PLEIOVAR – Development

Osorio Meirelles

January 2023

PLEIOVAR Scoring

- **1 -** Score thousands of genes with PLEIOVAR using genetic data and phenotypic data from a cohort. For each gene we estimate a score based on the chi-square distribution and obtain the corresponding p-values.
- **2 -** Generate a combined score for a small set of genes (gene network) based on a summation of the chi-square results for each gene from PLEIOVAR.

Goal: Run PLEIOVAR for a given cohort – Main Steps

- **Two main steps for PLEIOVAR:**
  - **Pre-processing**
  - **Association**
- **Scoring a gene set**

Pre-processing

- Takes input **genotype** datasets and **CONFIG** file with user specified parameters and generate **PC-SNP** files for every gene.
- Takes input **phenotype** dataset and generate **PC-Trait** file.
- Computationally Intensive (> 100 hours with a single CPU) using multiple CPU's is highly recommended.
- Processing done with Perl and R

Association

- Takes **PC-Trait** file and merge with the **PC-SNP** file for each gene.
- For each gene, generate a z^2 statistics for each PC-Trait vs. PC-SNP combination
- Obtain the genomic control (λ) of each PC-Trait by using the median z^2 across each PC-Trait.
- Correct the z^2 statistics for each PC-trait by dividing it by the corresponding λ and obtain the adjusted z^2

Association

- Sum the adjusted z^2 statistics (**SSQ**) for each gene
- Get the degrees of freedom (**DF**) for each gene, which is the number of PC-Traits vs. PC-SNPs
- Take **SSQ** and **DF** for each gene and based on the chi-squared statistics, obtain the corresponding p-value
- Save all the association results into the **results file**.
- Computationally fast (5 minutes with a single CPU)
- Processing done in R

Scoring a Gene Set

- Take a set of genes (generally from a gene network) specified by the user.
- Get PC-SNPs for all the genes in the set
- Estimate between genes variance inflation factor (VIF) for the gene set
- Find the **SSQ** for each genes in the **results-file**

Scoring a Gene Set

- Generate **Total_SSQ** and **Total_DF** for the gene set using the SSQ and DF for each gene from the results-file.
- Convert **Total_SSQ** and **Total_DF** into a z-score equivalent for the chi-square (Canal's transformation) https://www.semanticscholar.org/paper/A-normal-approximation-for-the-chi-square-Canal/22603bea0bcc5110e4cc9ba29df96fd06daa262c
- Correct the z-score for variance inflation
- Generate the corresponding p-value for the gene set.

Pre-processing: Input Files

- Cohort **phenotype** dataset.
- Cohort **genotype** dataset.
- Set of **RefSeq** genes with chr, start position, end position and gene name.
- User pre-defined parameters (**CONFIG** file).

Pre-processing: Output Files

- Simplified **genotype** file for each chromosome
  - For example: round dosages to nearest hundredths
- Simplified **genotype** file for every gene
- Subsets of **RefSeq** genes in multiple blocks (for parallel processing)
- **PC-Trait** file
- **PC-SNP** file for every gene

Association: single CPU

- Since the association step is much faster than the pre-processing step, we use only a single CPU.
- **Input:** We use the **PC-Trait** file and **PC-SNP** files for every gene as inputs for the association
- **Input:** We use **RefSeq** file as input to generate PLEIOVAR for each gene
- **Output:** Results-file is generated with the PLEIOVAR p-value for each gene

Detailed Steps

- 0-Define parameters in CONFIG file
- 1-Convert vcf files to SAMPLE files
- 2-Convert SAMPLE files to MICROFILES
- 3-Index MICROFILES (CUT_{chr} and TABLE_CUT_{chr} files
- 4-Cut and join MICROFILES into gene region (assemble) files
- 5-Generate PC-SNP files for each gene
- 6-Generate PC-trait file
- 7-Generate p-values and sort by p-values and save it into results_file
- 8- Generate z^2 Tables for each gene

Main Input Files

- **CONFIG** file: Insert your parameters
- **Gene definition** file (subset of **RefSeq** file)
  - Save it in **All_genes_hg19**
- **Trait** file: Save traits into **mainpheno.dat** file
- **Genotype file**: Transform dosages from *.vcf files into integers for each chromosome and save into: **SAMPLE_{chr}** (**SAMPLE_1**, **SAMPLE_2**, …, **SAMPLE_22**) following the specified format (see ahead)

