BMC Bioinformatics

SOFTWARE

Open Access

NBZIMM: negative binomial and zero-inflated mixed models, with application to microbiome/metagenomics data analysis

Xinyan Zhang[1] and Nengjun Yi[2*]

*Correspondence:
nyi@uab.edu
[2] Department of Biostatistics, University of Alabama at Birmingham, Birmingham, AL 35294, USA
Full list of author information is available at the end of the article

Abstract

Background: Microbiome/metagenomic data have specific characteristics, including varying total sequence reads, over-dispersion, and zero-inflation, which require tailored analytic tools. Many microbiome/metagenomic studies follow a longitudinal design to collect samples, which further complicates the analysis methods needed. A flexible and efficient R package is needed for analyzing processed multilevel or longitudinal microbiome/metagenomic data.

Results: NBZIMM is a freely available R package that provides functions for setting up and fitting negative binomial mixed models, zero-inflated negative binomial mixed models, and zero-inflated Gaussian mixed models. It also provides functions to summarize the results from fitted models, both numerically and graphically. The main functions are built on top of the commonly used R packages nlme and MASS, allowing us to incorporate the well-developed analytic procedures into the framework for analyzing over-dispersed and zero-inflated count or proportion data with multilevel structures (e.g., longitudinal studies). The statistical methods and their implementations in NBZIMM particularly address the data characteristics and the complex designs in microbiome/metagenomic studies. The package is freely available from the public GitHub repository https://github.com/nyiuab/NBZIMM.

Conclusion: The NBZIMM package provides useful tools for complex microbiome/metagenomics data analysis.

Keywords: Microbiome, Metagenomics, NBZIMM, Negative binomial mixed models, Zero-inflated mixed models

Background

The recent development of technology and computational tools promotes the generation microbiome/metagenomic data, providing research opportunities to identify the links between the microbiome and diseases [1]. 16S rRNA and whole-metagenome shotgun sequencing data are two types of microbiome/metagenomic data available [2, 3]. The downstream bioinformatics pipelines will convert the raw microbiome/metagenomics

sequence data into operational taxonomic units (OTUs) as count data (QIIME and mothur) or functional pathways as count data (Kraken) or proportion data (MetaPhlAn) [2, 3]. Processed microbiome/metagenomics data is over-dispersed and sparse, and has various sequencing depths, thus, it is needed to have analytic tools to address those features [4, 5]. Moreover, the number of measured taxa is usually large, thus requiring efficient algorithms for detecting significant taxa. In addition to these data features, many microbiome/metagenomic studies focused on investigated the temporal relationship of microbiome among subjects [1, 6]; for instance, they used longitudinal designs to collect multiple samples at various time points from the same subject. Thus, it is needed to account for correlation over time within and between subjects.

Various generalized linear or mixed models have been proposed to analyze the processed microbiome/metagenomic data. Negative binomial and zero-inflated negative binomial distributions are commonly used to address the over-dispersion and sparsity issues in count data, and mixed models are the standard approaches for dealing with multilevel data structures [7, 8]. Zero-inflated Gaussian distributions can be used to analyze the transformed zero-inflated count or proportion data, and zero-inflated Gaussian mixed models can handle multilevel data structures. An R package **metagenomeSeq** is available to analyze the transformed microbiome data with zero-inflated Gaussian models [9]. Meanwhile, the following R packages are available to analyze over-dispersed or sparse count data, including **pscl**, **mgcv**, **brms**, **gamlss**, **GLMMadaptive**, and **glmmTMB** [10–14]. The **pscl** package is a popular package in fitting zero-inflated and hurdle model but cannot handle repeated measures or longitudinal studies. The package **mgcv** can fit negative binomial mixed models (NBMMs), but the value for the dispersion parameter in NBMMs needs to be specified. Also, NBMMs fitted by **mgcv** can only include simple random effects without the options to handle within-subject residual correlation structures [7]. Besides, using **mgcv**, we can use zero-inflated Poisson mixed models to analyze sparse data but may not be appropriate to model the sparse over-dispersed count data. The other R packages can set up both NBMMs and zero-inflated negative binomial mixed models (ZINBMMs) but may not be ideal in analyzing microbiome/metagenomics data due to computational efficiency. The **brms** package fits the models using MCMC sampling, which is the slowest algorithm among the four packages. The package **gamlss** uses a Newton–Raphson or Fisher-scoring based algorithms [12]. But with the presence of random effects, the standard deviations for fixed effects fitted by **gamlss** are largely under-estimated. **GLMMadaptive** and **glmmTMB** use adaptive Gaussian quadrature and Laplace approximation, respectively, to fit ZINBMMs. Moreover, none of the above-mentioned R packages are designed to analyze microbiome/metagenomic data, thus we cannot directly summarize or visualize the analysis results of many taxa or functional units with those R packages. Alternatively, Chen and Li [15] proposed a zero-inflated Beta regression model with random effects (ZIBR) for analyzing longitudinal microbiome proportion data, which is implemented in the R package **ZIBR** [16]. However, **ZIBR** requires that the number of time points be the same for various subjects.

