

# Using RQT, an R/Bioconductor package for gene-level meta-analysis

Ilya Y. Zhbannikov

December 10, 2016

## 1 Overview

Despite the recent advances of modern GWAS methods, it still remains an important problem of addressing calculation an effect size and corresponding p-value for the whole gene rather than for single variant. We developed an R-package `rqt`, which offers gene-level GWAS meta-analysis. The package can be easily included into bioinformatics pipeline or used as stand-alone. Contact: [ilya.zhbannikov@duke.edu](mailto:ilya.zhbannikov@duke.edu) for questions of usage the `rqt` or any other issues.

Below we provide several examples that show GWAS meta-analysis on gene-level layer.

### 1.1 Methods in brief

The workflow of gene-level meta analysis consists of the following steps: (i) reducing the number of predictors, thereby alleviating correlation problem in variants (accounting for LD); (ii) then the regression model is fitted on the reduced dataset to obtain corresponding regression coefficient ("effect sizes"); (iii) these coefficients are then to be pooled into a total index representing a total gene-level effect size and corresponding statistics is calculated. P- and q- values are then calculated using this statistics from asymptotic approximation or permutation procedure; (iv) the final step is combining gene-level p-values calculated from each study with Fisher's combined probability method.

## 2 Installation

### 2.1 Most-recent version from GitHub

```
> require(devtools)
> devtools::install_github("izhbannikov/rqt")
```

## 3 Data description

### 3.1 Single dataset

In `rqt` requires the following datasets: (i) **phenotype** (a N by 1) matrix (i.e. a vector); and (ii) **genotype** - an object of class `SummarizedExperiment` containing one assay: (a N by M) matrix, where N - is the total number of individuals in the study and M is the total number of genetic variants. Optionally, `rqt` can accept covariates, in form of N by K matrix, where K is the total number of covariates used in the study. Phenotype can be dichotomous (0/1, where 1 indicates control and 0 case).

## 3.2 Meta-analysis

In meta-analysis, `rqt` requires a list of `M` (`M` - number of datasets used in meta-analysis) and optionally it accepts covariates in form described above.

## 4 Examples

### 4.1 Gene-level analysis on a single dataset

### 4.2 Dichotomous phenotype

```
> library(rqt)
> data <- data.matrix(read.table(system.file("extdata/test.bin1.dat",
+                                           package="rqt"), header=TRUE))
> pheno <- data[,1]
> geno <- data[, 2:dim(data)[2]]
> colnames(geno) <- paste(seq(1, dim(geno)[2]))
> geno.obj <- SummarizedExperiment(geno)
> obj <- rqtClass(phenotype=pheno, genotype=geno.obj)
> res <- rQTTest(obj, method="pca", out.type = "D")
> print(res)
```

Phenotype:

```
[1] 1 1 1 1 1 1
...
```

Genotype:

```
  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
[1,] 0 0 0 1 0 2 1 0 0 0 2 1 2 0 0 2 1 1 0 0 1 0 0 0 0 2 1 0
[2,] 1 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 2 1 2 1 0 1 0 0 0 2 1 1
[3,] 0 0 0 0 1 0 0 1 0 1 1 1 0 0 0 1 1 0 0 1 1 1 0 0 0 1 0 0
[4,] 0 0 1 0 1 0 0 1 1 0 0 0 1 0 0 0 1 0 0 2 0 1 0 0 0 1 0 2
[5,] 0 0 1 1 1 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 2 0
[6,] 0 0 1 1 1 0 0 1 0 1 1 0 1 0 0 2 0 0 1 0 1 1 0 0 0 2 0 0
 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
[1,] 1 0 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 1 1 0 0 0 1
[2,] 2 0 0 0 1 2 1 2 0 1 1 0 1 0 0 1 0 0 0 2 0 1 0 1 0
[3,] 0 0 0 0 0 1 1 1 2 2 0 0 0 1 0 1 0 2 1 1 1 0 0 0 1
[4,] 0 0 0 1 2 2 0 1 1 1 1 0 0 0 1 1 0 0 1 1 2 0 1 0 2
[5,] 0 0 0 0 1 1 0 0 1 1 0 0 2 2 0 0 0 1 2 1 0 0 0 0 1
[6,] 0 0 0 0 1 0 1 1 0 0 2 0 1 1 2 0 0 0 1 1 1 0 0 1 1
 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78
[1,] 1 1 1 1 0 0 0 0 1 1 0 1 1 1 0 1 1 2 1 1 1 0 0 2 1
[2,] 2 1 0 1 0 0 0 1 0 1 2 2 1 1 0 1 0 1 0 0 0 0 1 1 2
[3,] 2 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 2 1 0 1 1 0 0 0 0
[4,] 1 1 0 0 0 0 2 1 0 1 0 0 2 1 0 0 1 1 1 0 0 0 0 0 1
[5,] 1 1 1 0 0 0 2 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1
[6,] 0 1 1 0 0 0 0 0 0 0 1 2 1 1 0 0 0 0 0 1 1 1 1 1 1
 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
[1,] 1 1 0 1 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1
[2,] 1 1 2 0 0 0 1 2 0 2 1 0 0 2 1 0 0 0 0 1 1 2
[3,] 1 1 0 0 0 0 2 0 0 2 0 0 0 0 1 1 0 0 1 2 0 0
```

```
[4,] 0 1 2 0 1 0 1 2 0 0 0 0 0 0 1 0 0 0 1 0 0 0
[5,] 0 2 0 1 0 0 2 0 0 0 0 1 0 0 1 0 0 1 1 2 1 1
[6,] 2 1 0 0 1 0 0 0 0 2 0 1 0 0 0 0 0 1 1 0 1 1
...
```

