건강한 국민, 안전한 사회

한국인칩 유전체정보 분석 실습

미래의료연구부 유전체연구기술개발과





국립보건연구원

실습목차

- Genotype calling & QC
- Phasing & Imputation
- **Analysis**
- Visualization & Annotation

목차

- Genotype calling & QC
- Phasing & Imputation
- **Analysis**
  - **Data**
  - **Single variant association test**
  - **Burden test (Gene-based test, regional test)**
  - **PRS analysis**
- Visualization & Annotation

Data

- **Genotype data**: Imputed genotype data (KBAv2.0A, B)

```
hl.import_vcf("VCF.vcf.bgz").write("VCF.mt", overwrite=True)
mt = hl.read_matrix_table("VCF.mt")
```

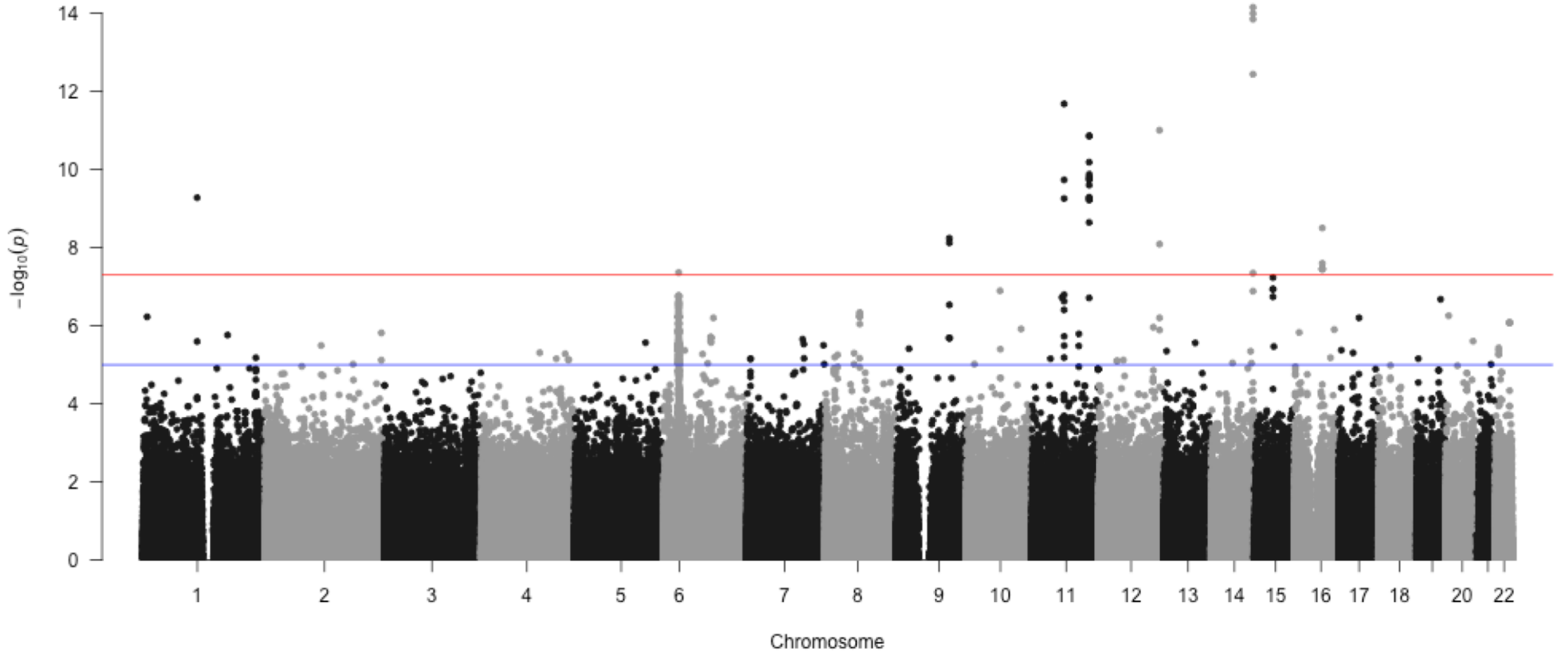
- **Phenotype data:** High density lipoprotein cholesterol (HDL)

