SNPs and exons
Use of scripts

Contents

1 Preamble

The aim of these scripts is finding the p-value (from GWAS summary statistics) transferred from SNPs to genes passing by Z-score (see A way to analyse GWAS data using exons) . The elements to be computed are :

- the distribution γ linking density of SNPs and distances from SNPs to exons
- the weighting of Z-score of p-value by the probability density function
- the result about genes under different forms

Chromosomes are numbered from 1 to 24, 23 for X and 24 for Y.

The hub file containing all parameters is SNP_exon_param.py for one set of GWAS data. After updating paths and start files, it is filled as the scripts progress.

2 Parameters

2.1 SNP_exon_param.py

First, create a directory structure beginning by root and name the source data by an acronym or a reference of GWAS:

```
your_root
└─ SNP_exons
   ├─ ref
   │  └─ NIHsequence
   └─ source
      ├─ dta
      ├─ LD_corr
      └─ result
```

Their contents are:

- ref: reference files about exons and genes and source of data about them, and, in NIHsequence, the sequence files extracted from NIH used for building reference files
- dta: the source of GWAS data and intermediary files used for computing results
- LD_corr: the files containing the correlation coefficient of linkage disequilibrium under diagonal matrix and files of genes after Weight_Z_SNP_by_gene.py, useless if LD correlation is computed by fitting and not extracted
- result: list of genes and result files

The parameters needed to read the SNP file from GWAS summary statistics: source file, title line and positions of fields:

- name of SNP source in directory dta
- presence of a title line
- positions of fields in every line

As the scripts are executed, complete the following parameters:

- limits of kept p-values after eliminating outlier p-values (inf_pv, sup_pv)
- shape and scale of γ distribution
- LD correlation coefficient by default (def_LDcorr)

2.2 SNP_exon_utils.py

The script SNP_exon_utils.py is necessary for the process, but it is not directly executed. It contains Utilitary function for reading files, fitting list of values, computing used functions and drawing graphs and histograms..

It contains the used here function reading the linkage desiquilibrium correlation coefficients (see Formats of Files). Il can be modified if an other format is used.

When the LD correlation coefficients are computed by a fitted function, this function is in SNP_exon_utils.py and can also be modified.

3 Scripts

3.1 exons_from_NIH_nh.py

The script exons_from_NIH_nh is run once to get the base positions of exons.

Input:
23 sequence files named sequence_chr01.txt to sequence_chr24.txt (23 for X and 24 for Y) extracted from NIH by:

1. www.ncbi.nlm.nih.gov/genome/51
2. in column RefSeq click on chromosome.
3. in box 'Send to' choose coding sequence, file, format FASTA nucleotide

Function:
Create 3 files containing positions of exons and genes and an abnormality file.

Output:
A file for exon position 'exon_pos.txt' and a file for gene position 'gene_pos.txt' containing: chromosome, begin base position, end base position, NIH name of genes, exon order for exon or exon number for gene. A third file 'exon_abn.txt' provides information on abnormalities of regular position of fields.

3.2 SNP_analyse_outliers.py

Python cannot compute the Z-score for p-value <5.552e-17. In that case, an error occurs in the next script SNP_bp_computed_Z.py).

Input:
Update in parameters to read the file of SNPs, the column number of fields (SNP_chr, SNP_name, SNP_bp, SNP_pv, numbered from 0) and if there is a line title. Make sure to correctly number chromosomes in the file of SNP (from 1 to 24, X:23, Y:24). If not, correct the file of SNPs.

Function:
Give information about SNPs to be added in parameters. Check if every line is readable and the uniqueness of name of SNP, if not 'b' is added to use the name as key (polymorphism with 2 nucleotides)

Output:
Abnormality messages. Information to eliminate outlier p-values: histogram and quantiles. So, minimum and maximum values are updated in parameters (inf_pv, sup_pv). List of excluded and duplicated SNPs (if they have the same name) which are renamed and kept (log option by default).

This script is to be run twice at least. Once the p-value range is chosen, the correlation coefficient is computed and updated in parameters.

3.3 SNP_bp_computed_Z.py

Input:
File of SNPs

Function:

Compute Z-score

Output:
An intermediary file '_Zpv.txt' containing SNP names, p-values, Z-score, chromosose and base position. P-values are inside the limits previously updated, which can take time. Adjust limits of SNP p-values if infinity as value.