A flexible and efficient R package is needed for analyzing processed multilevel or longitudinal microbiome/metagenomic data. Here, we introduce the freely available R package **NBZIMM** for analyzing processed complex microbiome/metagenomic data, which

implements our published methods [5, 17, 18]. In our previous work, we have evaluated and compared the three methods NBMMs, ZINBMMs and zero-inflated Gaussian (linear) mixed models (ZIGMMs) implemented in **NBZIMM** with various existing methods in different R packages. We found that ZINBMMs in **NBZIMM** is improved in computational time but was comparable in statistical analysis with two other R packages **GLMMadaptive** and **glmmTMB** to fit ZINBMMs [10, 11, 18]. We also found that ZINBMMs outperform NBMMs and ZIGMMs in power for sparse data but may be similar to the other methods in not highly sparse microbiome data through simulation studies [18]. We also have evaluated the methods in **NBZIMM** in public available datasets and found that the taxon identified by NBMMs and ZINBMMs are mostly either overlapped with the original paper or have been reported in previous research [5, 17, 18]. **NBZIMM** provides three main functions to fit NBMMs, ZINBMMs, and ZIGMMs. The main difference between **NBZIMM** and some other aforementioned R packages is the model-fitting algorithm. The algorithms used to fit NBMMs, ZINBMMs, and ZIGMMs in **NBZIMM** are all fast and stable [5, 17, 18]. It also has functions for numerically or graphically summarizing the results. The main functions are developed based on the commonly used R packages, **MASS**, and **nlme**, for analyzing negative binomial models and linear mixed models (LMMs) [7, 8], and inherit powerful features of the standard tools. The main functions are flexible to include various forms of fixed and random effects and allow the specification of within-subject residual correlation structures [7]. Here, we are presenting **NBZIMM** as an efficient and flexible tool and providing a detailed demonstration of how to use **NBZIMM** in analyzing processed longitudinal microbiome/metagenomic data.

Implementations

We describe our models and algorithms with a two-level design where samples are grouped in subjects. Assume that a microbiome/metagenomic study collects n subjects and n_i samples for the i-th subject. For each sample, we measure the counts for m taxa (OTU, species, genus, etc.), C_{ijh}; $h = 1, \cdots, m$, the total sequence read T_{ij}, and some relevant covariates X_{ij}. The goal is to identify if there is any microbiota taxa could be lined with covariates of interest in a study.

The function `glmm.nb` in **NBZIMM** allows us to analyze the data for taxon h with NBMMs:

$$C_{ijh} \sim \text{NB}(C_{ijh}|\mu_{ijh}, \theta_h),\ \log \mu_{ijh} = \log(T_{ij}) + X_{ij}\beta_h + G_{ij}b_{ih},\ b_{ih} \sim N(0, \Psi_h) \qquad (1)$$

where the dispersion θ_h determines the over-dispersion, the offset $\log(T_{ij})$ accounts for the varying total sequence reads, β_h is a vector of fixed effects, and b_{ih} are random effects. G_{ij} denotes the vector of group-level covariates. `glmm.nb` iteratively approximates the NBMMs by fitting a linear mixed model using the function `lme` from the package **nlme**. The dispersion θ_h is then updated using Newton–Raphson algorithm as in the function `glm.nb` of **MASS**. This framework allows us to incorporate the powerful features of `lme` into NBMMs.

The function `glmm.zinb` implements ZINBMMs that directly model true zeros and can be more efficient for analyzing taxa with excessive zeros than `glmm.nb`:

$$C_{ijh} \sim \begin{cases} 0 & \text{with probability } p_{ijh} \\ \text{NB}(C_{ijh}|\mu_{ijh}, \theta_h) & \text{with probability } 1 - p_{ijh} \end{cases} \quad (2)$$

Here, the means μ_{ijh} are modeled as above, and the zero-inflation probabilities p_{ijh} are assumed to depend on some covariates via a logistic regression $\text{logit}(p_{ijh}) = Z_{ij}\alpha_h$ or logistic mixed model $\text{logit}(p_{ijh}) = Z_{ij}\alpha_h + G_{ij}a_{ih}$, where Z_{ij} denotes the potential covariates associated with the excess zeros, α_h is a vector of fixed effects and the random effects $a_{ih} \sim \text{N}(0, \Phi_h)$. To fit the ZINBMMs, `glmm.zinb` uses an efficient and stable EM-IWLS algorithm, making use of the function `lme` from the package **nlme**.

Rather than directly analyzing the observed counts, some methods analyze transformed count data [9], for example, $y_{ijh} = \log_2(C_{ijh} + 1)$. Also, in some bioinformatics pipeline, such as MetaPhlAn, the whole metagenome shotgun sequencing data are processed and output in terms of relative abundance as proportion data. The transformation for proportion data is commonly chosen as arcsine square root transformation $y_{ijh} = \text{arcsine}\left(\sqrt{C_{ijh}/T_{ij}}\right)$. The function `lme.zig` in **NBZIMM** implements ZIGMMs for the transformed count or proportion data:

$$y_{ijh} \sim \begin{cases} 0 & \text{with probability } p_{ij} \\ N(y_{ijh}|\mu_{ijh}, \sigma^2) & \text{with probability } 1 - p_{ij} \end{cases} \quad (3)$$

where μ_{ijh} denotes the means and p_{ijh} are the zero-inflation probabilities. The function `lme.zig` uses an efficient EM algorithm to fit ZIGMMs.