Sample	Phenotype (for linear)	Phenotype (for logistic)	Age
sample1	1.0581	1	52
sample2	-0.6560	0	40

Single variant association test

Single variant test

- Single variant association test: phenotype과 연관이 있는 SNP 찾는 분석



Single variant test - Linear regression

- Linear regression analysis: quantitative trait (QT)

```
phenotype = (hl.import_table("phenotype.txt", types={"Sample":hl.tstr,
"HDL":hl.tfloat32, "Age":hl.tint32).key_by("Sample"))
mt_pheno = mt.annotate_cols(pheno = phenotype[mt.s])

# genotype data
linear = hl.linear_regression_rows(x=mt_pheno.GT.n_alt_alleles(), y=mt_pheno.pheno.HDL,
covariates=[1.0, mt_pheno.pheno.Age])   # covar이 없을 경우 covariates=[1.0] 로 넣으면 됨
# imputated data
linear = hl.linear_regression_rows(x=mt_pheno.DS, y=mt_pheno.pheno.HDL,
covariates=[1.0, mt_pheno.pheno.Age])   # covar이 없을 경우 covariates=[1.0] 로 넣으면 됨
```

Single variant test - Linear regression output

National Institute of Health
Republic of Korea

locus	alleles	n	sum_x	y_transpose_x	beta	standard_error	t_stat	p_value
11:76977	["G","A"]	504	1.49E+00	-5.66E-01	-2.40E+00	2.12E+00	-1.13E+00	2.58E-01
11:113611	["T","C"]	504	1.45E+00	-1.92E-01	-4.40E+00	4.97E+00	-8.86E-01	3.76E-01

- n: the number of samples
- sum_x: sum of input values x
- y_transpose_x: dot product of response vector y with the input vector x
- beta: fit effect coefficient of x
- standard_error: estimated standard error
- t_stat: t-statistic
- p_value: p-value

Single variant test - Logistic regression

- Logistic regression analysis: case-control

```
phenotype = (hl.import_table("phenotype.txt",types={"Sample":hl.tstr,
"HDL":hl.tint32, "Age":hl.tint32).key_by("Sample"))
mt_pheno = mt.annotate_cols(pheno = phenotype[mt.s])

#genotype data
logistic = hl.logistic_regression_rows(x=mt_pheno.GT.n_alt_alleles(),
y=mt_pheno.pheno.CASE, covariates=[1.0, mt_pheno.pheno.Age], test='wald')
# imputated data    # test = Wald test ('wald'), likelihood ratio test ('lrt'), Rao score test ('score'), Firth test ('firth')
logistic = hl.logistic_regression_rows(x=mt_pheno.DS, y=mt_pheno.pheno.Case,
covariates=[1.0, mt_pheno.pheno.Age], test='wald')
```