Covariates:  
data frame with 0 columns and 0 rows

Results:

```
$Qstatistic
      Q1      Q2      Q3
1 0.9086994 0.9938212 0.9086994
```

```
$p.value
      p.Q1      p.Q2      p.Q3
1 0.3404597 0.9285409 0.5654533
```

### 4.3 Continuous phenotype

```
> library(rqt)
> data <- data.matrix(read.table(system.file("extdata/test.cont1.dat",
+                                           package="rqt"), header = TRUE))
> pheno <- data[,1]
> geno <- data[, 2:dim(data)[2]]
> colnames(geno) <- paste(seq(1, dim(geno)[2]))
> geno.obj <- SummarizedExperiment(geno)
> obj <- rqtClass(phenotype=pheno, genotype=geno.obj)
> res <- rqtTest(obj, method="pca", out.type = "C")
> print(res)
```

Phenotype:  
[1] 3.422452 2.457394 2.708564 4.589394 5.461723 4.438707  
...

Genotype:

```
      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
[1,] 0 0 0 1 0 2 1 0 0 0 2 1 2 0 0 2 1 1 0 0 1 0 0 0 0 2 1 0
[2,] 1 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 2 1 2 1 0 1 0 0 0 2 1 1
[3,] 0 0 0 0 1 0 0 1 0 1 1 1 0 0 0 1 1 0 0 1 1 1 0 0 0 1 0 0
[4,] 0 0 1 0 1 0 0 1 1 0 0 0 1 0 0 0 1 0 0 2 0 1 0 0 0 1 0 2
[5,] 0 0 1 1 1 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 2 0
[6,] 0 0 1 1 1 0 0 1 0 1 1 0 1 0 0 2 0 0 1 0 1 1 0 0 0 2 0 0
      29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
[1,] 1 0 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 1 1 0 0 0 1
[2,] 2 0 0 0 1 2 1 2 0 1 1 0 1 0 0 1 0 0 0 2 0 1 0 1 0
[3,] 0 0 0 0 0 1 1 1 2 2 0 0 0 1 0 1 0 2 1 1 1 0 0 0 1
[4,] 0 0 0 1 2 2 0 1 1 1 1 0 0 0 1 1 0 0 1 1 2 0 1 0 2
[5,] 0 0 0 0 1 1 0 0 1 1 0 0 2 2 0 0 0 1 2 1 0 0 0 0 1
[6,] 0 0 0 0 1 0 1 1 0 0 2 0 1 1 2 0 0 0 1 1 1 0 0 1 1
      54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78
```

```

[1,] 1 1 1 1 0 0 0 0 1 1 0 1 1 1 0 1 1 2 1 1 1 0 0 2 1
[2,] 2 1 0 1 0 0 0 1 0 1 2 2 1 1 0 1 0 1 0 0 0 0 1 1 2
[3,] 2 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 2 1 0 1 1 0 0 0 0
[4,] 1 1 0 0 0 0 2 1 0 1 0 0 2 1 0 0 1 1 1 0 0 0 0 0 1
[5,] 1 1 1 0 0 0 2 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1
[6,] 0 1 1 0 0 0 0 0 0 0 1 2 1 1 0 0 0 0 0 1 1 1 1 1 1
      79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
[1,] 1 1 0 1 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1
[2,] 1 1 2 0 0 0 1 2 0 2 1 0 0 2 1 0 0 0 0 1 1 2
[3,] 1 1 0 0 0 0 2 0 0 2 0 0 0 0 1 1 0 0 1 2 0 0
[4,] 0 1 2 0 1 0 1 2 0 0 0 0 0 0 1 0 0 0 1 0 0 0
[5,] 0 2 0 1 0 0 2 0 0 0 0 1 0 0 1 0 0 1 1 2 1 1
[6,] 2 1 0 0 1 0 0 0 0 2 0 1 0 0 0 0 0 1 1 0 1 1
...