Main Output files

- **SAMPLE** files (genotype files) if we start with vcf files
- **MICROFILES** (small subsets of **SAMPLE_{chr}**)
- **Assemble** files (Gene region files – one for each gene)
- **PC-SNP** files (one for each gene)
- **PC-Trait** file
- **final-results file** (p-values for every gene)
- z^2 Table files (one for each gene used to calculate the SSQ and DF for each gene.
- Job files of the form *.lst (for parallel processing)
- Block files (equal sub-sets of genes from ***All_genes_hg19***)
- Indexing files (**CUT** and **Table_CUT**)

Input File Formats

- **CONFIG** file
  - **Line 1:** Folder of vcf files (to be converted to Sample_{chr})
  - **Line 2**: Output folder for all output datasets used in pre-processing
  - **Line 3**: Precision of genotype (# of decimals) (default is 2)
  - **Line 4**: Maximum number of SNPs in a MICROFILE (default is 500)
  - **Line 5**: Number of parallel jobs (each processes a block of genes which are a subset of All_genes_h19, with a set of genes in each block) (default is 64)
  - **Line 6**: Number of kb for gene extension (default is 50)
  - **Line 7**: PC-SNP variance explained cutoff (default is 0.75)
  - **Line 8**: PC-SNP Minor allele frequency cutoff (default is 0.005)
  - **Line 9**: PC-Trait variance explained cutoff (default is 0.75)
  - **Line 10**: Flag for variance inflation correction (default is 1)
  - **Line 11**: Output folder for the association step (PC-Traits.txt and results-file)

Input File Formats

- **Refseq file** (All_genes_hg19)
  - No header
  - Tab delimited
  - Values for: *chromosome#*, *start pos.*, *end pos.*, *gene name*

1	**11873**	**14409**	**DDX11L1**
1	**14361**	**29370**	**WASH7P**
1	**34610**	**36081**	**FAM138A**
1	**69090**	**70008**	**OR4F5**
1	**134772**	**140566**	**LOC729737**

Input File Formats

- **Trait** file (copy to mainpheno.dat)
  - Tab delimited
  - Header: "ID" $Trait_1$ $Trait_2$ … $Trait_k$
  - Other lines: Values for the header
  - ID $Trait_1$ $Trait_2$ … $Trait_k$
  - Example (using residuals from normal-inverse transformed traits).
  - For cross-sectional phenotype data, use residuals from regression (each ID has a row with one residual per trait)
  - For longitudinal phenotype data, use random-intercept estimates from mixed-model or ANOVA repeated measures (each ID has a row with the random-intercept estimate for each trait).

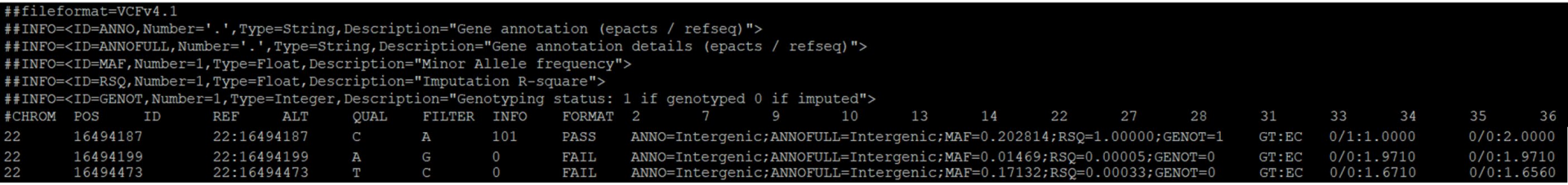
Input File Formats

mainpheno.dat

ID	HDL	LDL	TRI
2	-0.19	-0.15	0.69
7	0.20	-0.05	-0.05
8	0.13	-0.11	-0.06
9	-0.39	0.40	0.06
10	0.01	0.55	-0.91
13	0.13	-0.32	-0.31
14	0.81	0.11	-0.30
17	0.61	0.54	0.00
22	-0.22	0.11	-0.97
27	0.65	-0.68	-0.79
28	-0.18	0.06	0.09
31	0.01	0.86	0.84
33	0.03	-0.63	0.45
34	-0.67	-0.37	-0.24