Single variant test - Logistic regression output

National Institute of Health
Republic of Korea

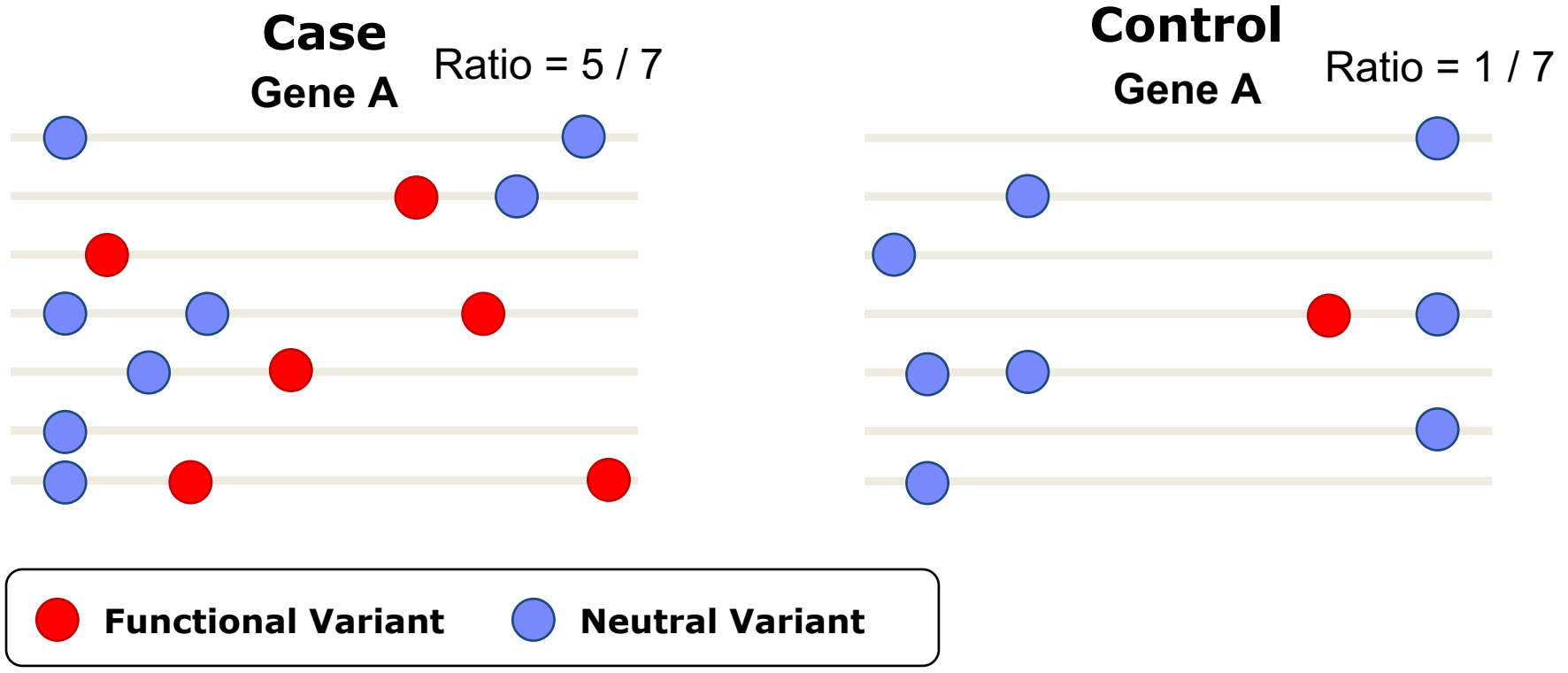
locus	alleles	beta	standard_error	z_stat	p_value	fit
11:76977	["G","A"]	-8.19E+01	6.39E+01	-1.28E+00	2.00E-01	{"n_iterations":9,"converged":true,"exploded":false}
11:113611	["T","C"]	2.40E+00	1.27E+01	1.88E-01	8.51E-01	{"n_iterations":4,"converged":true,"exploded":false}

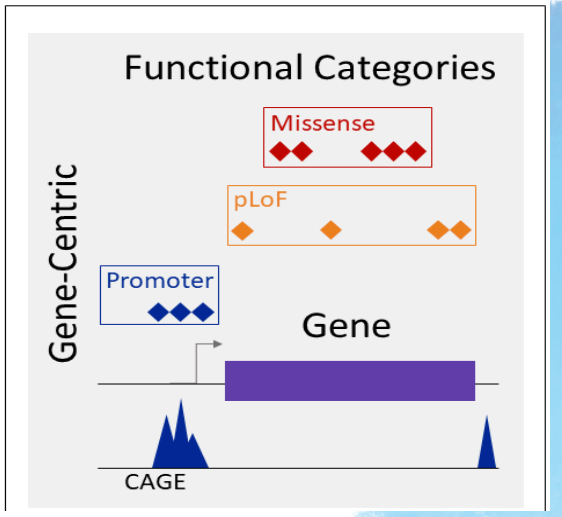
- beta: fit effect coefficient
- standard_error: estimated standard error
- z_stat: wald z-statistic
- p-value: wald p-value
- fit
  - n_iterations: number of iterations until convergence, explosion, or reaching the max
  - converged: if iteration converged
  - exploded: if iteration exploded