```

Covariates:  
data frame with 0 columns and 0 rows

Results:

```

$Qstatistic
      Q1      Q2      Q3
1 0.2846585 2.389537 2.219594

$p.value
      p.Q1      p.Q2      p.Q3
1 0.593664 0.7022459 0.2561996

```

#### 4.4 Preprocessing with Partial Least Square regression (PLS)

This method is used for continous outcome, i.e. `out.type = "C"`.

```

> library(rqt)
> data <- data.matrix(read.table(system.file("extdata/test.cont1.dat",
+                                           package="rqt"), header = TRUE))
> pheno <- data[,1]
> geno <- data[, 2:dim(data)[2]]
> colnames(geno) <- paste(seq(1, dim(geno)[2]))
> geno.obj <- SummarizedExperiment(geno)
> obj <- rqtClass(pheno=pheno, genotype=geno.obj)
> res <- rqtTest(obj, method="pls", out.type = "C")
> print(res)

```

Phenotype:  
[1] 3.422452 2.457394 2.708564 4.589394 5.461723 4.438707  
...

Genotype:

```

      1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
[1,] 0 0 0 1 0 2 1 0 0 0 2 1 2 0 0 2 1 1 0 0 1 0 0 0 0 2 1 0
[2,] 1 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 2 1 2 1 0 1 0 0 0 2 1 1

```

```

[3,] 0 0 0 0 1 0 0 1 0 1 1 1 0 0 0 1 1 0 0 1 1 1 0 0 0 1 0 0
[4,] 0 0 1 0 1 0 0 1 1 0 0 0 1 0 0 0 1 0 0 2 0 1 0 0 0 1 0 2
[5,] 0 0 1 1 1 1 1 1 0 1 0 1 0 0 0 0 0 0 0 1 0 0 1 0 1 2 0
[6,] 0 0 1 1 1 0 0 1 0 1 1 0 1 0 0 2 0 0 1 0 1 1 0 0 0 2 0 0
      29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
[1,] 1 0 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 1 1 0 0 0 0 1
[2,] 2 0 0 0 1 2 1 2 0 1 1 0 1 0 0 1 0 0 0 2 0 1 0 1 0
[3,] 0 0 0 0 0 1 1 1 2 2 0 0 0 1 0 1 0 2 1 1 1 0 0 0 1
[4,] 0 0 0 1 2 2 0 1 1 1 1 0 0 0 1 1 0 0 1 1 2 0 1 0 2
[5,] 0 0 0 0 1 1 0 0 1 1 0 0 2 2 0 0 0 1 2 1 0 0 0 0 1
[6,] 0 0 0 0 1 0 1 1 0 0 2 0 1 1 2 0 0 0 1 1 1 0 0 1 1
      54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78
[1,] 1 1 1 1 0 0 0 0 1 1 0 1 1 1 0 1 1 2 1 1 1 0 0 2 1
[2,] 2 1 0 1 0 0 0 1 0 1 2 2 1 1 0 1 0 1 0 0 0 0 1 1 2
[3,] 2 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 2 1 0 1 1 0 0 0 0
[4,] 1 1 0 0 0 0 2 1 0 1 0 0 2 1 0 0 1 1 1 0 0 0 0 0 1
[5,] 1 1 1 0 0 0 2 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1
[6,] 0 1 1 0 0 0 0 0 0 0 1 2 1 1 0 0 0 0 0 1 1 1 1 1 1
      79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
[1,] 1 1 0 1 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1
[2,] 1 1 2 0 0 0 1 2 0 2 1 0 0 2 1 0 0 0 0 1 1 2
[3,] 1 1 0 0 0 0 2 0 0 2 0 0 0 0 1 1 0 0 1 2 0 0
[4,] 0 1 2 0 1 0 1 2 0 0 0 0 0 0 1 0 0 0 1 0 0 0
[5,] 0 2 0 1 0 0 2 0 0 0 0 1 0 0 1 0 0 1 1 2 1 1
[6,] 2 1 0 0 1 0 0 0 0 2 0 1 0 0 0 0 0 1 1 0 1 1
...