VCF_{chr} to SAMPLE_{chr} File

Screenshot of SardiNIA vcf file for chromosome 22



```
##fileformat=VCFv4.1
##INFO=<ID=ANNO,Number='.',Type=String,Description="Gene annotation (epacts / refseq)">
##INFO=<ID=ANNOFULL,Number='.',Type=String,Description="Gene annotation details (epacts / refseq)">
##INFO=<ID=MAF,Number=1,Type=Float,Description="Minor Allele frequency">
##INFO=<ID=RSQ,Number=1,Type=Float,Description="Imputation R-square">
##INFO=<ID=GENOT,Number=1,Type=Integer,Description="Genotyping status: 1 if genotyped 0 if imputed">
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  2       7       9       10      13      14      22      27      28      31      33      34      35      36
22      16494187        22:16494187     C       A       101     PASS    ANNO=Intergenic;ANNOFULL=Intergenic;MAF=0.202814;RSQ=1.00000;GENOT=1    GT:EC   0/1:1.0000      0/0:2.0000
22      16494199        22:16494199     A       G       0       FAIL    ANNO=Intergenic;ANNOFULL=Intergenic;MAF=0.01469;RSQ=0.00005;GENOT=0     GT:EC   0/0:1.9710      0/0:1.9710
22      16494473        22:16494473     T       C       0       FAIL    ANNO=Intergenic;ANNOFULL=Intergenic;MAF=0.17132;RSQ=0.00033;GENOT=0     GT:EC   0/0:1.6710      0/0:1.6560
```

- 6 lines of description header
- 1 lines with header with field names
- 157,797 lines with SNP information
- In this study we selected only SNPs with FILTER = “PASS”

SAMPLE_{chr} File Formats

- **Genotype** File outputs to **SAMPLE**_{chr})
  - Space delimited
  - Header: “CHROM” “POS” id_1 id_2 … id_n
  - Other lines: Values for header

CHROM	POS	2	7	9	10	13	14	22
22	16849535	0.36	0.95	1.53	1.53	0.94	0.36	0.95
22	16849573	0.34	0.93	1.53	1.52	0.93	0.34	0.93
22	16849681	0.32	0.96	1.61	1.6	0.97	0.32	0.96

- The vcf to SAMPLE conversion can be skipped as long as we have the SAMPLE files with the format above.

Slurm

- To run multiple jobs, we first needed to download the Slurm application.

  Slurm: https://slurm.schedmd.com/documentation.html

- Then we use a shell program named runjobs.pl (included with all the other code) to run multiple jobs in the *.lst files (ahead).

1- SAMPLE_{chr}

- Make Pipeline folder **/data/Osorio/PIPELINE**
- Go to Pipeline folder
- Open **CONFIG** file and fill in **lines 1,2 and 3**
  - **Line1:** Input folder (where vcf files are located)
  - **Line 2:** Output folder (where simplified SAMPLE_{chr} files will be located)
  - **Line 3:** Number of decimals to round allele dosages for **SAMPLE_{chr}** files
- **Line 1:**
  /data/Osorio/VALIDATION_PLEIOVAR_EPACTS/VCF_files
- **Line 2:**
  /data/Osorio/VALIDATION_PLEIOVAR_SARDINIA_NEW
- **Line 3:**
  2
- Run "**perl MAIN_PLEIOVAR_1.pl**"

perl MAIN_PLEIOVAR_1.pl

- Runs the program (runjobs.pl) which runs parallel jobs for each chromosome, processing for each line "converter_jobs.lst" for each CPU.

```
perl ./CONVERTER.pl 1
perl ./CONVERTER.pl 2
perl ./CONVERTER.pl 3
perl ./CONVERTER.pl 4
……

perl ./CONVERTER.pl 20
perl ./CONVERTER.pl 21
perl ./CONVERTER.pl 22
```

- Program CONVERTER.pl uses the chromosome number as the main argument and generates the SAMPLE file for each chromosome which has more summarized dosage information.

2 – MICROFILES

- At this stage we have already generated the **SAMPLE_**{chr} files
- Now our next step is to generate the **MICROFILES**, which are subsets of the SAMPLE files of the form **SAMPLE_{chr}_{chunk}**, which contain a much smaller number of SNPS.
- Open **CONFIG** file and fill in **line 4** for max. number of SNPs in each **MICROFILE**
  - **Line 4**: **500** (default)
  - Each MICROFILE has with a header with “CHROM”, “POS” and the ID’s.
  - Each line corresponds to a SNP, starting with the chromosome, position and dosages of the SNP corresponding to each ID.
  - In addition to the header, each file is expected to have 500 lines (SNPs) but it could have less than 500 if it was originated near the last block in the corresponding SAMPLE_{chr} file.
- Run “**perl MAIN_PLEIOVAR_2.pl**”

perl MAIN_PLEIOVAR_2.pl

- This program runs the parallel jobs (one for each chromosome) in the file “microfiles_jobs.lst” (see below), where each CPU is assigned one command in each line.

```
perl ./CUTTER.pl 1
perl ./CUTTER.pl 2
perl ./CUTTER.pl 3
perl ./CUTTER.pl 4
……..
perl ./CUTTER.pl 20
perl ./CUTTER.pl 21
perl ./CUTTER.pl 22
```

- The main output are thousands of the form **SAMPLE_{chr}_{chunk}.**
- For example, **SAMPLE_22_18** which corresponds in this example to 500 SNPs from chromosome 22 and chunk 18.

