PQLseq User Manual
Version 1.10

Shiquan Sun, Jiaqiang Zhu, and Xiang Zhou
Department of Biostatistics, University of Michigan, Ann Arbor
shiquans@umich.edu and xzhousph@umich.edu

March 20, 2018

Contents

1 Introduction

1.1 What is PQLseq

PQLseq is an R package implementing the generalized linear mixed model for estimating molecular trait heritability and identifying differential genes/sites in RAN sequencing (RNAseq) or bisulfite sequencing (BSeq) studies. The models are to account for population stratification or structure and directly work with the raw read counts. PQLseq takes advantage of the parallel computing environment commonly available in modern computers by leveraging the parallel packages in R. PQLseq can easily handle hundreds of individuals and millions of units in large-scale genomic sequencing studies that are becoming increasingly common.

1.2 The Models

1.2.1 PMM

To detect differentially expressed genes, we examine one gene at a time with the following Poisson mixed model (PMM):

$$y_i \sim \text{Poi}(N_i\lambda_i), i = 1, 2, \cdots, n, \tag{1}$$

where N_i is the read depth of ith individual; y_i is the read count of the particular gene; λ_i is an unknown rate parameter.

1.2.2 BMM

To detect differentially methylated sites, we examine one site at a time with the following binomial mixed model (BMM):

$$y_i \sim \text{Bin}(r_i, \pi_i), i = 1, 2, \cdots, n, \tag{2}$$

where r_i is the read depth of ith individual; y_i is the read count of the particular site; π_i is an unknown parameter that represents the true proportion of methylated reads for the individual at the site, and we denote $\lambda_i = \frac{\pi_i}{1-\pi_i}$.

For the unknown parameter λ_i in both two models above, we use a log link to model it as a linear function of parameters:

$$\log(\lambda_i) = \mathbf{w}_i^T\boldsymbol{\alpha} + x_i\beta + g_i + e_i, \tag{3}$$

$$\mathbf{g} = c(g_1, \cdots, g_n)^T \sim \text{MVN}(0, \sigma^2h^2\mathbf{K}_{n\times n}), \tag{4}$$

$$\mathbf{e} = c(e_1, \cdots, e_n)^T \sim \text{MVN}(0, \sigma^2(1-h^2)\mathbf{I}_{n\times n}), \tag{5}$$

where $\mathbf{w}_i$ is a c-vector of covariates including an intercept and $\boldsymbol{\alpha}$ is a c-vector of corresponding coefficients; x_i is the predictor of interest and β is its coefficient; $\mathbf{g}$ is an n-vector of genetic random effects that model correlation due to population structure or individual relatedness; $\mathbf{e}$ is an n-vector of environmental residual errors that model independent variation; $\mathbf{K}$ is a known n by n relatedness matrix that can be calculated based on a pedigree or genotype data and that has been standardized to ensure $tr(\mathbf{K})/n = 1$ (this ensures that h^2 lies between 0 and 1, and can be interpreted as heritability, see [1]); $\mathbf{I}$ is an n by n identity matrix; σ^2h^2 is the genetic variance component; $\sigma^2(1-h^2)$ is the environmental variance component; h^2 represents the heritability of the log transformed rate (i.e. $\log(\lambda)$); and MVN denotes the multivariate normal distribution.

1.3 Hypothesis Testing

In both the BMM and PMM, we are interested in testing the null hypothersis $H_0 : \beta = 0$ and estimating heritability h^2. Unlike its linear counterpart, estimating β and h^2 from the binomial mixed model is notoriously difficult, as the joint likelihood consists of an n-dimensional integral that cannot be solved analytically. Previous versions of MACAU apply MCMC to draw posterior samples and rely on the asymptotic normality of both the likelihood and the posterior distributions to obtain the approximate maximum likelihood estimate β_j and its standard error $se(\beta_j)$. With β_j and $se(\beta_j)$, we can construct approximate Wald test statistics and p-values for hypothesis testing.

1.4 How to Cite PQLseq

Shiquan Sun, Jiaqiang Zhu, Sahar Mozaffari, Carole Ober, Mengjie Chen, and Xiang Zhou (2018). Heritability Estimation and Differential Analysis with Generalized Linear Mixed Models in Genomic Sequencing Studies. BioR$\mathcal{X}$iv.