```

Covariates:

data frame with 0 columns and 0 rows

Results:

\$Qstatistic

	Q1	Q2	Q3
1	0.1910558	71.34471	16.59228

\$p.value

	p.Q1	p.Q2	p.Q3
1	0.6620394	1.293491e-05	0.00011429

## 4.5 Preprocessing with Partial Least Square Discriminant Analysis (PLS-DA)

This method of data preprocessing is used for dichotomous outcome.

```

> library(rqt)
> data <- data.matrix(read.table(system.file("extdata/test.bin1.dat",
+                                           package="rqt"), header=TRUE))
> pheno <- data[,1]
> geno <- data[, 2:dim(data)[2]]
> colnames(geno) <- paste(seq(1, dim(geno)[2]))
> geno.obj <- SummarizedExperiment(geno)

```

```

> obj <- rqtClass(phenotype=pheno, genotype=geno.obj)
> res <- rQTest(obj, method="pls", out.type = "D", scale = TRUE)

PLS-DA
200 samples x 100 variables and 1 response
standard scaling of predictors and response(s)
      R2X(cum) R2Y(cum) Q2(cum) RMSEE pre ort pR2Y pQ2
Total  0.0187   0.301  -0.292   0.2   1   0 0.95 0.8
<simpleError in model$scoreMN: $ operator not defined for this S4 class>

```

```
> print(res)
```

Phenotype:

```
[1] 1 1 1 1 1 1
```

...

Genotype:

```

      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
[1,] 0 0 0 1 0 2 1 0 0 0 2 1 2 0 0 2 1 1 0 0 1 0 0 0 0 2 1 0
[2,] 1 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 2 1 2 1 0 1 0 0 0 2 1 1
[3,] 0 0 0 0 1 0 0 1 0 1 1 1 0 0 0 1 1 0 0 1 1 1 0 0 0 1 0 0
[4,] 0 0 1 0 1 0 0 1 1 0 0 0 1 0 0 0 1 0 0 2 0 1 0 0 0 1 0 2
[5,] 0 0 1 1 1 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 2 0
[6,] 0 0 1 1 1 0 0 1 0 1 1 0 1 0 0 2 0 0 1 0 1 1 0 0 0 2 0 0
      29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
[1,] 1 0 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 1 1 0 0 0 1
[2,] 2 0 0 0 1 2 1 2 0 1 1 0 1 0 0 1 0 0 0 2 0 1 0 1 0
[3,] 0 0 0 0 0 1 1 1 2 2 0 0 0 1 0 1 0 2 1 1 1 0 0 0 1
[4,] 0 0 0 1 2 2 0 1 1 1 1 0 0 0 1 1 0 0 1 1 2 0 1 0 2
[5,] 0 0 0 0 1 1 0 0 1 1 0 0 2 2 0 0 0 1 2 1 0 0 0 0 1
[6,] 0 0 0 0 1 0 1 1 0 0 2 0 1 1 2 0 0 0 1 1 1 0 0 1 1
      54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78
[1,] 1 1 1 1 0 0 0 0 1 1 0 1 1 1 0 1 1 2 1 1 1 0 0 2 1
[2,] 2 1 0 1 0 0 0 1 0 1 2 2 1 1 0 1 0 1 0 0 0 0 1 1 2
[3,] 2 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 2 1 0 1 1 0 0 0 0
[4,] 1 1 0 0 0 0 2 1 0 1 0 0 2 1 0 0 1 1 1 0 0 0 0 0 1
[5,] 1 1 1 0 0 0 2 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1
[6,] 0 1 1 0 0 0 0 0 0 0 0 1 2 1 1 0 0 0 0 0 1 1 1 1 1
      79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
[1,] 1 1 0 1 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1
[2,] 1 1 2 0 0 0 1 2 0 2 1 0 0 2 1 0 0 0 0 1 1 2
[3,] 1 1 0 0 0 0 2 0 0 2 0 0 0 0 1 1 0 0 1 2 0 0
[4,] 0 1 2 0 1 0 1 2 0 0 0 0 0 0 1 0 0 0 1 0 0 0
[5,] 0 2 0 1 0 0 2 0 0 0 0 1 0 0 1 0 0 1 1 2 1 1
[6,] 2 1 0 0 1 0 0 0 0 2 0 1 0 0 0 0 0 1 1 0 1 1

```

...

