Batch Effects Correction for Microbiome Data with Dirichlet-multinomial Regression

DAI ZHENWEI

22 August, 2018

1. Introduction

Metagenomic sequencing techniques enable quantitative analyses of the microbiome. However, combining the microbial data from these experiments is challenging due to the variations between experiments. The existing methods for correcting batch effects do not consider the interactions between variables—microbial taxa in microbial studies—and the overdispersion of the microbiome data. Therefore, they are not applicable to microbiome data.

We develop a new method, Bayesian Dirichlet-multinomial regression meta-analysis (BDMMA), to simultaneously model the batch effects and detect the microbial taxa associated with phenotypes. BDMMA automatically models the dependence among microbial taxa and is robust to the high dimensionality of the microbiome and their association sparsity.

The package BDMMA includes functions to perform meta-analysis of the metagenomic compositional data and select taxa significantly associated with the covariates. BDMMA is based on the following assumptions:

1. Only a small proportion of taxa are significantly associated with the covariates.
2. The taxonomic read counts follows a Dirichlet-Multinomial (DM) distribution.
3. The batch effects are indepdent of the covariates' effects.
4. The batch information for each sample is known.

Brief introduction to the BDMMA

Suppose the taxonomic read counts of a sample y_{ij} (the $j-th$ sample in $i-th$ batch) follow a DM distribution parameterized by $\gamma_{ij} = (\gamma_{ij1}, \gamma_{ij2}, ..., \gamma_{ijG})$,

$$f_{DM}(y_{ij}|\gamma_{ij}) = \frac{\Gamma(\gamma_{ij+})\Gamma(y_{ij+}+1)}{\Gamma(y_{ij+}+\gamma_{ij+})} \times \prod_{g=1}^{G} \frac{\Gamma(y_{ijg}+\gamma_{ijg})}{\Gamma(\gamma_{ijg})\Gamma(y_{ijg}+1)},$$

where $y_{ij+} = \sum_{g=1}^{G} \gamma_{ijg}$ and G encodes the number of all the taxa included in the analysis. We model the parameter γ_{ijg} with,

$$\gamma_{ijg} = \alpha_g \times \exp(\sum_{p=1}^{P} X_{ijp}\beta_{pg} + \delta_{ig}),$$

where g means the $g-th$ taxon; $X = (X_{ijp})_{N\times P}$ is the covariate matrix (N is the total number of samples and P is the number of covariate variables); $\boldsymbol{\delta} = (\delta_{ig})_{I\times G}$ is the batch effects matrix satisfying $\sum_{i=1}^{I} n_i\delta_{ig} = 0$ (n_i is the sample size of $i-th$ batch). α_g and β_{pg} encode the intercept and covariate coefficient respectively.

We adopt the Bayesian approach and provide proper prior distributions for the parameters. To select the taxa significantly associated with the variable of interest, we impose a spike-and-slab prior on the corresponding coefficients and estimate their posterior inclusion probability (PIP). The variable selection is conducted by thresholding PIP. In the next section, we provide an example to show the usage of functions in our package.

2. Analysis Example

The package includes a sample data set, named 'dat', which can be loaded directly. `dat` is a list containing four arguments, namely `X`: a `data.frame` of covariates containing a main effect variable in the 1st column and a confounding variable in the 2nd column; `Y`: a `data.frame` of the taxonomic read counts containing 80 samples and 40 taxa; `batch`: a numeric vector labeling the batch of each sample; `continuous`: a numeric vector indicating whether the variables are continuous($= 0$) or catagorical($\neq 0$).