3.4 SNP_exon_distances.py

Input:
The SNP file and the two previously written files: exon_pos.txt, gene_pos.txt

Function:
Create a file of SNPs given the relative position of SNPs to exons in the the 7 cases:

- 0: SNP inside exons
- 1: SNP inside the same gene
- 2: SNP inside the the 2 nearest genes
- 3: SNP inside this gene but outside the other nearest gene
- 4: SNP outside this gene but inside the other nearest gene
- 5: SNP between exons and outside the 2 nearest genes
- 6: SNP at the extremities and outside all genes

The algorithm is:

- create a sorted by position list by merging SNPs and exons
- explore this list in forward and reverse direction
- affect the case from the relative position of SNPs and exons and compute distances

Output:
The file '_sed.txt' as SNP exon distance is the central file for weighting SNP p-values. Every SNP is twice, even the records are identical. The fields are: chromosome, relative position case, SNP, gene, exon order and distance SNP to exon.

Counts of SNP positions and cases are displayed for checking. The SNPs outside range [inf_pv, sup_pv] are not taken into account.

3.5 SNP_exon_distance_fit.py

Input:
The previously written file '_sed.txt'

Function:
Analyse the relative positions of SNPs to exons and illustrate them with graphs:

- minimal distance between SNP and exon / normalized cumulative number, all SNPs, SNPs inside exons, SNPs outside exons
- minimal distance from all SNPs to exons / normalized cumulative number fitted with γ law
- minimal distance from SNPs inside gene to exons / normalized cumulative number fitted with γ law
- histogram of minimal distance from SNPs inside gene to exons
- Q-Q plot minimal distance from SNPs inside gene to exons with γ law

Compute the parameters of γ distribution.

Output:
Give information about fit. The parameters scale and shape of gamma distribution must be added in the parameter file. They are used to weight p-values.

3.6 SNP_exon_dist_Z_plot.py

Input:
Files ’_sed’ and ’_pvZ’.

Function:
Draw graph Z-score of SNPs with effect in function of minimal distance of SNPs to exons anf histogram of extreme values

Output:
The scatter graph of Z-score in function of min distance SNP to exon and histogram focusing on the highest values of Z-score to target the analysis of effect. The default part of SNPs can be changed in the main.

3.7 Weight_Z_SNP_by_gene.py

Input:
Files ’_sed’ and ’_pvZ’.

Function:
Create files needed to compute Z-scores of genes from Z-scores of SNPs and weighting elements

Output:
One file by combination of genes and position od SNP inside or between genes (name of gene_position_g.txt) and a list of genes which can be used to extract LD correlation coefficient (_list_g.txt). The gene file contains SNPs (eventually twice), their Z-scores and the corresponding weights from γ law.

3.8 R and Python Scripts to use LDlinkR for extracting LD correlation coefficients

This part only concerns the extracting LD correlation coefficients and and not LD correlation coefficients got by fitting. Here, the sofware used to extract LD correlation coefficients is LDlinkR.
(https://cran.r-project.org/web/packages/LDlinkR/vignettes/LDlinkR.html).
An other source can be used, but the file format may be not the same and the script must be adapted.
the scripts are:

3.8.1 LD_corr.R

extract from LDlinkR the correlation matrix for one gene_position, which is automatized by the both following scripts , only the triangular matrix is saved in the file ‘name of gene_position_c.txt’

3.8.2 update_todo.R

update the _list_g.txt file by setting flag to 1 of already extracted gene_position and create the temporary file _list_g$.txt contining the gene_position to extract the LD correlation of their SNPs

3.8.3 LD_corr_by_gene.R

proceed extraction of gene_position in _list_g$.txt, if an uncorrectable errors occurs, set in _list_g.txt the corresponding flag to -1 otherwise try again the extraction after creating _list_g$.txt byt the previous script

3.8.4 LD_corr_manage_number_limit.py

Some errors may be corrected: a part of SNPs unknown, oversize, disponibility of servers. For oversize, the script identify the gene_position beyond limits and create from a list new files in limits in abnormality directory which must be copied from abnormality directory to LD_corr directory.

3.8.5 LD_corr_analyse_default_value.py

Analyse LD correlation values in the form of histogram and compute the LDcorr_defaut to input in param file (value of LD correlation if it is absent). The number of SNPs must be input in the script to simplify it.