Covariates:

data frame with 0 columns and 0 rows

Results:

```
$Qstatistic
  Q1 Q2 Q3
1 NA NA NA

$p.value
  p.Q1 p.Q2 p.Q3
1   NA   NA   NA
```

## 4.6 Using additional covariates

Quite often, researchers want to supply not only genetic data but also specific covariates, representing some physiological parameters or environment (for example, to evaluate hypotheses of gene-environment interactions). In such cases, the package `rqt` can accept additional covariates, in form of N by K matrix, as provided below:

```
> library(rqt)
> data <- data.matrix(read.table(system.file("extdata/test.bin1.dat",
+                                           package="rqt"), header = TRUE))
> pheno <- data[,1]
> geno <- data[, 2:dim(data)[2]]
> colnames(geno) <- paste(seq(1, dim(geno)[2]))
> geno.obj <- SummarizedExperiment(geno)
> covars <- read.table(system.file("extdata/test.cova1.dat", package="rqt"),
+                        header=TRUE)
> obj <- rqtClass(phenotype=pheno, genotype=geno.obj, covariates = covars)
> res <- rQTest(obj, method="pca", out.type = "D")
> print(res)
```

```
Phenotype:
[1] 1 1 1 1 1 1
...
```

```
Genotype:
  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
[1,] 0 0 0 1 0 2 1 0 0 0 2 1 2 0 0 2 1 1 0 0 1 0 0 0 0 2 1 0
[2,] 1 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 2 1 2 1 0 1 0 0 0 2 1 1
[3,] 0 0 0 0 1 0 0 1 0 1 1 1 0 0 0 1 1 0 0 1 1 1 0 0 0 1 0 0
[4,] 0 0 1 0 1 0 0 1 1 0 0 0 1 0 0 0 1 0 0 2 0 1 0 0 0 1 0 2
[5,] 0 0 1 1 1 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 2 0
[6,] 0 0 1 1 1 0 0 1 0 1 1 0 1 0 0 2 0 0 1 0 1 1 0 0 0 2 0 0
 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
[1,] 1 0 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 1 1 0 0 0 1
[2,] 2 0 0 0 1 2 1 2 0 1 1 0 1 0 0 1 0 0 0 2 0 1 0 1 0
[3,] 0 0 0 0 0 1 1 1 2 2 0 0 0 1 0 1 0 2 1 1 1 0 0 0 1
[4,] 0 0 0 1 2 2 0 1 1 1 1 0 0 0 1 1 0 0 1 1 2 0 1 0 2
[5,] 0 0 0 0 1 1 0 0 1 1 0 0 2 2 0 0 0 1 2 1 0 0 0 0 1
[6,] 0 0 0 0 1 0 1 1 0 0 2 0 1 1 2 0 0 0 1 1 1 0 0 1 1
 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78
[1,] 1 1 1 1 0 0 0 0 1 1 0 1 1 1 0 1 1 2 1 1 1 0 0 2 1
[2,] 2 1 0 1 0 0 0 1 0 1 2 2 1 1 0 1 0 1 0 0 0 0 1 1 2
[3,] 2 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 2 1 0 1 1 0 0 0 0
```

```

[4,] 1 1 0 0 0 0 2 1 0 1 0 0 2 1 0 0 1 1 1 0 0 0 0 0 1
[5,] 1 1 1 0 0 0 2 1 1 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 1
[6,] 0 1 1 0 0 0 0 0 0 0 0 1 2 1 1 0 0 0 0 0 1 1 1 1 1
      79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
[1,] 1 1 0 1 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0 1 1
[2,] 1 1 2 0 0 0 1 2 0 2 1 0 0 2 1 0 0 0 0 1 1 2
[3,] 1 1 0 0 0 0 2 0 0 2 0 0 0 0 1 1 0 0 1 2 0 0
[4,] 0 1 2 0 1 0 1 2 0 0 0 0 0 0 1 0 0 0 1 0 0 0
[5,] 0 2 0 1 0 0 2 0 0 0 0 1 0 0 1 0 0 1 1 2 1 1
[6,] 2 1 0 0 1 0 0 0 0 2 0 1 0 0 0 0 0 1 1 0 1 1
...

