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:

```
root
├── dta
├── ref
│   └── NIHsequence
└── 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 sources of GWAS data and intermediary files used for computing results
- result: list of genes and result files

The names of GWAS data are gathered in an array to cope with several sources. They are numbered. Their own parameters are input in 'if' or 'elif' blocks corresponding to number

their number selects the parameters needed to read the SNP file: title line and positions of fields:

- name of SNP file 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)
- correlation coefficient between p-values (SNP_R)
- shape and scale of γ distribution

To get detailed analysis, input these values:

- top number and/or file of list of genes

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.py

Python cannot compute the Z-score for p-value <5.552e-17. In that case, an error occurs in the next script SNP_compute_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). The log option gives the list of excluded and duplicated SNPs (if they have the same name) which are renamed and kept.

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_compute_Z.py

Input:
File of SNPs

Function:
Compute Z-score

Output:
An intermediary file '_Zpv.txt' containing SNP names, p-values and Z-score with p-values inside the limits previously updated. Adjust limits of SNP p-values if infinity as value

3.4 SNP_exon_distance.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_analyse.py

Input:
The previously written file '_sed.txt'

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

Input:
Files '_sed' and '_pvZ'

Function:
Compute Z-score of genes by weighting Z-score of SNPs with probility distribution function of γ parameterized by shape and scale from prameter file

Output:
A file '_Zgn' 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 InOut with 0 if internal effect and 1 if splicing effect.
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.

3.7 SNP_on_genes_top.py

Input:
Files '_sed' and '_pvZ'. Top and eventually gene_file in parameter file if top == 0.

Option:
Option determined by the value of top to enter in parameter file:

- top genes sorted by Z-score for internal effect (SNPs inside exon) and splicing effect (SNPs outside exon and inside genes)
- p-value of a list of genes red in a file in result directory as gene_list in parameters

Output:
Top genes are displayed
If top == 0, result in a file identified by gene_list and name of GWAS data. An array of contributions of Z-scores of effective SNPs to Z-score of genes for internal effect and splicing effect in a result file.

3.8 SNP_exon_dist_Z.py

Input:
Files '_sed' and '_pvZ'. Top and eventually gene_file in parameters if top == 0.

Option:
Option determined by the value of top to enter in parameter file:

- Draw the scatter graph of Z-score in function of min distance SNP to exon and 3 graphs (top > 0)
- Draw the scatter graphs of SNPs inside genes in gene_list by gene (top = 0)

Output:
The scatter graph of Z-score in function of min distance SNP to exon and 3 graphs 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. The top list of SNPs sorted by Z score is displayed.

Once the gene_list is red, the positions of SNPs inside every gene are obtained by scatter graphs by gene. The list of involved SNPs by gene is displayed .

3.9 SNP_Z_on_gene.py

Input:
Files '_sed', '_pvZ' and _Zgn and a threshold Z_SNP_thr input in parameters

Output:
Tabular list where p-value Z are above the threshold fixed with SNP_exon_dist_Z.py. The header is: SNP, SNP Z-score, effect of SNP on gene by Z-score transferred to gene, gene and gene Z.
Comment: the sum of effects on a gene may be greater than gene Z-score, the Z-score of another SNP not in the list and having an effect on the gene can be negative.

3.10 Manhattan_plot.py

From the file '_Zgn', draw the Manhattan plots for the both cases 'InOut'.

4 Supplement

Some scripts for other analyses which would be interesting.

4.1 exon_gene_analyse.py

Explicit geometric intersection between genes and exons and show the repartition of exons along the genome.

4.2 SNP_gap_analyse.py

Show the probability law of the distribution of gaps between SNPs along the genome.

4.3 gene_compare.py

Gather the files '_Zgn' in an array to compare results of several GWAS identified by a list of data numbers in parameters .

4.4 fixed_param.py

Once the parameters are set, generate the the files needed to compute p-values of genes ('_pvZ.txt', '_sed.txt' and '_Zgn.txt')

5 Utilities

These scripts are necessary for the process, but they are not directly executed.

5.1 SNP_exon_utils.py

Utilitary function for reading files, fitting list of values, computing used functions and drawing graphs and histograms.

5.2 SNP_on_genes_utils.py

This class processes weighting Z-scores to computed Z-scores of genes and the contribution by SNPs.

June 16, 2022 by daniel.rovera@gmail.com - Institut Curie