Shiquan Sun, Michelle Hood, Laura Scott, Qinke Peng, Sayan Mukherjee, Jenny Tung, and Xiang Zhou (2017). Differential Expression Analysis for RNAseq using Poisson Mixed Models. *Nucleic Acids Research*. 45(11): e106.

Amanda J. Lea, Jenny Tung and Xiang Zhou (2015). A flexible, efficient binomial mixed model for identifying differential DNA methylation in bisulfite sequencing data. *PLoS Genetics*. 11: e1005650

2 Installation

The R package of PQLseq is freely available at `http://www.xzlab.org/software.html`. It can be installed as a regular R package on a UNIX/Linux operating system. Currently, PQLseq do not support Mac or Windows operating system.

2.1 Packages Dependence

PQLseq dependents on several existing packages which we list as follows:

- R package `Rcpp`: `https://cran.r-project.org/web/packages/Rcpp/index.html`
- R package `RcppArmadillo`: `https://cran.r-project.org/web/packages/RcppArmadillo/index.html`
- R package `foreach`: `https://cran.r-project.org/web/packages/foreach/index.html`
- R package `parallel`: `https://cran.r-project.org/web/packages/doParallel/index.html`

Note: If you install `RcppArmadillo` package, the `Rcpp` package will be installed automatically. Whatever you use parallel setting, the package `foreach` must be installed before you runing PQLseq

2.2 Installing PQLseq

Before installing PQLseq package, please make sure your system has installed the packages mentioned above Section (Section 2.1).

To install PQLseq into ”your directory”, in an R session you can use

```
> install.packages("PQLseq_0.1.0.tar.gz", lib = "your_directory")
```

3 Input Files

3.0.1 PMM

When you run the PMM, PQLseq requires three input files containing gene expression read counts, relatedness matrix, and predictor variable of interest.

3.0.2 BMM

When you run the BMM, PQLseq requires four input files containing methylated read counts, total read counts, relatedness matrix, and predictor variable of interest.

3.1 Count File

This file contains a table of read counts with a header. The first column lists gene/site IDs while the first row lists individual IDs. An example file with two genes and four individuals is as follows:

```
gene   idv1   idv2   idv3   idv4
gene1   2      4      3      8
gene2   3      0      15     9
```

To read it into R as a data frame, you can use

```
> counts <- read.table("counts.txt", header = T, row.names=1)
```

3.2 Total Count File

3.2.1 PMM

This file has two rows containing column names and row names. The first row lists individual IDs, the same as count file. The second row is the total number of read counts. It will be calculated automatically if users do not provided. An example predictor file with four individuals as follows:

```
num   idv1     idv2    idv3     idv4
total 12540    13654   19564    19846
```

3.2.2 BMM

This file contains a table of read counts with a header. Both read count file and total read count file are the same format. But this file must be provided by users. An example predictor file with four individuals and two sites as follows:

```
num   idv1   idv2   idv3   idv4
site1   3      5      5      10
site2   5      1      16      9
```

To read it into R as a data frame, you can use

```
> totalcounts <- read.table("totalcounts.txt", header = T, row.names=1)
```

3.3 Predictor File

This file contains the predictor variable of interest. The value can be binary phenotype (disease) or quantitative phenotype (trait). Each line is a number indicating the phenotype value for each individual in turn, in the same order as in the count files. Notice that only numeric values are allowed and characters will not be recognized by the software. Missing phenotype information is denoted as NA. The number of rows should be equal to the number of individuals in the read count file. An example predictor file with four individuals as follows:

```
1.2
NA
2.7
-0.2
```

To read it into R as a data frame, you can use

```
> pheno <- read.table("pheno.txt", header = F)
```

Missing values in the data frame should be recognizable by R as NA. For example, if you use `.` to denote missing values in the text file, you can use

```
> pheno <- read.table("pheno.txt", header = F, na.strings = ".")
```