The data inputs and specifications of fixed and random effects in the above functions are similar to `lme`, allowing us to incorporate all the forms of fixed and random effects implemented in `lme` into our models. In the functions `glmm.zinb` and `lme.zig`, there are two options to include various forms of fixed and random effects in the zero-inflated part. The function `lme` can specify various forms of within-subject residual correlation structures [7], for instance, AR(1), which also have been incorporated into our framework. Moreover, the models fitted by our main functions, `glmm.nb`, `glmm.zinb`, and `lme.zig`, can be directly fed to the function `summary` from **nlme**, which returns the estimates, standard deviations and p values of fixed effects, and the estimates of the variances of random effects, etc. These features made **NBZIMM** easy to use, flexible and comprehensive in modeling and stable and efficient in computation.

Although the above three functions only model one taxon at a time, **NBZIMM** includes a wrapper function `mms` to screen all included taxa, by repeated calling to `glmm.nb`, `glmm.zinb`, `lme`, or `lme.zig`. The function `fixed` extracts the estimates, standard deviations and p values for fixed effects of all the taxa and covariates, while `get.fixed` extracts the estimates, standard deviations and p values of fixed effects for a given taxon or covariate. The **NBZIMM** package provides two functions to graphically display the analytic results from `mms`. The function `plot.fixed` plots the estimates, intervals, and p values for numerous fixed effects. It uses different colors to distinguish between significant and insignificant effects. The function `heat.p` displays **ggplot2**-based heat map to visualize p values and positive or negative signs of significant effects for numerous taxa and multiple covariates.

Results

In the recent published microbiome studies [19, 20], both sequence data and processed abundance data tables were made available. **NBZIMM** is only capable to analyze the processed abundance data tables. In this section, we will demonstrate the use of NBMMs, ZINBMMs, and ZIGMMs in analyzing processed longitudinal microbiome/metagenomics data. The example datasets in the demonstrations are consisted of three components: clinical data, taxa abundance data table (such as OTU microbiome data) and taxonomy table. The taxa abundance data that **NBZIMM** could analyze are processed abundance data table such as the file in the HMP DACC (https://www.hmpdacc.org/hmsmcp2/) [19].

Demonstrations of NBMMs and ZINBMMs for directly analyzing longitudinal microbiome/metagenomics count data

NBMMs could analyze over-dispersed longitudinal microbiome/metagenomics count data. We will describe the functions *glmm.nb* and *mms* and show how to set up the NBMMs with those two functions. The functions *glmm.nb* and *mms* work by calling the function *lme* from package **nlme** repeatedly to fit LMMs, so both functions have the same options to include different forms of random effects and within-subject correlation structures as in the function *lme*.

To address the sparsity issue in longitudinal microbiome/metagenomics count data, ZINBMMs is also available to analyze over-dispersed and zero-inflated longitudinal microbiome/metagenomic count data in **NBZIMM**. The functions *glmm.zinb* and *mms* in **NBZIMM** are created to fit the ZINBMMs. The function *glmm.zinb* and *mms* call to the function *lme* from the package **nlme** repeatedly to fit the weighted linear mixed model. And they also call *glm* in the package **stats** or *glmPQL* in **MASS** to fit the logistic regression or logistic mixed model. Similar to the function *glmm.nb*, the specification of random effects and within-subject correlation structures in *glmm.zinb* and *mms* are the same as described in the function *lme*.

Here, we demonstrate the use of *glmm.nb*, *glmm.zinb*, and *mms* to analyze an example data with NBMMs and ZINBMMs. The microbiome data is publicly available from a case–control longitudinal study, which is designed to explore the differences of the vaginal microbiota composition between pregnant and non-pregnant women [6]. This study included 22 pregnant women who delivered at term and 32 non-pregnant women and collected samples at multiple time points. The data consist of two data components: OTU and Clinical Data. OTU contains microbiome count data for 900 samples with 143 taxa; Clinical Data contains data of clinical variables for all the samples with 9 variables, including subject ID, pregnant status, total sequencing read, age, race, etc. In the R package, Clinical Data is saved in the data component with the name of SampleData.

```
data(Romero) # see help("Romero")
names(Romero)
[1] "OTU"        "SampleData"
```

In the SampleData component, there are 9 clinical variables; variable *Subject_ID* is the subject ID, variable *pregnant* represents pregnant status, variable *Total.Read. Counts* are the total sequencing read, variable *GA_Days* indicates the time of sample collection, and there are two other covariates age and race included in the dataset.

```
head(Romero$SampleData)
  Subject_ID Sample_ID GA_Days Age Race Nugent.Score CST Total.Read.Counts
1       N001     33604   19.29  19    1            0  II              4338
2       N001     35062   23.29  19    1            0  II              4610
3       N001     36790   27.71  19    1            0  II              3596
4       N001     37504   29.71  19    1            0  II              4405
5       N001     39108   33.71  19    1           NA  II              3878
6       N001     40105   36.00  19    1            0 III              4610
  pregnant
1        1
2        1
3        1
4        1
5        1
6        1
```

The function `nonzero` calculates the proportion of non-zero values for each response and plots the proportions.

```
non = nonzero(y = Romero$OTU, total = N, plot = F)
```

