomics.org/plink/

- reference panel 준비

· 아래 웹사이트에서 1,000 genomes phase 3 데이터를 다운로드 받고, plink binary 형식으로 저장 및 인종에 맞추어 데이터 사용

* 데이터 및 인종 정보 참고: https://www.cog-genomics.org/plink/2.0/resources#1kg_phase3

- PRS 계산을 위한 타겟 유전체정보 준비

· imputation된 데이터를 plink binary format으로 변환 필요

* plink binary format 변환 및 옵션 등은 plink 홈페이지 참고

- Summary statistics 준비

· Summary statistics 정보를 다운로드 받고 필수 정보를 추출함

· 필수 정보: SNP (마커 아이디), Effective allele(coded allele), effect (beta), P (P-value), SNP과 P의 컬럼 header 명칭은 SNP, P로 설정

* SNP 아이디는 reference panel과 동일한 아이디 필요

○ Clumping 분석

- Plink를 이용하여 clumping 분석

· 아래의 명령어를 참고하여 다양한 threshold를 적용하여 clumping 분석

plink --bfile 1KGP3_EAS --clump sum_stat.txt --clump-p1 0.0001 --clump-p2 0.01 --clump-r2 0.1 --clump-kb 500 --out CLUMP_OUT

* --clump <summary stat 파일

  --clump-p1 <index SNP의 p-value threshold>

  --clump-2 <secondary SNP의 p-value threshold>

  --clump-r2 <LD threshold>

  --clump-kb <clumping하는 구간, index SNP에서부터 +-로 kb 단위 거리>

· clumping으로 선별된 SNP(*.clumped의 SNP 컬럼 확인)에 해당하는 정보만 summary stat과 plink format으로 준비 (--extract 옵션 사용)

- Plink를 이용하여 PRS 계산

plink --bfile MYDATA_clump --score sum_stat_clump.txt 1 2 3 header sum include-cnt double-dosage --out MYDATA_clump_PRS

* MYDATA_clump: clumping으로 선별된 SNP만 포함한 plink binary format 파일
  sum_stat_clump.txt: clumping으로 선별된 SNP만 포함된 summary statistics

  --score 파일 <마커아이디 컬럼 번호> <effective allele 컬럼 번호> <effect 컬럼 번호> header (header가 있는 경우)

5 LDpred (v1) 분석 방법

□ LDpred v1 분석이란?

○ LD 값을 보정하여 effect size를 보정하는 방법으로, summary statistics의 모든 유전변이 정보의 effect를 조정함

- 참고 논문: Vilhjalmsson et al. Modeling linkage disequilibrium increases accuracy of polygenic risk scores, American Journal of Human Genetics 2015

□ LDpred 분석 방법

○ LDpred 분석을 위한 준비

- LDpred 설치방법: 참고 https://github.com/bvilhjal/ldpred

· pip install --user ldpred

· dependency가 있는 package는 h5py, scipy, libplinkio (package 이름은 plinkio) pip로 설치

· pip 설치는 인터넷 연결이 필요함

- reference panel 준비

· 아래 웹사이트에서 1,000 genomes phase 3 데이터를 다운로드 받고, plink binary 형식으로 저장 및 인종에 맞추어 데이터 사용

* 데이터 및 인종 정보 참고: https://www.cog-genomics.org/plink/2.0/resources#1kg_phase3

· 준비한 데이터는 chromosome 별로 분류하여 저장

- Summary statistics 준비

· 아래의 형식에 맞춰서 데이터 준비

chr	pos	ref	alt	reffrq	info	rs	pval	effalt
chr1	4600233	A	C	0.3494	1	1:4600233_A/C	0.5965	0.0017
chr1	183728249	T	C	0.0251	1	1:183728249_T/C	0.9188	-0.0011