3.9 Z_gene_from_LDcorr_files.py

Input:
Files 'gene_pos.txt', 'name of gene_position_g.txt' by gene and 'name of gene_position_c.txt' by genes (position of SNPs 0: inside, 1:outside)

Function:
Compute Z-score of genes by weighting Z-score of SNPs (Z) with probility distribution function of γ (W) and normalized by LD correlation matrix previous extracted.

Output:
A file of genes containing: chromosome, gene (can be twice if internal and splicing effect), begin of gene as base position, end of gene as base position, computed Z-score of gene, p-value computed from Z-score and Int_Spl with 0 if internal effect and 1 if splicing effect.
This file is named source_Zgc.txt when LD correlation is detailed by gene. On console, the name of gene is followed by 'd' when LD correlation file is absent. If it is present and so used, it is followed by 'c'.
Comment: the chromosomes X (23) and Y (24) having common genes, some genes expected in chromosome 23 (female phenotype) can appear in chromosome 24 and mutually for male phenotype.

3.10 Z_gene_from_fit_LDcorr.py

Input:
File sed_file and pvZ_file

Function:
Compute Z-score of genes by weighting Z-score of SNPs (Z) with probility distribution function of γ (W) and normalized by LD correlation matrix computed by $\sqrt{\frac{1}{1+0.017*\sqrt{distance}}}$ (distance between SNP, see document for explanations). This formula can be replaced in utils script (SNP_exon_utils.py).

Output:
A file of genes containing: chromosome, gene (can be twice if internal and splicing effect), begin of gene as base position, end of gene as base position, computed Z-score of gene, p-value computed from Z-score and Int_Spl with 0 if internal effect and 1 if splicing effect. This file is named source_Zgd.txt.

3.11 Manhattan_plot.py

From the file '_Zgc' or '_Zgd' (LD correlation default) in function of option, draw the Manhattan plots for the both cases 'Int_Spl'.

3.12 compare_Z_genes.py

Compare Z-scores of common genes from two files in result dir. The names of files and the format must be input in the script (position of gene name and Z-score if not a result of SNP exon).

4 Formats of Files

4.1 ref/exon_pos.txt

0: chromosome
1: begin base position
2: end base position
3: NIH name of gene
4: exon order

4.2 ref/gene_pos.txt

0: chromosome
1: begin base position
2: end base position
3: NIH name
4: exon number

4.3 dta/source_pvZ.txt

0: SNP_name
1: SNP p-value
2: SNP Z score
3: chromosome
4: base position

4.4 dta/source_sed.txt

Every SNP is twice, even the records are identical, fields for twice SNP:
0: chromosome
1: relative position case
2: SNP
3: gene
4: exon order
5: distance SNP to exon

4.5 LD_corr/files of genes: name of gene_position_g.txt

File by every gene, SNPs may be twice:
0: SNP name
1: Z-score of SNP
2: weight from γ law

4.6 LD_corr/_list_g.txt

CSV file used to extract LD correlation between SNP of genes from the file list above with header :
"gene_int_spl";"snp_nb";"flag" numbered as in R:
1: gene_int_spl: concatenation of gene name and position inside exon (0) or between exons (1)
2: snp_nb: number of not indentical SNPs influencing the gene
3: flag to show to show to extracting script what it can do.

The meanings of flags are:
0: extract the LD correalation coefficient of SNP list in gene file
1: extraction went well or there is only one SNP so no extraction
-1: to input by hands, extraction went wrong, try again after correction, if not extracted, default value is taken.

The temporary filr _list_g$.txt updated by the R script has the same format.

4.7 LD_corr/list of gene_name_c.txt

File of linkage desiquilibrium matrix under triangular form, format csv: $SNP1; c_{1,1}; c_{1,2}; ......; c_{1,n}\ blank$
$SNP2; c_{2,1}; c_{2,2}; ...; c_{2,n-1}\ blank$
$SNPn-1; ; c_{n-1,1}\ blank$
$SNPn\ blank$
The matrix is completed before computing. This format spares size and results from using R to extract LD correlation. An other format can be use, but the read function must be modified.

4.8 dta/source_Zgc.txt and _Zgd.txt

0: Chr: chromosome
1: Gene: gene influenced by SNPs
2: BPbeg: begin base position
3: BPend: end base position
4: Z : Z-score of gene
5: PV: p-value of gene
6: Int_Spl: 0 or 1, 0: SNP inside exon, 1: SNP inside gene influencing splicing