Analysis of a single taxon each time

We first show the analysis of a single taxon (Lactobacillus) with the function `glmm.nb` and `glmm.zinb`, respectively. The case–control group variable is defined as pregnancy women vs non-pregnant women. We consider the sample collection time as the time variable and also include covariates, age, and race. The main research goal is to explore the association between the bacterial taxonomic composition for the vaginal microbiota and the group variable of interest. Different models can be used to explore the association. The following codes are to run two examples of using a random intercept model by assuming negative binomial or zero-inflated negative binomial distributions. An offset term is needed to adjust for the library size N when analyzing a longitudinal microbiome count data. By assuming negative binomial distribution for the OTU, we can model it with `glmm.nb`:

```
y = Romero$OTU[, names(nonzero.p)[1]] # Lactobacillus
f1 = glmm.nb(y ~ GA_Days + Age + Race + pregnant +
             offset(log(Total.Read.Counts)), data = Romero$SampleData,
             random = ~ 1 | Subject_ID)
```

Or assuming zero-inflated negative binomial distribution, we can model it with *glmm.zinb*.

```
f2 = glmm.zinb(y ~ GA_Days + Age + Race + pregnant +
             offset(log(Total.Read.Counts)), data = Romero$SampleData,
             random = ~ 1 | Subject_ID, zi_fixed = ~1, zi_random =NULL)
```

In both functions, we have a *fixed* term to include the formula for the fixed-effects part to specify the outcome and various predictors. We also provide a *random* term to include the formula for the random-effects part. It only contain the right-hand side part, e.g., ~1 | subject. The *data* term is to include a data frame with all the variables. In function *glmm.zinb*, there are two terms *zi_fixed* and *zi_random*, which are to specify the formula for the fixed and random effects of the zero-inflated part. The two terms, *zi_fixed* and *zi_random*, both only contain the right-hand side part. The following codes show two examples of using NBMMs and ZINBMMs to analyze the single taxon (Lactobacillus). The first example is a random slope model with NBMMs. The second example is a random slope model while controlling a covariate in the fixed effects of zero-inflated part using ZINBMMs.

```
#NBMMs
f3 = glmm.nb(y ~ GA_Days + Age + Race + pregnant +
             offset(log(Total.Read.Counts)), data = Romero$SampleData,
             random = ~ list(Subject_ID = pdDiag(~GA_Days)))
#ZINBMMs
f4 = glmm.zinb(y ~ GA_Days + Age + Race + pregnant +
             offset(log(Total.Read.Counts)), data = Romero$SampleData,
             random = ~ list(Subject_ID = pdDiag(~GA_Days)),
             zi_fixed = ~pregnant, zi_random = NULL)
```

We can also incorporate autoregressive of order 1, AR(1) for correlation matrix R_i to describe dependence among observations in the *correlation* term. The correlation matrix is not restricted to be AR(1) but can take any form that is available in *corClasses* from **nlme**.

```
#NBMMs
f5 = glmm.nb(y ~ GA_Days + Age + Race + pregnant +
             offset(log(Total.Read.Counts)), data = Romero$SampleData,
             random = ~ 1 | Subject_ID, correlation = corAR1())
#ZINBMMs

f6 = glmm.zinb(y ~ GA_Days + Age + Race + pregnant +
             offset(log(Total.Read.Counts)), data = Romero$SampleData,
             random = ~ 1 | Subject_ID, zi_fixed = ~1,
             zi_random = NULL, correlation = corAR1())
```

To summarize the results we use *summary* function.

```
summary(f1) #NBMMs
Linear mixed-effects model fit by maximum likelihood
 Data: NULL
  AIC BIC logLik
   NA  NA     NA

Random effects:
 Formula: ~1 | subject
        (Intercept)  Residual
StdDev:    1.958786 0.9942486

Variance function:
 Structure: fixed weights
 Formula: ~invwt
Fixed effects: y ~ Days + Age + Race + preg + offset(log(N))
                Value Std.Error  DF   t-value p-value
(Intercept) -4.528065 0.4688713 842 -9.657374  0.0000
Days        -0.012375 0.0372144 842 -0.332527  0.7396
Age         -0.545787 0.3800906  50 -1.435940  0.1572
Race        -0.138350 0.3042039  50 -0.454794  0.6512
preg        -0.955077 0.7571598  50 -1.261394  0.2130
 Correlation:
     (Intr) Days   Age    Race
Days -0.007
Age  -0.378  0.001
Race -0.636  0.002  0.323
preg -0.610  0.040  0.681  0.227

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-1.0384100 -0.6681150 -0.2068998  0.3441043  8.4823027

Number of Observations: 897
Number of Groups: 54
```

Analysis of all taxa with a given nonzero proportion with function mms

For the microbiome/metagenomics data, it is more common that we have many taxa of interest, it is easier to analyze them all at one time with the function *mms*. For both NBMMs and ZINBMMs, the function *mms* can analyze multiple taxa at one time. In this function, *y* term is to include a matrix of responses for all taxa of interest; the *fixed* term here is to set a one-sided formula of the form ~ x (i.e., the response is omitted); the right side of ~ is the same as in *glmm.nb*, or *glmm.zinb*.

