Simulation and power analysis for bisulfite sequencing data sets

I. Introduction and user instructions

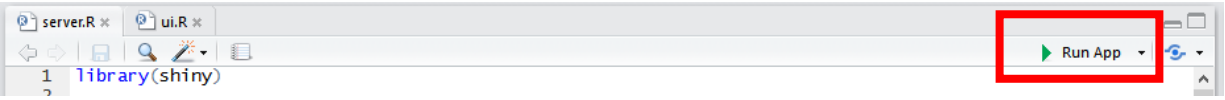
This application provides a platform for users to simulate and analyze bisulfite sequencing data to assist in study design and power analysis. It is run from Rstudio and requires the R packages *Shiny, MASS*, *matrixcalc*, *Matrix, qvalue*, *impute, gap*, and *EMMREML*. The *qvalue* and *impute* packages are available through Bioconductor, the other packages are all available through the CRAN repository.

If you use this app in a publication, please cite:
Lea, A.J., Vilgalys, T.P., Durst, P.A.P., and Tung, J. Maximizing ecological and evolutionary insight from bisulfite sequencing data sets. *bioRxiv*. dx.doi.org/10.1101/091488

A. Starting the R Shiny app

Download the R shiny app components (the server.R, ui.R, and www folder) to a folder that will serve as the working directory.

Open either the server.R or ui.R program in Rstudio and press “Run App”. Alternately, the command *shiny::runApp(*'[working directory name]'*)* will also start the application. Rstudio will open the correct shiny application even if only the server or ui file is open.



When the app opens, you will see a set of preset parameters. Select the type of model, sample size, and type of predictor variable that you would like to run, then press submit to run the simulation. Then, adjust the options and click submit to run a new simulation (see below).

B. Simulations

As the application starts, the user will have the chance to select the type of model and adjust parameters before running a simulation. If the ‘type of predictor’ or ‘type of model’ is changed, new drop-down menus/slide bars for providing option-specific information will become available before a new simulation is run (e.g., when a linear mixed model is selected, the heritability and covariance matrix will become available as input parameters).

Each simulation generates simulated count data (the total number of reads and the number of methylated reads) for 5000 sites based on a predefined predictor variable. Users may adjust the number of samples, the type of predictor variable (binary, continuous and normally distributed, or user defined; for user defined options, see ‘*User-defined variables’* below), the type of statistical model used for data analysis (unpaired t-test, linear model, linear mixed model, or beta-binomial model), and properties of the data including the range of mean coverage values per site, the proportion of variance explained by the predictor variable for true positive sites, the false discovery rate threshold, and the proportion of the data set simulated as true positives. The expected proportion of variance explained by the predictor variable for the remainder of the data set (the true negatives) is set to 0 (note, because the data are simulated, the observed PVE for true negative sites will be slightly greater than 0).

C. User-defined variables

Some users may wish to simulate results for a specific predictor variable of interest. To input user-defined values, the application requires the user to upload a file containing a single

column of predictor values. This column should have row number equal to the number of samples *n*, with no header or other information. An example of a binary predictor value for 30 individuals is packaged with the app in the file 'example_pred30.txt'.

Similarly, when running a linear mixed model simulation, it is required that the user provide a kinship or relatedness matrix (K matrix). This file should not contain column or row names, just an *n*-by-*n* matrix. An example matrix with a sample size of 30 and no covariance between samples is packaged with the app in the file 'example_covar_n30.txt'. Note that if the user inputs both a relatedness matrix and a predictor variable, individuals must be ordered identically in both files.

D. Output

In the user interface, the basic output is a Q-Q plot displaying the distribution of $-\log_{10}$(p-values) compared to those generated from a uniform distribution. The accompanying legend lists the proportion of true positives detected (at the user-specified FDR, assuming a uniform null) and the percent of true negatives detected as false positives. Because users may also wish to use the simulated data for more sophisticated downstream applications or to test other software, the simulated data and predictor files are saved at the end of each simulation. 5000-by-n matrices of count data are saved to the user's working directory under the file names 'simulated_counts.txt' and 'simulated_mcounts.txt'. The counts table refers to the total number of reads simulated per site, the mcounts table refers to the number of reads simulated as methylated at each site. An additional n-by-1 vector of predictor variables ('simulated_predictor.txt') is saved if the predictor was simulated. Finally, the results of the statistical test are saved as a 5000-by-2 table of p- and q-values.

II. Count data simulations

In each simulation, the number of total read counts per sample i and per site j, noted here as c_{ij}, is generated for 5000 sites based on random draws from a negative binomial distribution as follows. First, the mean coverage for each site is drawn from the uniform distribution between the user-defined minimum and maximum coverage values. Second, counts per individual are simulated using a negative binomial based on the coverage for that site (from step 1) and a shape parameter based on a regression line fit from a previously generated mammalian RRBS dataset (Lea *et al.* 2015).

The methylation level, π_{ij},, for each sample at each site is simulated as a linear function of the predictor variable and the effect size, β_j, with noise, $\varepsilon_{ij} \sim N(0, \sigma^2)$. The amount of random variance, σ^2, is fixed at 3 for these simulations, but can be changed by modifying the sigma2 variable in the server.R file (line 16). π_{ij} is transformed to the proportion of reads methylated (bounded [0,1]), p_{ij}, using an inverse-logit link function.

The number of methylated reads, m_{ij}, for each individual at each site is then simulated by drawing from a binomial distribution parameterized by the proportion of methylated reads, π_{ij}, and number of total reads at that site, c_{ij}, such that $m_{ij} \sim bin(c_{ij}, p_{ij})$ for each site and individual. These count data are then modeled as a function of the predictor using the user-specified type of model.

III. References

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