* rs 컬럼의 아이디는 reference panel, summary statistics 및 타겟 유전체정보 모두 일치해야함

· alt (alternative allele)가 effective allele이며, reffrq의 경우 reference panel의 plink binary format 자료를 아래의 명령어를 참고하여 frequency 정보를 추출하고 summary statistics에 포함함

```
plink --bfile 1KGP3_EAS –freq --out 1KGP3_EAS_FRQ
```

· 준비한 데이터는 chromosome 별로 분류하여 저장

- PRS 계산을 위한 타겟 유전체정보 준비

· imputation된 데이터를 plink binary format으로 변환 필요

* plink binary format 변환 및 옵션 등은 plink 홈페이지 참고

○ LDpred 분석

- Step 1: Coordinate data: reference 데이터와 summary statistics 정보 일치 및 포맷 변환

```
ldpred coord \
 --rs rs \
 --A1 alt \
 --A2 ref \
 --pos pos \
 --chr chr \
 --pval pval \
 --eff effalt \
 --ssf-format CUSTOM \
 --N 100000 \
 --ssf sumstat.chr1 \
 --out=MYDATA_COORD_CHR1 \
 --eff_type LINREG \
 --gf 1KGP3_EAS_CHR1 \
 > MYDATA_COORD_CHR1.log
```

* 각 분석은 chromosome 단위로 분석 필요

* N: 샘플 수, eff_type: LINREG, OR, LOGOR, BLUP 중 선택, gf: reference panel의 plink binary format file의 이름

- Step 2: LDpred SNP weight 생성

```
ldpred fast \
 --cf MYDATA_COORD_CHR1 \
 --ldr 270 \
 --ldf MYDATA_CHR1.ld \
 --out MYDATA_CHR1.weight \
 --N 100000 \
 > MYDATA_CHR1_WEIGHT.log
```

* ldpred fast(BLUP방법)/gibbs (gibbs sampling 방법) 중에서 선택
* 각 분석은 chromosome 단위로 분석 필요
* 분석 방법에 따라 weight가 다양하게 생성됨. Step 3을 통해 최적값 확인
* ldr 값은 ladius로 chromosome의 크기와 마커수를 고려하여 (보통 나눠서) 설정

- Step 3: 개인별 PRS 계산

```
ldpred score \
 --gf MYDATA_CHR1 \
 --rf MYDATA_CHR1.weight \
 --out MYDATA_PRS_CHR1.score \
 --pf MYPHENO.txt \
 --pf-format STANDARD \
 > /ADATA/JHM/PRS/LDpred/OUT/FPG_PRS_CHR22_score_log.log
```

* --pf Phenotype파일: phenotype 파일은 standard 포맷일 경우 ID, phenotype값
* phenotype 파일을 입력하면 최적의 weight를 찾아주며, log 파일 내에서 optimal polygenic score로 확인이 가능함

6 LDpred (v2) 분석 방법

□ LDpred v2 분석이란?

○ LDpred v1을 개량하여 더 정확도가 높고 genome-wide로 분석이 가능하도록 개발된 분석법

- 참고 논문: Prive et al. LDpred2: better, faster, stronger, Bioinformatics 2020

□ LDpred v2 분석 방법

○ LDpred v2 분석을 위한 준비

- LDpred v2 설치방법: 참고 https://choishingwan.github.io/PRS-Tutorial/ldpred/

  * 참고: https://privefl.github.io/bigsnpr/articles/LDpred2.html

- reference panel 준비

· 아래 웹사이트에서 1,000 genomes phase 3 데이터를 다운로드 받고, plink binary 형식으로 저장 및 인종에 맞추어 데이터 사용

  * 데이터 및 인종 정보 참고: https://www.cog-genomics.org/plink/2.0/resources#1kg_phase3

· LDpred v2의 accessory file 중 HapMap phase 3 정보를 확인하여 reference panel에서 HapMap phase 3에 해당되는 정보만 추출

  * plink --bfile 1KGP3_EAS --extract hapmap3_list.txt --make-bed --out 1KGP3_EAS_HM3