```

Covariates:

```

      COV1
1 -0.612463927
2 -0.464158885
3  0.006153597
4 -0.732109468
5 -0.223530136
6 -0.744903822

```

Results:

```

$Qstatistic
      Q1      Q2      Q3
1 2.012625 3.761859 2.012625

$p.value
      p.Q1      p.Q2      p.Q3
1 0.1559952 0.8258796 0.2895166

```

## 5 Meta-analysis

```

> library(rqt)
> data1 <- data.matrix(read.table(system.file("extdata/phengen2.dat",
+                                           package="rqt"), skip=1))
> pheno <- data1[,1]
> geno <- data1[, 2:dim(data1)[2]]
> colnames(geno) <- paste(seq(1, dim(geno)[2]))
> geno.obj <- SummarizedExperiment(geno)
> obj1 <- rqtClass(pheno=pheno, genotype=geno.obj)
> data2 <- data.matrix(read.table(system.file("extdata/phengen3.dat",
+                                           package="rqt"), skip=1))
> pheno <- data2[,1]
> geno <- data2[, 2:dim(data2)[2]]
> colnames(geno) <- paste(seq(1, dim(geno)[2]))
> geno.obj <- SummarizedExperiment(geno)
> obj2 <- rqtClass(pheno=pheno, genotype=geno.obj)
> data3 <- data.matrix(read.table(system.file("extdata/phengen.dat",
+                                           package="rqt"), skip=1))

```



```

> pheno <- data3[,1]
> geno <- data3[, 2:dim(data3)[2]]
> colnames(geno) <- paste(seq(1, dim(geno)[2]))
> geno.obj <- SummarizedExperiment(geno)
> obj3 <- rqtClass(phenotype=pheno, genotype=geno.obj)
> res.meta <- rQTestMeta(list(obj1, obj2, obj3))
> print(res.meta)

```

```

$final.pvalue
[1] 0.01995627

```

```

$pvalueList
[1] 0.006016377 0.367634896 0.245240026

```

```

$df
[1] 6

```

```

$chi.comb
[1] 15.03891

```

## 6 Session information

```

> sessionInfo()

```

```

R version 3.3.2 (2016-10-31)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: macOS Sierra 10.12.1

```

```

locale:
[1] C

```

```

attached base packages:
[1] parallel stats4 stats graphics grDevices utils datasets
[8] methods base

```

```

other attached packages:
[1] rqt_0.99.3 SummarizedExperiment_1.4.0
[3] Biobase_2.34.0 GenomicRanges_1.26.1
[5] GenomeInfoDb_1.10.1 IRanges_2.8.1
[7] S4Vectors_0.12.1 BiocGenerics_0.20.0
[9] ropls_1.6.2 plyr_1.8.4
[11] pls_2.5-0 glmnet_2.0-5
[13] foreach_1.4.3 Matrix_1.2-7.1

```

```

loaded via a namespace (and not attached):
[1] Rcpp_0.12.8 lattice_0.20-34 codetools_0.2-15 grid_3.3.2
[5] zlibbioc_1.20.0 CCP_1.1 XVector_0.14.0 iterators_1.0.8
[9] tools_3.3.2 CompQuadForm_1.4.2

```

## 7 References

### References

- [1] Wu, M. C., Lee, S., Cai, T., Li, Y., Boehnke, M., and Lin, X. (2011) Rare Variant Association Testing for Sequencing Data Using the Sequence Kernel Association Test (SKAT). *The American Journal of Human Genetics*, 89, 82-93.
- [2] Li, B. and Leal, S.M. (2008) Methods for Detecting Associations with Rare Variants for Common Diseases: Application to Analysis of Sequence Data. *The American Journal of Human Genetics*, 83, 311-321.
- [3] Liu D., Leal S. (2010) A Novel Adaptive Method for the Analysis of Next-Generation Sequencing Data to Detect Complex Trait Associations with Rare Variants Due to Gene Main Effects and Interactions, *PLoS Genet.*, 6(10).
- [4] Madsen, B.E, Browning, S.R. (2009) A Groupwise Association Test for Rare Mutations Using a Weighted Sum Statistic, *PLoS Genet.*, 5(2).
- [5] Lee, J., Kim, Y.J., Lee, J., T2D-Genes Consortium, Kim, B-J., Lee, S., Park T. (2016) Gene-set association tests for next-generation sequencing data, *Bioinformatics*, 32(17).
- [6] Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.*, Vol. 58, No. 1, pages 267-288.