The terms *data*, *random*, *correlation*, *zi_fixed*, and *zi_random* are the same as in the functions *glmm.nb*, or *glmm.zinb*. The term *Method* is to specify the distribution to be assumed for the response data, such as "nb" for negative binomial or "zinb" for zero-inflated negative binomial. The term *sort* is to sort by the nonzero proportions of the responses into decreasing order. The term *min.p* is to set an inclusion criteria to analyze taxa with a given nonzero proportion. The following

example analyze all the taxa with the proportion of non-zero values > 0.2 through the term *min.p* with NBMMs and ZINBMMs.

```
#NBMMs
f7 = mms(y = Romero$OTU, fixed = ~ GA_Days + Age + Race + pregnant +
        offset(log(Total.Read.Counts)), data = Romero$SampleData,
        random = ~ 1 | Subject_ID, min.p = 0.2, method = "nb")
#ZINBMMs
f8 = mms(y = Romero$OTU, fixed = ~ GA_Days + Age + Race + pregnant +
        offset(log(Total.Read.Counts)), data = Romero$SampleData,
        random = ~ 1 | Subject_ID, min.p = 0.2, method = "zinb",
        zi_fixed = ~1, zi_random = NULL)
```

Visualize the results

To visualize the results through **NBZIMM**, there are several options available in the package. First, we can generate a plot with function *plot.fixed* to view the significant taxa associated with various covariates and corresponding *p* values. We will show the example plots for both NBMMs and ZINBMMs in Figs. 1 and 2. And comparing Figs. 1 and 2 in analyzing the taxa from Romero et al. [6], the results from NBMMs and ZINBMMs are similar with slight differences in *p* values.

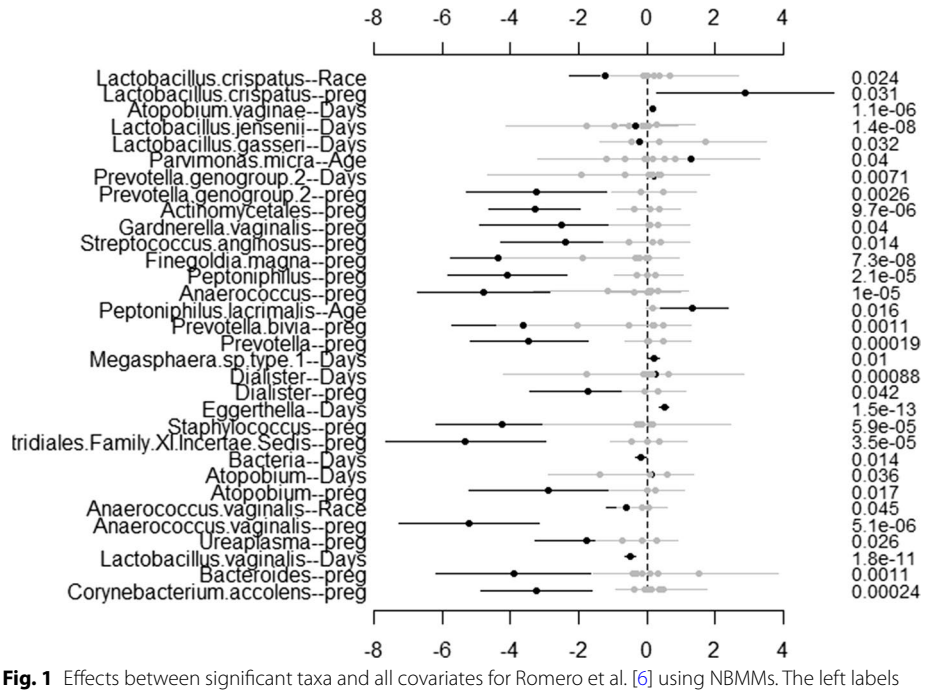

Fig. 1 Effects between significant taxa and all covariates for Romero et al. [6] using NBMMs. The left labels the taxa name and corresponding covariate. The right labels the *p* value