Burden test - Gene-based association test

Gene-based test

- Gene-based test: 희귀변이에 대한 효율적인 연관성분석 (variant sets defined by **gene**)
- Testing the ratio of individuals with functional variants between cases and controls





Gene-based test - Linear regression

- Linear regression analysis: quantitative trait (QT)

```
phenotype = (hl.import_table("phenotype.txt",types={"Sample":hl.tstr,
"HDL":hl.tfloat32, "Age":hl.tint32).key_by("Sample"))
mt_pheno = mt.annotate_cols(pheno = phenotype[mt.s])

mt_vep = hl.vep(mt_pheno, "vep.json")   # VEP annotation 다음강의에서 설명

linear = hl.skat(key_expr=mt_vep.vep.transcript_consequences.gene_symbol, weight_expr
= 1.0, x=mt_vep.GT.n_alt_alleles(), y=mt_vep.pheno.HDL, covariates=[1.0,
mt_vep.pheno.Age], logistic = False)
```

Gene-based test - Linear regression output

id	size	q_stat	p_value	fault
[gene1,gene2]	10513	6.14E+04	1.86E-01	0
[gene1,gene3]	7560	4.41E+04	1.08E-01	0

- id : the group parameter
- size: the number of variants in this group
- q_stat: the Q statistic, see Notes for why this differs from the paper
- p_value: the test p-value for the null hypothesis that the genotypes have no linear influence on the phenotypes
- fault: 0-If converged is true, 1 or 2-If converged is false

Gene-based test - Logistic regression

- Logistic regression analysis: case-control

```
phenotype = (hl.import_table(”phenotype.txt",types={"Sample":hl.tstr,
"HDL":hl.tint32, ”Age":hl.tint32).key_by("Sample"))
mt_pheno = mt.annotate_cols(pheno = phenotype[mt.s])

mt_vep = hl.vep(mt_pheno, "vep.json")

logistic = hl.skat(key_expr=mt_vep.vep.transcript_consequences.gene_symbol,
weight_expr = 1.0, x=mt_vep.GT.n_alt_alleles(), y=mt_vep.pheno.CASE, covariates=[1.0,
mt_vep.pheno.AGE], logistic = True)
```

Gene-based test - Logistic regression_output

id	size	q_stat	p_value	fault
[gene1, gene2]	2	2.36E+02	1.38E-01	0
[gene1, gene3]	3	1.90E+01	2.00E+00	1

- id : the group parameter
- size: the number of variants in this group
- q_stat: the Q statistic, see Notes for why this differs from the paper
- p_value: the test p-value for the null hypothesis that the genotypes have no linear influence on the phenotypes
- fault: 0-If converged is true, 1 or 2-If converged is false

Significant SNP

- Significant P-value threshold
  - Genome-wide threshold: $5x10^{-8}$
  - Suggestive threshold: $1x10^{-5}$

```
significant = gwas.filter(gwas.p_value<=5e-8)
```