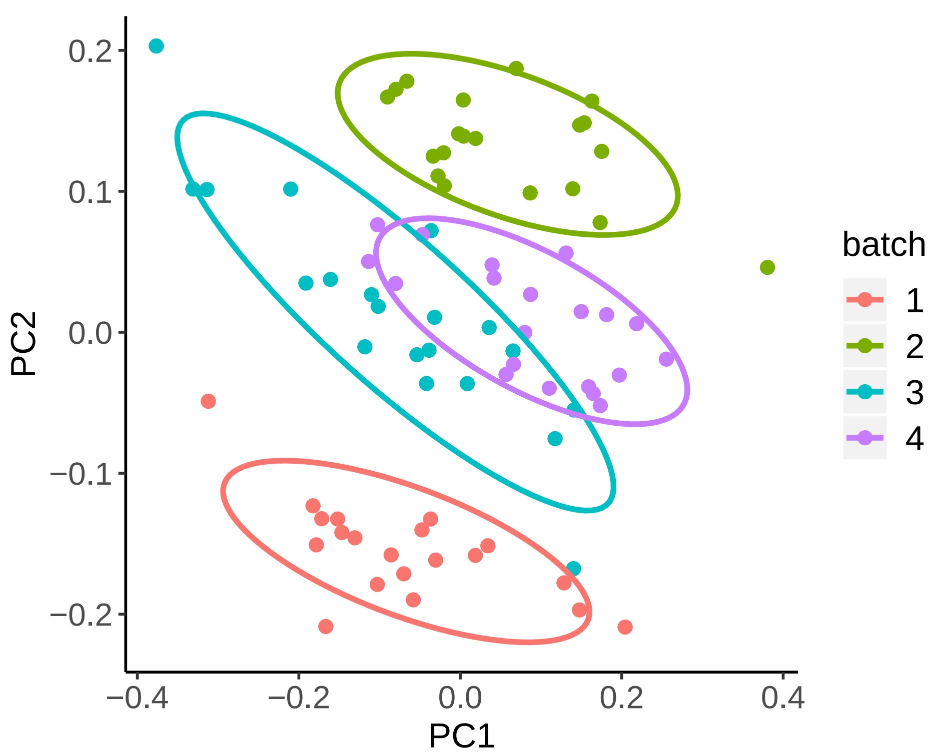
```
library(BDMMA)
```

```
data(dat)
attributes(dat)
```

```
## $names
## [1] "X"          "Y"          "batch"      "continuous"
```

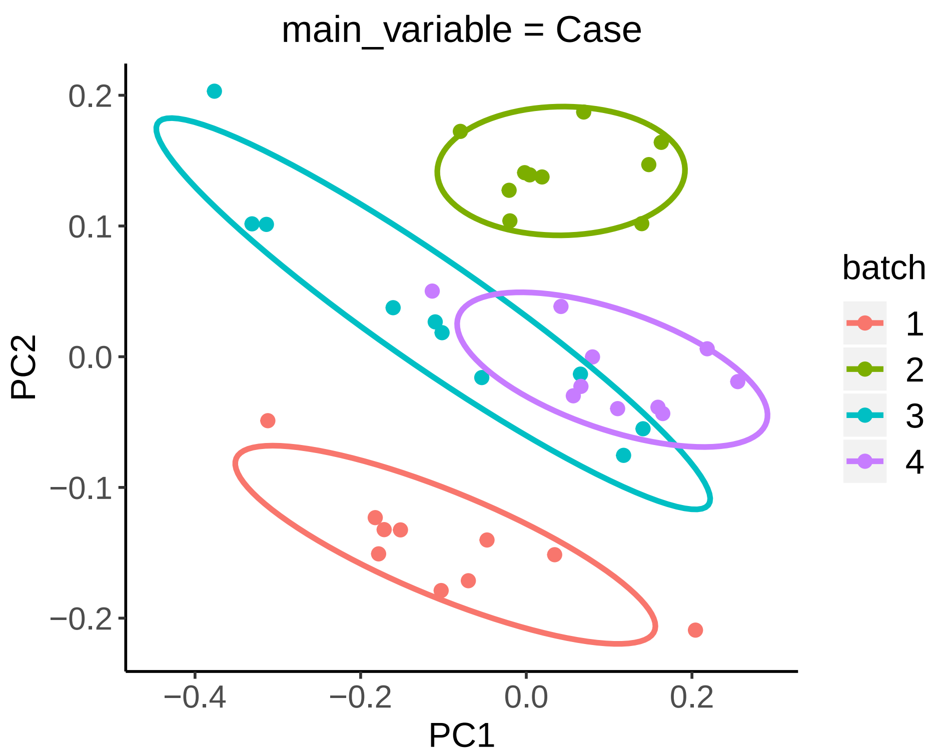
`BDMMA` provides a function to visualize the differences of the taxonomic composition across cohorts with the principal coordinate analysis. `VBatch` plots the first two principal coordinates of the corresponding samples and the 80% confidence ellipse of each batch.

```
figure = VBatch(dat$Y, batch = dat$batch, method = "bray")
print(figure)
```

