KnockoutQT Documentation

02/07/2015 – KnockoutQT V3

Samuel Lessard & Guillaume Lettre

1. Introduction

The *KnockoutQT* script (knockoutQT.rb) aims at finding gene knockouts (KOs) and testing their association with a given quantitative phenotype. To find KOs, *KnockoutQT* considers homozygotes for LoF as well as compound heterozygotes. The program can integrate phase information in order to differentiate *trans* events (compound heterozygote) from *cis* events (two variants segregating on the same haplotype). *KnockoutQT* tests the association between specific gene KOs and a quantitative phenotype by testing if KO samples have a phenotype that is more extreme (in either direction) than the rest of the samples. For each gene, an association *P*-value is calculated by performing phenotype permutations.

The script is adapted for files in either PLINK binary (.bim, .bed, .fam) or VCF format (.vcf or .vcf.gz). The latter is recommended as the program can only read phase information from VCF files with phased genotypes.

A R-script (knockoutQT.R) is also provided with the ruby script. The ruby script deals with finding KO events, whereas the R-script computes the association *P*-values. The ruby script automatically runs the R-script.

The statistical analysis performed by the script is adapted for phenotypes with symmetric distributions. Hence, it may be necessary to transform the data prior to the analysis (e.g. inverse normal transformation). The script also does not deal with covariates – those should be regressed out of the phenotype before performing the analysis.

2. Dependencies

R, Ruby, and PLINK must be installed (any recent versions). The knockoutQT.rb and knockoutQT.R scripts must be in the same directory.

3. Main usage

ruby ./knockoutQT.rb --input [input_file] --annotation [annotation_file] --output [output_prefix] -m [phenotype_file]

Input_file: Files in PLINK binary format or a VCF file. The program will automatically detect if the file is a VCF from the extension. When using **PLINK files, do not include the extension** (e.g. *--input PLINK* will read PLINK.bed, PLINK.bim, and PLINK.fam files).

annotation_file: File that links each variant (typically loss-of-function variants) to their gene. Only variants that should be considered in the analysis must be included in this file (i.e. do not include synonymous SNPs if you don't want them analyzed as a loss-of-function variant). SNP IDs for VCF **must** be in the format: var_chr[CHR]_[POSITION].

Example 1:

SNP_ID	GENE_ID
rs1234	GENE1
rs1235	GENE2
rs1236	GENE3
…	

Example 2 (for VCF):

SNP_ID	GENE_ID
var_chr1_1234	GENE1
var_chr3_2222	GENE2
var_chr8_3333	GENE3
…	

output_prefix: Output basename. See below for outputted files.

phenotype_file: File that contains the phenotypic values for each sample. One or multiple phenotypes can be included. First 2 columns must be sample ID (NOTE: FID and IID should be identical).

Example:

FID	IID	PHEN1	PHEN2	…
123	123	2.2	2.1	…
124	124	2.1	3.1	…
125	125	4.8	2.8	…

4. Options:

1. Mandatory arguments

-a, --annotation FILE Annotation file associating marker ID to a gene ID

-i, --input FILE PLINK files basename or VCF file. Default is PLINK binary format (.bim, .bed .fam). Automatically detects VCF (.vcf or .vcf.gz). VCF is recommended. E.g. [-i PLINK] will detect PLINK.bim, PLINK.bed, and PLINK.fam. [-i VCF.vcf] will read VCF.vcf

-o, --output FILE Output file basename.

-m, --multipheno FILE File with phenotype(s) information. Phenotype file MUST have a header. Two first columns are individual IDs. All phenotypes in files will be tested.

2. File handling options:

-p, --phase Use phase information. Will throw an error if phase is not available. For VCF files only.

-G, --gene-col NUM Number of the column containing gene ID in annotation file. Default is 2.

-M, --marker-col NUM Number of the column containing marker ID in annotation file. For VCF files, marker ID MUST be in the var_chr<CHR>_<POS> format. Default is 1.

-s, --A-sep SEP Column separator in annotation file. Default is: "\t".

-A, --A-header Annotation file has a header. Default is false.

3. Analysis options:

-n, --nperm NUM Number of permutations; default = 10000.

-t, --multithread NUM Number of threads to use. Applicable only with multiple phenotypes. Default is 1.

-f, --max-maf NUM Maximum minor allele frequency. Default is 0.05 (5%).

-r, --remove_monomorphic Remove SNPs with AF=0. Use carefuly as very rare SNPs might be marked AF=0 in VCF files!!! Default is false.

-X, --chrX Include chrX non-pseudoautosomal regions in analysis - or any other haploid genotypes. For VCF files only. Default behavior is to skip SNP with only one strand (i.e. GT=0 is excluded; GT=0/0 is included)

```
Program options:

    -h, --help                  Display this screen.
```

<u>5. Output files</u>

Multiple files are created while running the script, most of which can be disregarded. Temporary files have a ".temp" extension. The main result files are described here.

1) .assoc file: outputs the association results, including the *P*-values. This file contains the <u>main results</u> for the analysis. This file contains multiple columns:

GENE	Gene ID
N_carriers	Number of homozygote or compound heterozygote KOs
MEAN	Phenotypic mean of KOs
SD	Standard deviation of KOs
MIN	Lowest phenotypic value of KOs
MAX	Highest phenotypic value of KOs
N_noncarriers	Number of individuals that are not detected as KOs (i.e. controls)
MEAN_noncarriers	Mean phenotype of controls
P	Association *P*-value
N_markers	Number of markers analyzed in gene
minMAF	Lowest minor allele frequency
maxMAF	Highest minor allele frequency
nHZ	Number of KOs with homozygous LoF
nCH	Number of KOs with compound heterozygote LoF

2) .out file: Details the SNPs analysed in each gene. For each gene, the information will be outputted in the following format:

```
> GENE_NAME
SNPS: snp_1 [freq=0.03], snp_2 [freq=0.05]
ID: ko_individual_1, ko_individual_2, ko_individual_3
Mean_phenotype: 2.55
Pvalue: 0.11
```

When using multiple phenotypes, these files will be outputted separately for each phenotype. An additional file is outputted (.all.assoc file). This file is a matrix of all phenotypes and genes, with their associated mean and *P*-value.

3) .all.assoc

```
GENE   mean_PHEN1   P_PHEN1     mean_PHEN2   P_PHEN2 ….
Gene1  6.3          0.0001      …
Gene2  2.3          0.3         …
…
```

4) .info: This file links the gene KOs to the phenotype. It is a temporary file used by the R-script to calculate P-values.

5) .knockouts.count: A file listing the number of homozygous (nHz) and compound heterozygous (nCH) KOs per individuals. The sum of these corresponds to the total number of KO per individual.

```
ID nHz nCH
ID_1 2 0
ID_2 0 1
ID_3 0 0
```

6. Examples

1) The user wants to run the knockoutQT.rb script with 50000 permutations on the plink.bam, plink.bim, and plink.fam files using an annotation file *annotation.txt* and phenotype file *phenotypes.txt*. The user wants to use 4 threads to run 4 phenotypes in parallel.

   - ➢ *ruby ./knockoutQT.rb --input plink --annotation annotation.txt --output output -m phenotypes.txt -n 50000 -t 4*

2) The user wants to run the script on phased genotypes of variants with MAF<0.01 using the file vcfFile.vcf.gz

   - ➢ *ruby ./knockoutQT.rb --input vcfFile.vcf.gz --annotation annotation.txt --output output -m phenotypes.txt -p -f 0.01*