Top signal

- Clumping

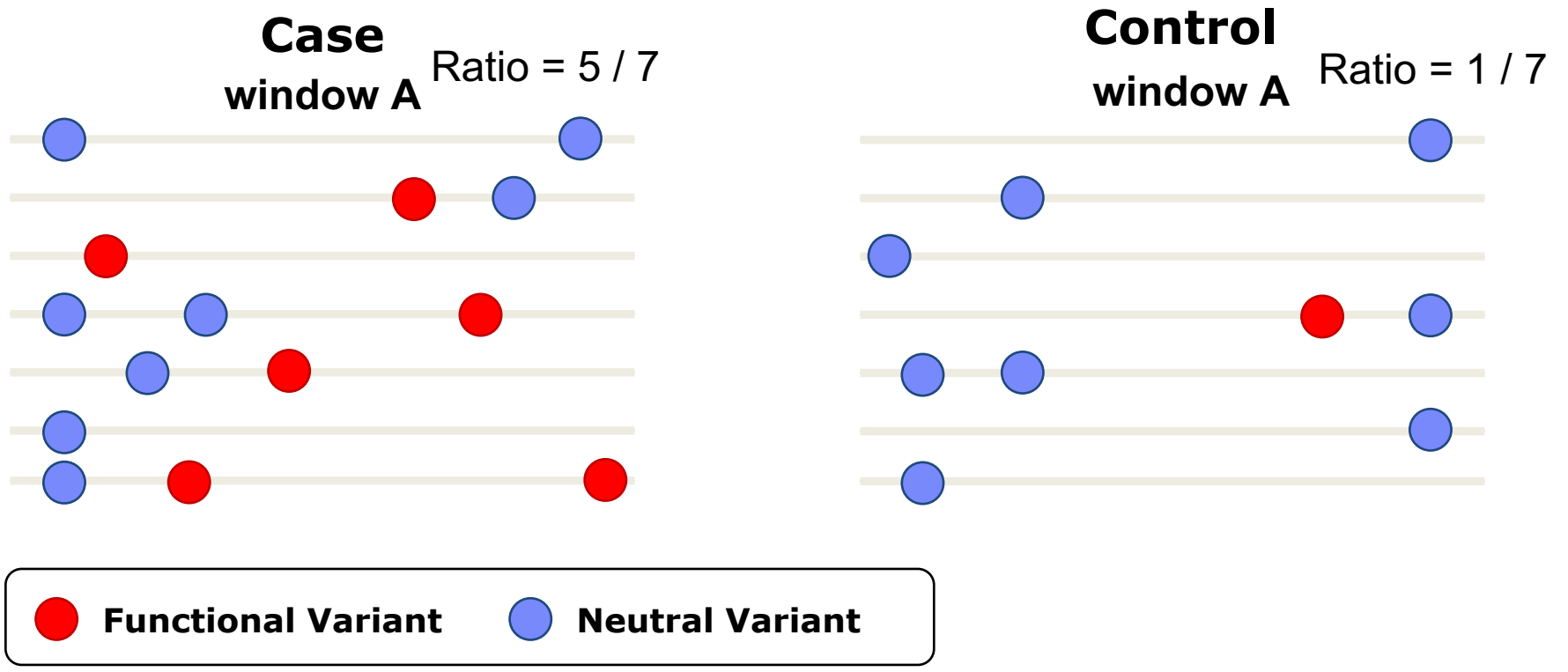
```
gwas = linear.select(SNP = hl.variant_str(linear.locus, linear.alleles), P =
linear.p_value)
gwas = gwas.key_by(gwas.SNP)
gwas = gwas.select(gwas.P)
gwas.export('linear_sumstat.tsv', header=True)
hl.export_plink(mt, 'mt', fam_id=mt.s, ind_id=mt.s)

$ plink --bfile mt --clump linear_sumstat.tsv --clump-best --clump-p1 5e-8 --clump-p2
5e-8 --clump-r2 0.5 --clump-kb 500 --out mt
```

Burden test - Regional association test

Regional association test

- regional test: 희귀변이에 대한 효율적인 연관성분석 (variant sets defined by **window**)
- Testing the ratio of individuals with functional variants between cases and controls



PRS analysis

PRS analysis

- PRS (Polygenic Risk Score) analysis: 개인의 질병 등에 대한 유전적 위험도를 평가하는 점수

  $PRS_j = \sum_{i=1}^{n} \beta_i G_{ij}$ (β: effect size , i: SNPs 1, ⋯, n, j: individual)