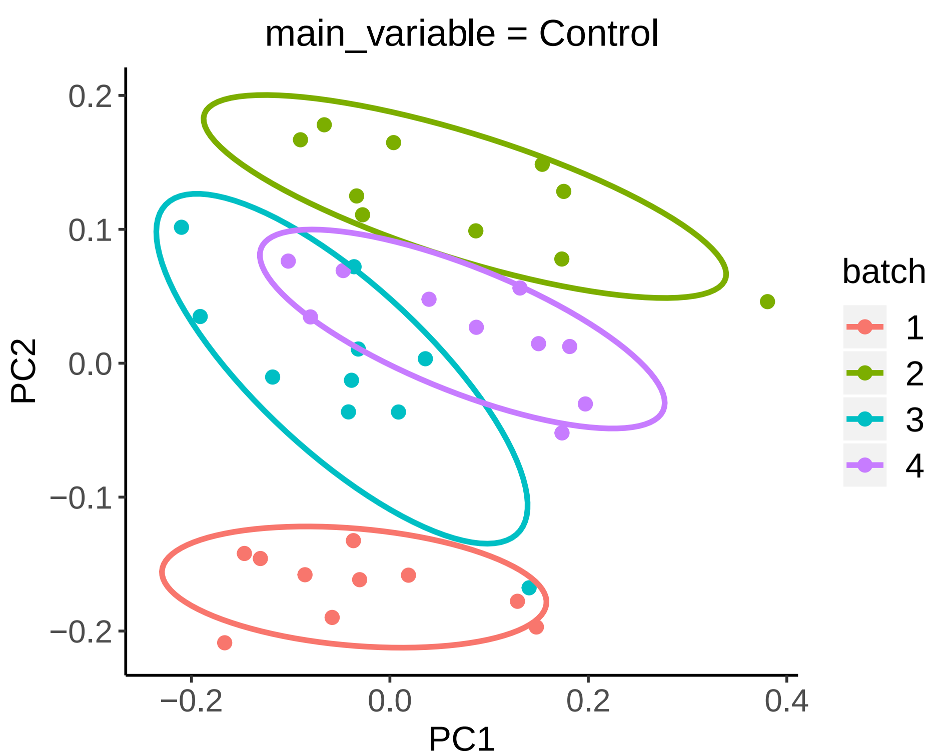


For a case/control study, `VBatch` can also visualize the batch effect of the case and control samples respectively.

```
main_variable = dat$X[,1]
main_variable[main_variable == 0] <- "Control"
main_variable[main_variable == 1] <- "Case"
figure = VBatch(dat$Y, batch = dat$batch, main_variable = main_variable, method = "bray")
print(figure[[1]])
```




```r
print(figure[[2]])
```



Then, the main function `BDMMA` can work directly on `dat`. `BDMMA` provides the freedom to ignore the low abundant taxa. `BDMMA` runs a Markov chain Monte Carlo algorithm to sample from the posterior distribution. The users can set the lengths of the burn-period and the sampling period for the Markov chain. In this example, the lengths of burn in and sampling period are set to 4000.

```
output <- BDMMA(dat$X, dat$Y, dat$batch, dat$continuous, burn_in = 4000, sample_period = 4000)
```

```
## #################### Start MCMC ####################
##
## Iteration = 1000
## Iteration = 2000
## Iteration = 3000
## Iteration = 4000
## Iteration = 5000
## Iteration = 6000
## Iteration = 7000
## Iteration = 8000
```

```
print(output$selected.taxa)
```

```
## $MIM
## [1] "V10" "V30"
##
## $bFDR
## [1] "V10" "V30"
```

```
head(output$parameter_summary)
```

```
##             mean    X2.5.     X25.     X50.     X75.   X97.5.
## alpha_1 8.818891 7.909825 8.484932 8.825488 9.151515 9.725720
## alpha_2 2.854233 2.499286 2.711751 2.844610 2.988345 3.290127
## alpha_3 4.572910 4.064181 4.385487 4.581013 4.763543 5.032157
## alpha_4 1.457261 1.223430 1.376422 1.457310 1.537859 1.706371
## alpha_5 2.536235 2.089981 2.387578 2.543544 2.683222 2.947916
## alpha_6 3.960763 3.426874 3.763716 3.970552 4.159734 4.469610
```

```
print(output$PIP)
```

```
## [1] 0.9490127 0.9235191
```