3 – UCSC Gene Regions

- **Transform UCSC file to All_genes_hg19 (Run program hg19_generator.pl)**
  - We can use the already generated **All_genes_hg19** file
  - If we want to generate **All_genes_hg19** from scratch, we do:

    perl hg19_generator.pl in {folder where All_genes_hg19 is located **} This program can become outdated as files in the UCSC are often being updated.**
- **NOTE:** all we need to do is to use **All_genes_hg19** (tab delimited) already in **/data/Osorio/PIPELINE** (example below) and remember that **All_genes_hg19** does not have a header

chromossome	beg_pos	end_pos	GENE
1	11873	14409	DDX11L1
1	14361	29370	WASH7P
1	34610	36081	FAM138A
1	69090	70008	OR4F5
1	134772	140566	LOC729737
1	323891	328581	LOC100132062

3 – UCSC Gene Regions

- **We further reduced All_genes_hg19 such that**
  - Each gene should have a unique **chr**, **start** and **end**
  - One gene per row (eliminate gene duplicates)
    - If the same genes have more than one start/end position, we select the gene based on the shortest region)
  - Total number of cases are reduced from ≈ **30K** to ≈ **22.5K**

4 - Indexing files before generating gene region files

- **CUT_{chr}** files are generated by using the corresponding vcf file and extracting the first two columns (contains the chromosome and position of each SNP in the **SAMPLE_{chr}** file.
- In this particular case we extract the first two columns from the **SAMPLE_{chr}** files.
- We will now use the code and corresponding job files:
  - **cut_generator.pl , jobs_cut_generator.lst**
  - **locator_chunks.pl , Jobs_locator_chunks.lst**
- Run "**perl MAIN_PLEIOVAR_3.pl**" which generates **CUT_{chr}** files, one for each chromosome.

CHROM	POS
1	752566
1	752721
1	753405
1	753474

perl MAIN_PLEIOVAR_3.pl

- This program runs the parallel jobs (one for each chromosome) in the file “jobs_cut_generator.lst” (see below), where each CPU is assigned one command in each line.

```
perl ./cut_generator.pl 1
perl ./cut_generator.pl 2
perl ./cut_generator.pl 3
perl ./cut_generator.pl 4
……..
perl ./cut_generator.pl 20
perl ./cut_generator.pl 21
perl ./cut_generator.pl 22
```

- The program **cut_generator.pl** generates the **CUT_{chr}** file for each chromosome.

4 - Indexing files before generating gene region files

- Run "**perl MAIN_PLEIOVAR_4.pl**" which generates **Table_CUT_**{chr} files, one for each chromosome.
- **Table_CUT_**{chr} has the **chr**, **chunk#**, first SNP position (first_pos), last SNP position (last_pos)

CHROM	CHUNK#	first_pos	last_pos
1	1	752566	892745
1	2	893194	995481
1	3	996184	1106946
1	4	1107147	1185260

- The **Table_CUT**_{chr} files will be needed when running the next step which generates a file for each gene region as they are used to speed up the processing/reading of the MICROFILES.

perl MAIN_PLEIOVAR_4.pl

- This program runs the parallel jobs (one for each chromosome) in the file “jobs_locator_chunks.lst” (see below), where each CPU is assigned one command in each line.

```
perl ./locator_chunks.pl 1
perl ./locator_chunks.pl 2
perl ./locator_chunks.pl 3
perl ./locator_chunks.pl 4

……..

perl ./locator_chunks.pl 20
perl ./locator_chunks.pl 21
perl ./locator_chunks.pl 22
```

- The program **locator_chunks.pl** generates the **Table_CUT_{chr}** file for each chromosome.

5 – Assembling the gene region files

- Open **CONFIG** file and fill in **line 5**, the number of blocks (in example below we specified 64 blocks), so program takes the gene list file **All_genes_hg19** and breaks it into 64 smaller files (blocks) of about equal size to one another (except the last block which is likely to have less genes).
  - **Line5: 64**
- Fill in **line 6** which is the extension in Kb before the gene start position and after the end position.
  - **Line 6**: **50**
- Run "**perl MAIN_PLEIOVAR_5.pl**" which generates the blocks, then generates the **jobs_assemble.lst** file which assembles a gene file for each gene region (in each block file).

perl MAIN_PLEIOVAR_5.pl

- The program Blockmaker.r generates the block files.
- Next, we create the **jobs_assemble.lst** file which assembles a gene file for each gene region (in each block file).