· 준비한 데이터는 LDpred v2 설치를 통해 부가적으로 설치된 bigsnpr 패키지를 이용하여 plink 파일을 rds 파일로 변환

```
R ## R statistics software에서 분석 필요
library(bigsnpr)
snp_readBed("1KGP3_EAS_HM3.bed")
q()
```

  * 1KGP3_EAS_HM3.rds 생성 여부 확인 필요

- Summary statistics 준비

· 아래의 형식에 맞춰서 데이터 준비 (HapMap phase3의 SNP만 포함)

chr rsid pos a0 a1 beta beta_se p n_eff

* rsid는 reference panel과 일치해야함, a1이 effective allele임
* beta = effect (beta), beta_se = standard error
* n_eff는 quantitative trait의 경우는 sample size이며, case-control의 경우는 4 / (1/case수 + 1/control수)

- PRS 계산을 위한 타겟 유전체정보 준비 (HapMap phase3의 SNP만 포함)

· imputation된 데이터를 plink binary format으로 변환 필요

* plink binary format 변환 및 옵션 등은 plink 홈페이지 참고

· 준비한 데이터는 LDpred v2 설치를 통해 부가적으로 설치된 bigsnpr 패키지를 이용하여 plink 파일을 rds 파일로 변환

```
R ## R statistics software에서 분석 필요
library(bigsnpr)
snp_readBed("MYDATA_HM3.bed")
q()
```

* MYDATA_HM3.rds 생성 여부 확인 필요

○ LDpred v2 분석

- 모든 분석은 R에서 bigsnpr library를 통해 분석을 수행함

- Step 1: Summary statistics, reference panel 로딩 및 LD 계산

```
## Summary stat file name
ss.name <- "MY.SUM.STAT.txt"
NCORES <- 60 ## parallel computing에 사용할 core 수

## library loading
library(bigsnpr)
library(data.table)
library(magrittr)
options(bigstatsr.check.parallel.blas = FALSE)
options(default.nproc.blas = NULL)
```

```
## Reference panel 불러오기
obj.bigSNP <- snp_attach("1KGP3_EAS_HM3..rds")
tmp <- tempfile(tmpdir = "./tmp-data")
on.exit(file.remove(paste0(tmp, ".sbk")), add = TRUE)
corr <- NULL
ld <- NULL

## Summary stat 불러오기
sumstats <- bigreadr::fread2(ss.name)
names(sumstats) <- c("chr", "rsid", "pos", "a0", "a1", "beta", "beta_se", "p", "n_eff")
## Summary stat과 reference panel의 SNP 매칭
fam.order <- NULL
map <- obj.bigSNP$map[-3]
names(map) <- c("chr", "rsid", "pos", "a1", "a0")
info_snp <- snp_match(sumstats, map)

#Reference panel을 이용하여 LD 계산
genotype <- obj.bigSNP$genotypes
CHR <- map$chr
POS <- map$pos
POS2 <- snp_asGeneticPos(CHR, POS, dir = "LDpredV2/ACC", ncores = NCORES)
##LDpred V2의 accessory file들의 위치

# calculate LD 부분
for (chr in 1:22) {
  # Extract SNPs that are included in the chromosome
ind.chr <- which(info_snp$chr == chr)
ind.chr2 <- info_snp$`_NUM_ID_`[ind.chr]
# Calculate the LD
corr0 <- snp_cor(
```

```
        genotype,
        ind.col = ind.chr2,
        ncores = NCORES,
        infos.pos = POS2[ind.chr2],
        size = 3 / 1000
)
    if (chr == 1) {
        ld <- Matrix::colSums(corr0^2)
        corr <- as_SFBM(corr0, tmp)
    } else {
        ld <- c(ld, Matrix::colSums(corr0^2))
        corr$add_columns(corr0, nrow(corr))
    }
}

## LD score regression
df_beta <- info_snp[,c("beta", "beta_se", "n_eff", "_NUM_ID_")]
ldsc <- snp_ldsc(    ld,
                length(ld),
                chi2 = (df_beta$beta / df_beta$beta_se)^2,
                sample_size = df_beta$n_eff,
                blocks = NULL)
h2_est <- ldsc[["h2"]]    ## LDSC 기반 estimated genetic heritability
```