3.4 Relatedness Matrix File

PQLseq requires a relatedness matrix file. It contains a $n \times n$ matrix, where each row and each column corresponds to individuals in the same order as in the count file or in the predictor file, and ith row and jth column is a number indicating the relatedness value between ith and j th individuals. If you only have genotypes, you can use GEMMA [1, 2, 3] to compute the relatedness matrix. For details, please refer to the GEMMA manual available at `www.xzlab.org/software.html`. An example relatedness matrix file with three individuals is as follows:

```
 0.3345   -0.0227   0.0103
-0.0227    0.3032  -0.0253
 0.0103   -0.0253   0.3531
```

To read it into R as a data frame, you can use

```
> kinship <- read.table("kinship.txt", header = F)
```

3.5 Covariates File (optional)

One can provide an optional covariates file for fitting PMM or BMM. The covariates file is similar to the above predictor file. An example covariates file with four individuals and two covariates (the first column is the intercept) is as follows:

```
1   1   -1.5
1   2    0.3
1   2    0.6
1   1   -0.8
```

If a column of 1s is not provided, then the software will automatically add one at the end of the covariate matrix.

To read it into R as a data frame, you can use

```
> covariates <- read.table("covariates.txt", header = F)
```

4 Running PQLseq

If PQLseq has been successfully installed into ”your directory”, you can load it in an R session using

```
> library(PQLseq, lib.loc = "your directory")
```

4.1 Fitting PMM

Here we provide a simple example of fitting PMM using PQLseq. Assume you have the predictor of interest, saved in a plain text file `pheno.txt`, the relatedness matrix and saved it in a text file `kinship.txt`, the read counts, saved in `counts.txt`. In this example we fit the PMM using

```
> fit = pqlseq(RawCountDataSet=counts, Phenotypes=pheno, RelatednessMatrix=kinship,
fit.model="PMM")
```

Notice: to make sure the order of individuals in the `kinship.txt`, the order of individuals in the read count `counts.txt` matches the order of individuals in the phenotype file `pheno.txt`.

4.2 Fitting BMM

Here we provide a simple example of fitting BMM using PQLseq. Assume you have the predictor of interest, saved in a plain text file `pheno.txt`, the relatedness matrix and saved it in a text file `kinship.txt`, the read count, saved in `counts.txt`, and total read count, saved in `totalcounts.txt`. In this example we fit the BMM using

```
> fit = pqlseq(RawCountDataSet=counts, Phenotypes=pheno, RelatednessMatrix=kinship,
LibSize=totalcounts, fit.model="BMM")
```

Notice: to make sure the order of individuals in the `kinship.txt`, the order of individuals in the read counts and total counts, matches the order of individuals in the phenotype file pheno.txt.

4.3 Output

There will be two output files, both inside an output folder in the current directory. An example assoc file with a few sites is shown below:

```
geneID   numIDV  beta  se_beta    pvalue    h2    sigma2 converged
gene1    205      0.018   0.021    0.407   0.036   0.022       TRUE
gene2    203    -0.121   0.321    0.016   0.175   0.279       TRUE
```

The columns are: gene/site ID, number of individuals analyzed at the given gene/site, β, $se(\beta)$, p-value, h^2, σ^2, over-dispersion (in our case we set its value as 1), whether the algorithm is converged or not.

5 Advanced options

5.1 Parallel parameter

By default we set the number of cores to 1:

```
numCore = 1
```

5.2 Quality assessment parameter

By default at least two individuals that read count are the larger than 5, otherwise we filter them out.

```
filtering=TRUE
```

5.3 Model fitting parameters

By default we set the maximum number of iteration to 500 and tolerance to declare convergence to 1e-5:

```
fit.maxiter = 500, fit.tol = 1e-5
```

References

[1] Xiang Zhou, Peter Carbonetto and Matthew Stephens (2013). Polygenic modeling with Bayesian sparse linear mixed models. *PLoS Genetics*. 9(2): e1003264.

[2] Xiang Zhou and Matthew Stephens (2012). Genome-wide efficient mixed-model analysis for association studies. *Nature Genetics*. 44: 821-824.

[3] Xiang Zhou and Matthew Stephens (2014). Efficient multivariate linear mixed model algorithms for genome-wide association studies. *Nature Methods*. 11(4): 407¨C409.

[4] Amanda J. Lea, Jenny Tung and Xiang Zhou (2015). A flexible, effcient binomial mixed model for identifying differential DNA methylation in bisulfite sequencing data. *PLoS Genetics*. 11: e1005650.