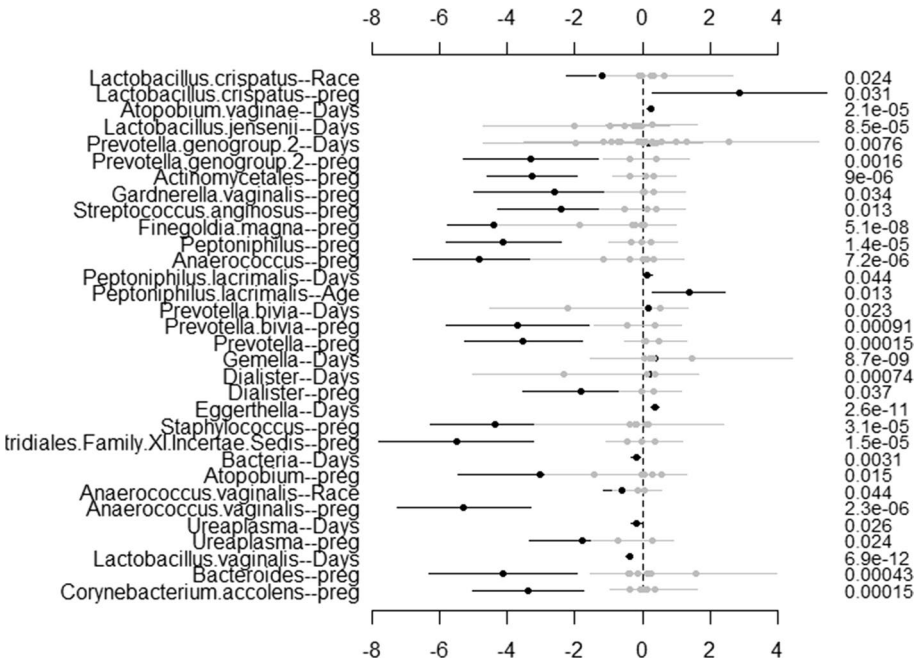

Fig. 2 Effects between significant taxa and all covariates for Romero et al. [6] using ZINBMMs. The left labels the taxa name and corresponding covariate. The right labels the *p* value

```
# NBMMs
out = fixed(f7)$dist
out = out[out[,2]!="(Intercept)", ]
res = out[, 3:5]
par(mfrow = c(1, 1), cex.axis = 1, mar = c(2, 14, 4, 4))
plot.fixed(res=res, threshold=0.05, gap=500, col.pts=c("black",
"grey"), cex.axis=1, cex.var=1)
# ZINBMMs
out = fixed(f8)$dist
out = out[out[,2]!="(Intercept)", ]
res = out[, 3:5]
par(mfrow = c(1, 1), cex.axis = 1, mar = c(2, 14, 4, 4))
plot.fixed(res=res, threshold=0.05, gap=500, col.pts=c("black",
"grey"), cex.axis=1, cex. var=1)
```

The second option is that we can choose only to display the significant taxa specially associated with a covariate of interest. To generate this plot, the object `res` needs to be updated with the function `get.fixed` first and then plotted using the function `plot.fixed`. The following code is to generate Fig. 3 to visualize the associations between the variable pregnant and the taxa from Romero et al. [6] using NBMMs.

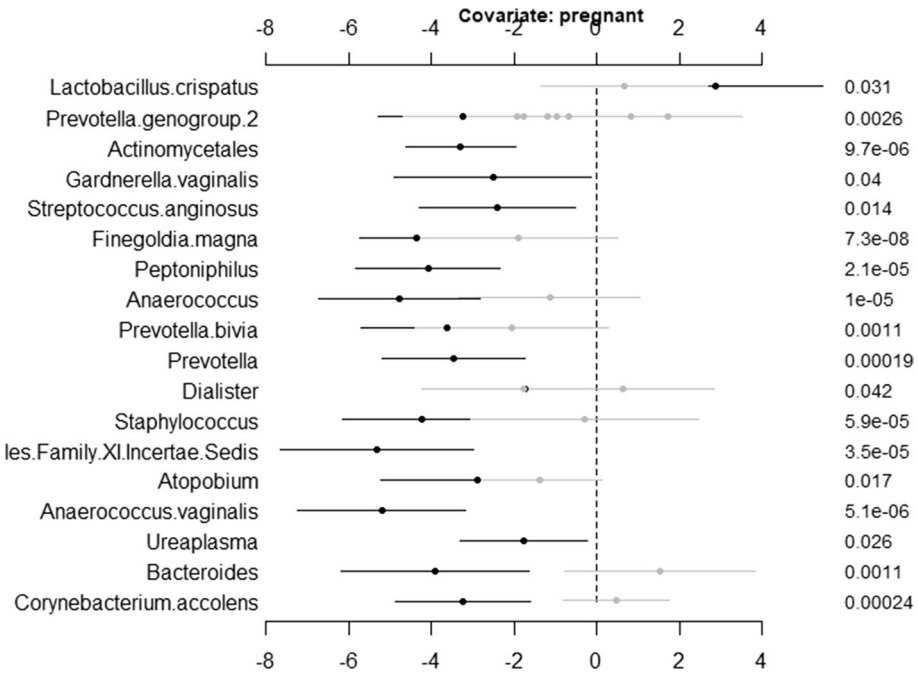

Fig. 3 Effects between significant taxa and covariate Pregnant for Romero et al. [6] using NBMMs. The left labels the taxa name and corresponding covariate. The right labels the *p* value

```
# NBMMs
res = get.fixed(f7, part="dist", vr.name="preg", sort=F)
par(mfrow = c(1, 1), cex.axis = 1, mar = c(2, 10, 4, 4))
plot.fixed(res, threshold=0.05, gap=300, main="Covariate: pregnant",
           cex.axis=1, cex.var=1)
```

Moreover, another option is available to generate a heat map to present the results through the function *heat.p*. The following code is to generate a heat map to visualize the associations between all four covariates and the taxa from Romero et al. [6] using NBMMs in Fig. 4.

```
# NBMMs
out = fixed(f7)$dist
out = out[out[,2]!="(Intercept)", ]
heat.p(df=out, zigzag=c(T,F), abbrv=c(T,F), margin=c(1.5,10),
y.size=8)
```

Demonstrations of ZIGMMs in analyzing longitudinal microbiome/metagenomics count data

Longitudinal microbiome/metagenomics count data can be analyzed with ZIGMMs in **NBZIMM** through the functions *lme.zig* and *mms*. For count data, ZIGMMs analyze