- Step 2: (option #1) Infinitesimal model

```
genofile <- "MYDATA_HM3.rds"
phenofile <- “MYDATA_pheno.txt”

#inf model
beta_inf <- snp_ldpred2_inf(corr, df_beta, h2 = h2_est)
```

```
## Loading MYDATA genotype
obj.bigSNP <- snp_attach(genofile)
genotype <- obj.bigSNP$genotypes
fam <- obj.bigSNP$fam
ind.test <- 1:nrow(genotype)

##### PRS 계산
pred_inf <- big_prodVec(     genotype,
                             beta_inf,
                             ind.row = ind.test,
                             ind.col = info_snp$`_NUM_ID_`)
#### pred_inf에 계산된 PRS 값이 있으며, MYDATA의 plink format fam 파일 순서로 정리됨
```

- Step 2: (option #2) Grid model

```
genofile <- "MYDATA_HM3.rds"
phenofile <- “MYDATA_pheno.txt”

#grid model
p_seq <- signif(seq_log(1e-4, 1, length.out = 17), 2)
h2_seq <- round(h2_est * c(0.7, 1, 1.4), 4)
grid.param <-
      expand.grid(p = p_seq,
      h2 = h2_seq,
      sparse = c(FALSE, TRUE))

# Get adjusted beta from grid model
beta_grid <- snp_ldpred2_grid(corr, df_beta, grid.param, ncores = NCORES)

## Loading MYDATA genotype
obj.bigSNP <- snp_attach(genofile)
genotype <- obj.bigSNP$genotypes
fam <- obj.bigSNP$fam
```

```
ind.test <- 1:nrow(genotype)

# Grid model PRS 계산
pred_grid <- big_prodMat(    genotype,
                beta_grid,
                ind.col = info_snp$`_NUM_ID_`)
#### pred_grid에 계산된 PRS 값이 있으며, MYDATA의 plink format fam 파일 순서로 정리됨
```

- Step 2: (option #3) Auto model

```
genofile <- "MYDATA_HM3.rds"
phenofile <- “MYDATA_pheno.txt”

# Get adjusted beta from the auto model
multi_auto <- snp_ldpred2_auto(
            corr,
            df_beta,
            h2_init = h2_est,
            vec_p_init = seq_log(1e-4, 0.9, length.out = NCORES),
            ncores = NCORES
)
beta_auto <- sapply(multi_auto, function(auto) auto$beta_est)

## Loading MYDATA genotype
obj.bigSNP <- snp_attach(genofile)
genotype <- obj.bigSNP$genotypes
fam <- obj.bigSNP$fam
ind.test <- 1:nrow(genotype)

# Auto weighted PRS 계산
pred_auto <- big_prodMat(genotype,
            beta_auto,
            ind.row = ind.test,
            ind.col = info_snp$`_NUM_ID_`)
```

```
# scale the PRS generated from AUTO
pred_scaled <- apply(pred_auto, 2, sd)
final_beta_auto <-
          rowMeans(beta_auto[,
          abs(pred_scaled -median(pred_scaled)) <
          3 * mad(pred_scaled)])
pred_auto <-
          big_prodVec(genotype,
          final_beta_auto,
          ind.row = ind.test,
          ind.col = info_snp$`_NUM_ID_`)
#### pred_auto에 계산된 PRS 값이 있으며, MYDATA의 plink format fam 파일 순서로 정리됨
```

7 PRScs 분석 방법

□ PRScs 분석이란?