Method	# of markers	Software	Limitation
PRS using validated markers	~ hundreds	-	Limited # of markers
Unadjusted PRS	~ millions	-	No LD info.
Clumping(or Pruning) + P-value thresholding	~ millions	PRSice	Not optimized
Bayes based analysis (Individual data)	~ millions	BayesR	Accurate but slow
local LD info (Summary stat.)	~ millions	LDpred	Less accurate (Fast and efficient)
local LD + Shrinkage factor (Summary stat.)	**~ millions**	**PRS-CS**	**Less accurate (Fast and efficient)**

Data

- **Genotype data**: Imputed genotype data (KBAv2.0A, B)
- **GWAS summary statistic**: BBJ HDL-C GWAS summary statistic
  - BBJ PheWeb https://pheweb.jp/downloads
- **LD reference panel**: 1000 Genome Project phase3 East Asian
  - https://github.com/getian107/PRScs - Download the LD reference panels and extract files

Data - Genotype data

- Genotype data: PLINK (.bim, .fam, .bed)

```
hl.import_vcf('.vcf.bgz').write(.mt')
mt = hl.read_matrix_table('.mt')
hl.export_plink(mt, 'mt', fam_id=mt.s, ind_id=mt.s)
```

Data - Genotype data

National Institute of Health
Republic of Korea

- Genotype data matching **rsID**

```
mt_vep = hl.vep(mt, 'vep.json')
mt_vep_rs =
mt_vep.annotate_rows(info=mt_vep.info.annotate(rsID=mt_vep.vep.colocated_variants.id[0]))
hl.export_vcf(mt_vep_rs, 'mt_vep.vcf.bgz')

$ bcftools query -f '%ID\t%rsID\n' mt_vep.vcf.bgz | grep 'rs' > rsID.txt
$ plink --bfile mt --update-name rsID.txt --make-bed --out mt_rsID
```

Data - GWAS summary statistic

- GWAS summary statistic formatting

SNP	A1	A2	BETA	P
rs4970383	C	A	-0.0064	4.7780e-01

SNP	A1	A2	OR	P
rs4970383	C	A	0.9825	0.5737

SNP: rsID, A1: alternative(effect) allele, A2: reference allele

```
# BBJ GWAS summary statistic
$ awk '{print $2"\t"$6"\t"$7"\t"$12"\t"$11}' BBJ_GWASsumstats.txt  | \
sed 's/ALLELE1/A1/' | sed 's/ALLELE0/A2/' | sed 's/P_LINREG/P/' > GWASsumstats.txt
```

PRS-CS

- Adjusted effect size calculation using PRS-CS

```
$ python3 PRScs.py --bim_prefix=mt_rsID \  # required, genotype bim prefix matching rsID
 --ref_dir= ldblk_1kg_eas \                # required, LD reference panel path
 --sst_file=GWASsumstats.auto.txt \        # required, formatting GWAS summary statistics
 --n_gwas=74970 \   # required, GWAS summary statistics sample size
 --out_dir=PRS \    # required
 --chrom=CHR
```

CHROM	rsID	POS	Alternative (Effect) allele	Reference allele	Adjusted BETA (Adjusted effect size)
11	rs3741411	199256	G	A	-2.076827e-04
11	rs11245997	204680	A	G	1.428821e-05

- PRS calculation

```
$ plink --bfile mt_rsID \
 --out mt_rsID_PRS \
 --score PRS_pst_eff_a1_b0.5_phiauto.txt \   # PRS-CS output
 2 4 6 header sum    # 사용할 컬럼, 합계 표기
```

FID	IID	PHENO	CNT	CNT2	SCORESUM PRS value
sample1	sample1	-9	20590	4808	0.133214
sample2	sample2	-9	20590	4658	-0.0150038

건강한 국민, 안전한 사회

국립보건연구원