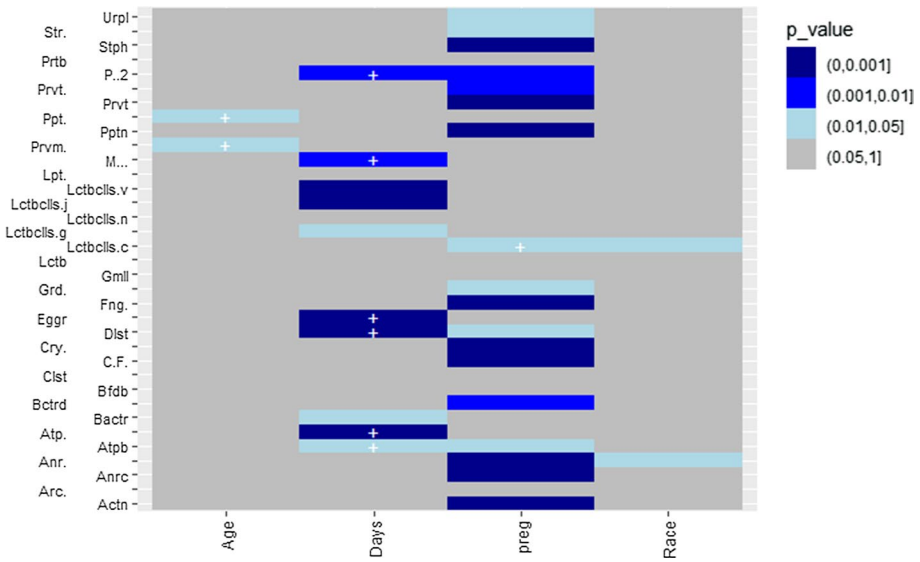

Fig. 4 A heat map for *p* values from analyzing Romero et al. [6] with NBMMs. The taxa on y-axis are abbreviated. The sign "+" indicates a positive effect

the count data with the transformation $\log_2(C_{ijh}+1)$. The function `lme.zig` calls the function `lme` repeatedly from the R package **nlme** and `glm` or `glmPQL` from the package **MASS**. `lme` fits the weighted linear mixed model; and `glm` or `glmPQL` fits the binomial logistic or mixed logistic model. We demonstrate the use of ZIGMMs in analyzing the same public microbiome count data from Romero et al. [6].

Analysis of a single taxon each time

We first show the analysis of a single taxon (Lactobacillus) with the function `lme.zig`. The group variable of interest and covariates are the same as mentioned in "Demonstrations of NBMMs and ZINBMMs for directly analyzing longitudinal microbiome/metagenomics count data". section The following code is an example of analyzing a single taxon with ZIGMMs using a random intercept. An offset term is needed to adjust for the library size N when analyzing transformed longitudinal microbiome count data.

```
y = otu[, names(nonzero.p)[1]] # Lactobacillus
y = log2(y + 1)
f9 = lme.zig(y ~  GA_Days + Age + Race + pregnant +
             offset(log(Total.Read.Counts)), data = Romero$SampleData,
             random = ~ 1 | Subject_ID, zi_fixed = ~1, zi_random = NULL)
```

In function `lme.zig`, we also have `fixed` and `random` terms to include the formula for the fixed-effects and random-effects parts. The `data` term is to include a data frame with all the variables. Also, in this function, we can add fixed and random effects to the zero-inflated part for ZIGMMs through terms `zi_fixed` and `zi_random`. We can incorporate any correlation matrix R_i available in `corClasses` from **nlme** to describe dependence among observations through the `correlation` term. The following codes show two examples of using ZIGMMs to analyze a single taxon. The first example is

random slope model incorporated a correlation structure AR(1). The second example is a random intercept model while controlling a covariate in the fixed effects of zero-inflated part.

```
f10 = lme.zig(y ~  GA_Days + Age + Race + pregnant +
              offset(log(Total.Read.Counts)), data = Romero$SampleData,
              random = list(Subject_ID = pdDiag(~GA_Days)),
              correlation = corAR1(), zi_fixed = ~1, zi_random = NULL)

f11 = lme.zig(y ~  GA_Days + Age + Race + pregnant +
              offset(log(Total.Read.Counts)), data = Romero$SampleData,
              random = ~ 1 | Subject_ID, zi_fixed = ~ pregnant,
              zi_random = NULL)
```

Similar as *glmm.nb* and *glmm.zinb*, we can also summarize the results using *summary* function for *lme.zig*.

```
summary(f9)
```

Analysis of all taxa with a given nonzero proportion with function mms

Similarly, for many taxa of interest, we analyze them all at one time with the function *mms*. The term *Method* needs to be set as 'zig' for ZIGMMs. The following example analyzes all the taxa with the proportion of non-zero values > 0.2 through the term *min.p* using ZIGMMs.

```
otu_log = apply(Romero$OTU, 2, function(x) as.numeric(log2(x+1)))

f12 = mms(y = otu_log, fixed = ~ GA_Days + Age + Race + pregnant +
        offset(log(Total.Read.Counts)), data = Romero$SampleData,
        random = ~ 1 | Subject_ID, zi_random = NULL,
        min.p = 0.2, method = "zig")
```

Besides, for visualization of the results, the options are the same as for NBMMs and ZINBMMs.