○ LD 값 및 shrinkage factor를 이용하여 effect size를 보정하는 방법으로, summary statistics의 모든 유전변이 정보의 effect를 조정함

- 참고 논문: Ge et al. Polygenic prediction via Bayesian regression and continuous shrinkage priors, Nature Communications 2019

□ PRScs 분석 방법

○ PRScs 분석을 위한 준비

- PRScs 설치방법: 참고 https://github.com/getian107/PRScs
- reference panel 준비

· 아래 웹사이트에서 분석하고자 하는 인종의 LD reference panel을 다운로드 받음 (1,000 Genomes project phase 3 데이터로 HapMap phase3의 SNP만 사용함)

* https://github.com/getian107/PRScs

- Summary statistics 준비

· 아래의 형식에 맞춰서 데이터 준비 (HapMap phase3의 SNP만 포함)

SNP A1 A2 BETA P

* SNP의 ID는 rs id이며 reference panel의 아이디와 일치해야함
* BETA = effect (beta), P = P-value

- PRS 계산을 위한 타겟 유전체정보 준비 (HapMap phase3의 SNP만 포함)

· imputation된 데이터를 plink binary format으로 변환 필요

* plink binary format 변환 및 옵션 등은 plink 홈페이지 참고

○ PRScs 분석

- Step 1: PRScs AUTO 모델 분석

export MKL_NUM_THREADS=5 && \

export NUMEXPR_NUM_THREADS=5 && \

export OMP_NUM_THREADS=5 && \

PRScs.py \

--ref_dir=reference 위치 \

--bim_prefix=타겟유전체정보 파일 이름 (확장자 제외) \

--sst_file= summary statistics 파일 \

--n_gwas=샘플사이즈 \

--out_dir=out파일이름 > out파일 로그파일

* export MKL_NUM_THREADS 등: core 사용 갯수를 제한함. 여기서는 5개

* ref_dir: reference panel 정보 위치 (1KG EAS 정보임)

* bim_prefix: target이 되는 파일의 마커 정보 추출 (여기서는 24K로 고정)

* sst_file: summary statistics 파일

* n_gwas: 샘플 사이즈

* PRScs의 다른 옵션을 주지 않으면 PRScs는 auto 모드로 분석됨

- Step 2: PRScs AUTO 분석 결과를 이용하여 PRS 계산

plink –bfile MYDATA_HM3 --score PRScs결과값 2 4 6 header sum include-cnt double-dosage --out MYDATA_PRScs_PRS

8 LOGO 분석 방법

□ 10-fold LOGO(leave on group out)

○ 독립된 연구 집단에서 수행한 GWAS 결과가 없거나, 연구 집단의 특이성 등을 고려하여 한국인 인구집단에 적용이 제한되는 경우 10-fold LOGO(leave on group out) 메타분석을 수행함

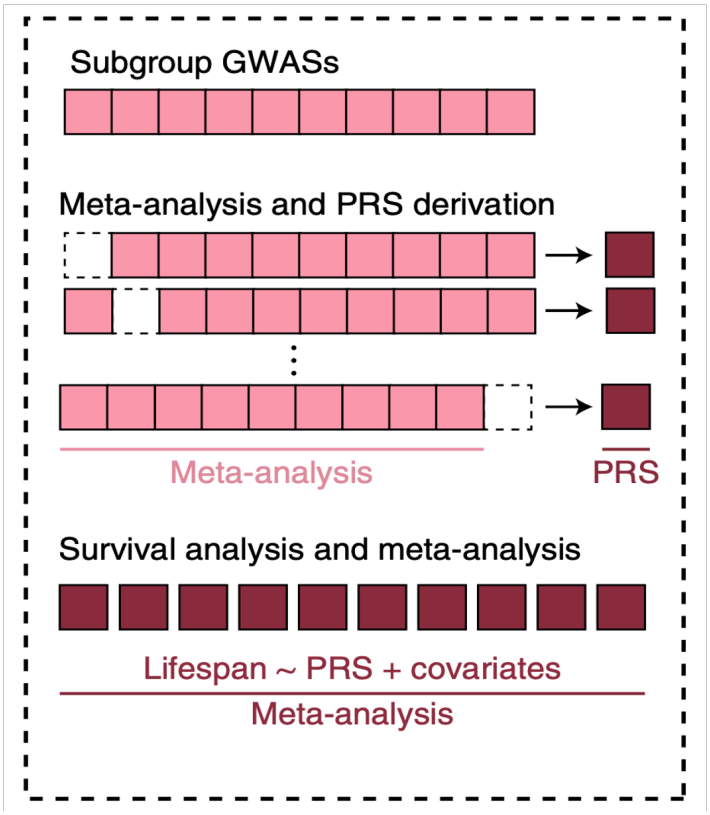
- 여러 집단으로 나눠서 분석함으로써 가상의 독립된 연구집단 GWAS 결과를 활용하는 효과를 적용할 수 있음