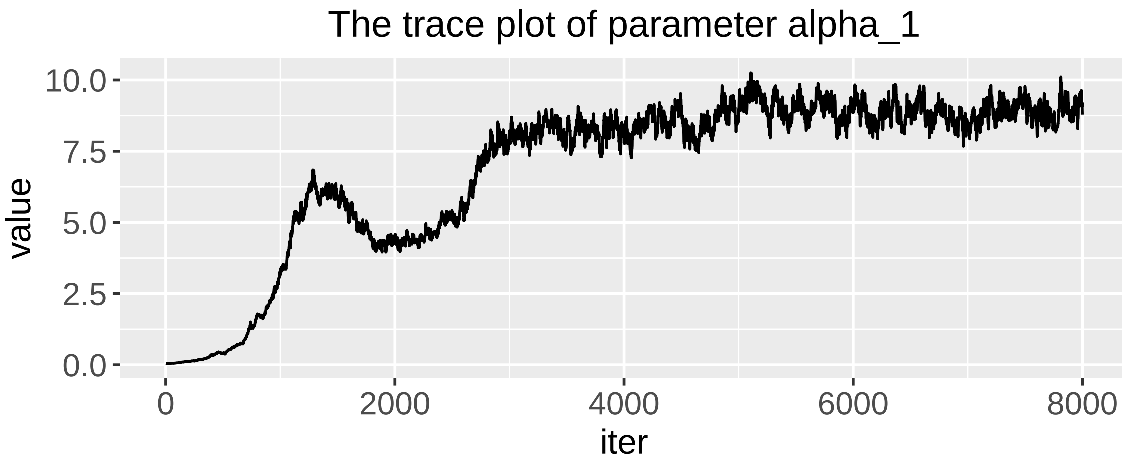
```
print(output$bFDR)
```

```
## [1] 0.06373407
```

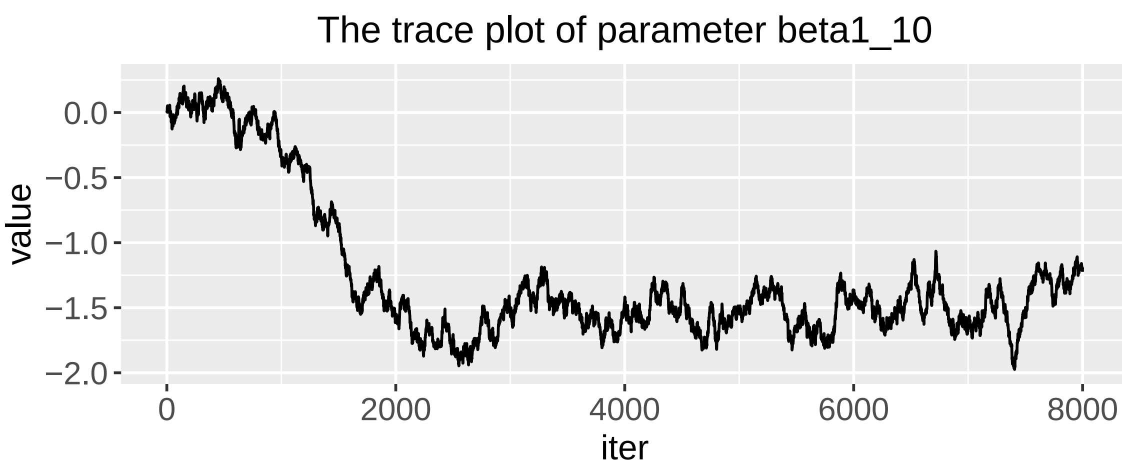
The output includes three arguments, selected.taxa, parameter_summary and trace. The selected taxa with median inclusion model (cut PIP larger than 0.5) and controling Bayesian false discovery rate are listed in `output$selected.taxa`. The default bFDR level is set to 0.1 in the function. `BDMMA` selects `V10` and `V30` as significantly associated taxa. Users can check the mean and quantiles of parameters' posterior distribution in `output$parameter_summary`. `output$PIP` shows the PIPs of the selected taxa. Given the selected microbial taxa, `output$bFDR` provides the corresponding bFDR. `output$trace` includes the trace of the parameters and the function `trace_plot` can be used to check the convergence of the markov chain.

```
figure <- trace_plot(trace = output$trace, param = c("alpha_1", "beta1_10"))
print(figure)
```

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


The trace plot of parameter alpha_1
10.0
7.5
5.0
2.5
0.0
value
0
2000
4000
6000
8000
iter

```
##
## [[2]]
```


The trace plot of parameter beta1_10
0.0
−0.5
−1.0
−1.5
−2.0
value
0
2000
4000
6000
8000
iter