R --slave --no-save --no-restore --no-environ --silent --args block1 < Assemble.r
R --slave --no-save --no-restore --no-environ --silent --args block2 < Assemble.r
R --slave --no-save --no-restore --no-environ --silent --args block3 < Assemble.r
……
R --slave --no-save --no-restore --no-environ --silent --args block64 < Assemble.r

- Assemble.r locates the MICROFILES which include each gene region.
- Next, it appends the rows from different chunks and joins into one.
- Next, it cuts the SNPs (lines) outside of the gene region, to save only the dosages within each gene region into a file.
- Saves into a file of the form **{gene}_assembled**

6 – Generating PC-SNP files

- Open **CONFIG** file and fill in lines **7** and **8**. Line **7** is PC-SNP cutoff for variance explained and line **8** is the minor allele frequency cutoff (SNPs that have a MAF lower than the cutoff will not be used to run the principal components and .generate PC-SNPs)
  - **Line 7: 0.75**
  - **Line 8: 0.005**

- Run "**perl MAIN_PLEIOVAR_6.pl**" which generates the **jobs_PC-SNPs.lst** file which runs the PC-SNPs.r for each gene region (in each block file).

perl MAIN_PLEIOVAR_6.pl

- Jobs_PC-SNP.lst

```
R --slave --no-save --no-restore --no-environ --silent --args block1 < PC-SNP.r
R --slave --no-save --no-restore --no-environ --silent --args block2 < PC-SNP.r
R --slave --no-save --no-restore --no-environ --silent --args block3 < PC-SNP.r
.....................................................................
R --slave --no-save --no-restore --no-environ --silent --args block62 < PC-SNP.r
R --slave --no-save --no-restore --no-environ --silent --args block63 < PC-SNP.r
R --slave --no-save --no-restore --no-environ --silent --args block64 < PC-SNP.r
```

- Each of the 64 CPU runs a block of genes through the program PC-SNP.r
- PC-SNP.r saves in RDS format, the PC-SNP files, variance explained, cumulative variance explained and loadings of the PC-SNPs

7 - Generating PC-traits

- Copy phenotype file **mainpheno.dat** (with header) to pipeline folder
- Enter parameters before generating PC-Traits.txt and before running the association step.
- Open **CONFIG** file and fill in lines **9** the cutoff criteria in variance explained for PC-Traits.
  - **Line 9: 0.99**
- Fill in line **10** with the flag (1 or 0) for correcting for variance inflation, with **1** (default) indicating to correct for inflation.
  - **Line 10: 1**
- Enter the output folder in line **11** indicating the folder where PC-Traits.txt and results-file are saved.
  - **Line 11: /data/Osorio/PLEIOVAR_TEST/OUTPUT**

7 - Generating PC-traits

- Run "**perl MAIN_PLEIOVAR_7.pl**" which takes mainpheno.dat and generates the PC-Traits.txt file (no header).

Mainpheno.dat

ID	HDL	LDL	TRI
2	-0.19	-0.15	0.69
7	0.20	-0.05	-0.05
8	0.13	-0.11	-0.06
9	-0.39	0.40	0.06
10	0.01	0.55	-0.91
13	0.13	-0.32	-0.31
14	0.81	0.11	-0.30

PC-Traits.txt

ID	PCT1	PCT2	PCT3
2	-0.54	-0.45	0.23
7	0.11	0.09	0.15
8	0.14	0.01	0.12
9	-0.35	0.06	-0.45
10	0.47	0.71	-0.65
13	0.45	-0.08	0.09
14	0.37	0.65	0.45

8 - Running Association

- Next step is to run the association step which generate a p-value for every gene region.
- Run: **perl MAIN_PLEIOVAR_8.pl**
  - This will generate the results-file (unsorted)
- A Table of z^2 statistics is generated for each pairwise combination of **PC-Traits** and **PC-SNPs**
  - z^2 are adjusted (or not) for variance inflation
  - SSQ is obtained by adding the adjusted z^2 for each gene and DF is estimated by **#PC-traits** x **#PC-SNPS**.
  - Under a chi-squared distribution with **DF** degrees of freedom, a p-value is generated for the corresponding **SSQ**.
  - Output file for gene regions corresponding to each CPU is saved into the **results-file**

9 – Sorting file

- Run “**Perl MAIN_PLEIOVAR_9.pl**” which will sort **results-file** by p-value to **sorted_results-file.**

10 – Generating z2_Tables

- Run "**Perl MAIN_PLEIOVAR_10.pl**"
- Will take genes and their corresponding Z2_Table where adjusted Z^2 are saved in a file for each gene
  - Tab delimited
  - RDS formats