① PRS 계산할 전체 집단을 10개 집단으로 분류

② 1~9번 집단에서 각각 연관성 분석 및 통합 메타분석 수행

③ 메타분석 결과 summary statistics를 이용해 PRS 모형 구축 및 구축된 모형을 이용해 10번 집단의 PRS 계산

④ 위와 같이 독립된 집단의 연관성 분석 결과를 이용하여 10개 집단 모두 PRS 계산



<그림 4. LOGO 메타분석을 이용한 PRS 계산 예 (Sakaue et al. Nat Med. 2020)>

□ (참고) 메타 분석 방법

○ Metal software 설치

- 아래의 링크를 참고하여 설치함

  * https://en.wikipedia.org/wiki/Metal

  * https://genome.sph.umich.edu/wiki/METAL_Quick_Start

○ Input data 준비

- 아래의 정보를 포함하여 input data 작성

· Marker ID : variant의 ID로 메타 분석에 사용되는 모든 데이터가 동일한 ID를 가지고 있어야함

· Effective Allele : effect (beta)를 추정할 때 사용되는 coded allele 정보

· Other Allele : 그 외 allele 정보

· Effect : 연관성 분석 결과의 effect (beta) 정보

· Stderr : effect의 standard error 값

· P-value : 연관성 분석 결과의 p-value

· N : 연관성 분석에 사용된 샘플 사이즈

○ Metal script 예시

- Metal을 실행하고 아래의 스크립트를 참고하여 메타분석 수행

```
SCHEME STDERR     ## weight 옵션: STDERR 또는 SAMPLESIZE
MARKER SNP    ## MarkerID 컬럼 이름
ALLELE ALT REF  ## Effective allele, Other allele 컬럼 이름
EFFECT BETA    ## Effect 컬럼 이름
STDERR SE      ## Stderr 컬럼 이름
PVALUE PVALUE  ## P-value 컬럼 이름
PROCESS DATA1.input  ## Input file 1번 이름
PROCESS DATA2.input  ## Input file 2번 이름
```

OUTFILE meta_out .txt ## output 이름. .txt(확장자) 앞에 공백 필수

ANALYZE HETEROGENEITY ## Meta 분석 (heterogeneity 분석 옵션)

QUIT ## 종료

- input 마다 컬럼 명칭이 다른 경우는 아래와 같이 process 이전에 각 파일의 컬럼 명칭을 지정하고 process 수행

MARKER SNP ## 1번 파일의 MarkerID 컬럼

ALLELE ALT REF ## 1번 파일의 Effective, Other allele 컬럼

EFFECT BETA ## 1번 파일의 Effect 컬럼

STDERR SE ## 1번 파일의 Stderr 컬럼

PVALUE PVALUE ## 1번 파일의 P-value 컬럼

PROCESS DATA1.input ## Input file 1번 이름

MARKER SNPID ## 1번 파일의 MarkerID 컬럼

ALLELE EA OA ## 1번 파일의 Effective, Other allele 컬럼

EFFECT Effect ## 1번 파일의 Effect 컬럼

STDERR SE ## 1번 파일의 Stderr 컬럼

PVALUE Pvalue ## 1번 파일의 P-value 컬럼

PROCESS DATA2.input ## Input file 2번 이름