SNPs and exons
Use of scripts

1 Preamble

The aim of these scripts is finding the p-value (from GWAS) transferred from SNPs to genes (see analyse_GWAS_by_exon.pdf). 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

The central file containing all parameters is SNP_exon_param.py. After updating paths and start files, it is filled as the scripts run.

2 Parameters

2.1 SNP_exon_param.py

First, create a directory structure beginning by root and name the source data by an acronym 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
- dta: intermediary files used for computing results and the source of data about SNPs
- result: list of genes and result files

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
- top number and/or file of list of genes

3 Scripts

3.1 exons_from_NIH_nh.py

Input:
23 sequence files named sequence_chr01.txt to sequence_chr23.txt (right chromosome in function of sex) extracted from NIH by:

- www.ncbi.nlm.nih.gov/genome/51
- in column RefSeq click on chromosome.
- 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.

3.2 SNP_analyse.py

Python cannot compute the Z-score for p-value <5.552e-17 (if not error in next scriptSNP_compute_Z.py).

Input:
Update in parameters to read the file of SNPs, the column number of fieds and if there is a line title.

Function:
Give information about SNPs to be added in parameters

Output:
Information to eliminate outlier p-values: histogram and quantiles. So, minimum and maximum values are updated in parameters.

Once the p-value range 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.

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

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 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 graph. Compute the parameters of gamma distribution.

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

3.6 SNP_on_genes_top.py

Input:
Two files '_sed' and '_pvZ'

Option:
Option to enter in parameter file:

- top genes sorted by Z-score for internal effect (SNPs inside exon), splicing effect (SNPs outside exon and inside genes) and both (Z-score are added artificially, but the results are separated)
- a list of genes from a file

Output:
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. A gene can be twice.

3.7 SNP_on_genes_file.py

From the two files '_sed' and '_pvZ', write 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

3.8 Manhattan_plot.py

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

4 Utilities

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

4.1 SNP_exon_utils.py

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

4.2 SNP_on_genes_utils.py

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

5 Supplement

Some scripts for other analyses which would be interesting.

5.1 exon_gene_analyse.py

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

5.2 SNP_gap_analyse.py

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

Contents

May 31, 2021 by daniel.rovera@gmail.com - Institut Curie