Demonstrations of ZIGMMs in analyzing longitudinal microbiome/metagenomics proportion data

ZIGMMs is also applicable in analyzing longitudinal microbiome/metagenomics proportion data with arcsine square root transformation through the functions *lme.zig* and *mms*. We will demonstrate the use of ZIGMMs in analyzing longitudinal microbiome/metagenomics proportion data through the following public data demonstration. This data demonstration used a published dataset from Vincent et al. [21]. Vincent et al. [21] collected fecal samples from 98 subjects and used whole metagenome shotgun sequencing to examine the composition of their fecal microbiota. The prospective cohort study was carried out among 8 patients who were Clostridium difficile infected or colonized (CDI) and other 90 patients. The clinical covariates in the dataset are gender, age, and days from first collection of the fecal samples. The dataset was downloaded

with R package curatedMetagenomicData, which contains clinical data and metagenomic data [22]. The metagenomic data is in the form of proportion data. So, the proportion data will be transformed with arcsine square root transformation to be analyzed by ZIGMMs. First, we will load the data object "vincent2016.RData" into R and check the clinical covariates. The data object includes two parts, 'clinical' and 'out'. There are 4 clinical covariates included in the data object 'clinical'; variable *subjectID* is the subject ID, variable *days_from_first_collection* indicates the time of sample collection, variable *number_reads* is the total sequencing read, and there are two other covariates age and gender included in the data object 'clinical'. Also, we generated a grouping variable for patients with CDI vs controls and scaled the matrix of metagenomics data by 100 to relative proportions.

```
load("vincent2016.RData")

group = ifelse(clinical$study_condition == "CDI", 1, 0)
N = clinical[, "number_reads"]  # total reads
subject = clinical[, "subjectID"]
age = clinical[, "age"]
gender = clinical[, "gender"]
days = clinical[, "days_from_first_collection"]
clinical = cbind.data.frame(group, N, subject, age, gender, days)

pheno = as.matrix(t(otu)) # transpose the microbiome/metagenomic  data
pheno = apply(pheno, 2, function(x) as.numeric(x)/100)
non = nonzero(y = pheno, total = N, plot = F)
nonzero.p = non[[1]]
```

Analysis of a single taxon each time

We first show the analysis of a single taxon (Prevotella Bivia) with the function `lme.zig`. We are interested to compare CDI with healthy controls, which defines the group variable of interest in this example. We consider the sample collection time as the time variable and also include covariates, age, and gender. The main research goal is to explore the diversity and composition of the fecal microbiota between CDI vs healthy controls. Different models can also be used to explore the relationship. The following code is an example of a random intercept model using ZIGMMs. An offset term should not be included as we are analyzing proportion data. In function `lme.zig`, we have described the terms `fixed`, `random`, `data`, `zi_fixed`, `zi_random`, and `correlation` in "Demonstrations of ZIGMMs in analyzing longitudinal microbiome/metagenomics count data" section. By assuming zero-inflated Gaussian distribution for the transformed relative abundance proportion data, we can model it with `lme.zig` as:

```
y = pheno[, names(nonzero.p)[1]] # Prevotella_bivia
y = asin(sqrt(y))
f13 = lme.zig(y ~ group + days  + age + gender,
              random = ~ 1 | subject, data = clinical, zi_fixed = ~1)
```

The following codes show another two examples of using ZIGMMs to analyze the relative abundance proportion for a single taxon. The first example is random slope model incorporated a correlation structure AR(1). The second example is a random intercept model while controlling the group variable in the fixed effects of zero-inflated part. We can then also summarize the results using *summary* function for the results generated by *lme.zig*.

```
f14 = lme.zig(y ~ group + days  + age + gender,
              random = list(subject = pdDiag(~days)),
              correlation = corAR1(), data = clinical,
              zi_fixed = ~1, zi_random = NULL)
f15 = lme.zig(y ~ group + days  + age + gender,
              random = ~ 1 | subject, data = clinical,
              zi_fixed = ~ group, zi_random = NULL)
summary(f14)
```

Analysis of all taxa with a given nonzero proportion with function mms

Similar as mentioned in "Demonstrations of NBMMs and ZINBMMs for directly analyzing longitudinal microbiome/metagenomics count data" and "Demonstrations of ZIGMMs in analyzing longitudinal microbiome/metagenomics count data" sections, for many taxa of interest, we can also analyze their relative abundance proportion data all at one time using the function *mms*. To use ZIGMMs, the term *Method* needs to be set as 'zig' for ZIGMMs. Here we use the following example to show how to analyze the transformed relative abundance proportion data for the first 30 taxa using ZIGMMs. And we used the term *min.p* to control that only the taxa with the proportion of non-zero values > 0.2 to be analyzed.

```
f16 = mms(y = pheno[, 1:30], fixed = ~ group + days  + age + gender,
       random = ~ 1 | subject, min.p = 0.2,
       method = "zig", data = clinical, zi_fixed = ~1,
       zi_random = NULL)
```

Visualize the results

To visualize the results, we have the same three options as mentioned in "Demonstrations of NBMMs and ZINBMMs for directly analyzing longitudinal microbiome/metagenomics count data" section. Figure 5 is generated by *plot.fixed* to view the significant taxa associated with various covariates and corresponding *p* values. We choose only to check the significant taxa especially associated with the variable group shown in Fig. 6. We first updated the object *res* with the function *get.fixed*, setting the term *vr.name* as group. We then plotted Fig. 6 using the function *plot.fixed*. Another option is to generate a heat map as an example shown in Fig. 7 using the function *heat.p*.

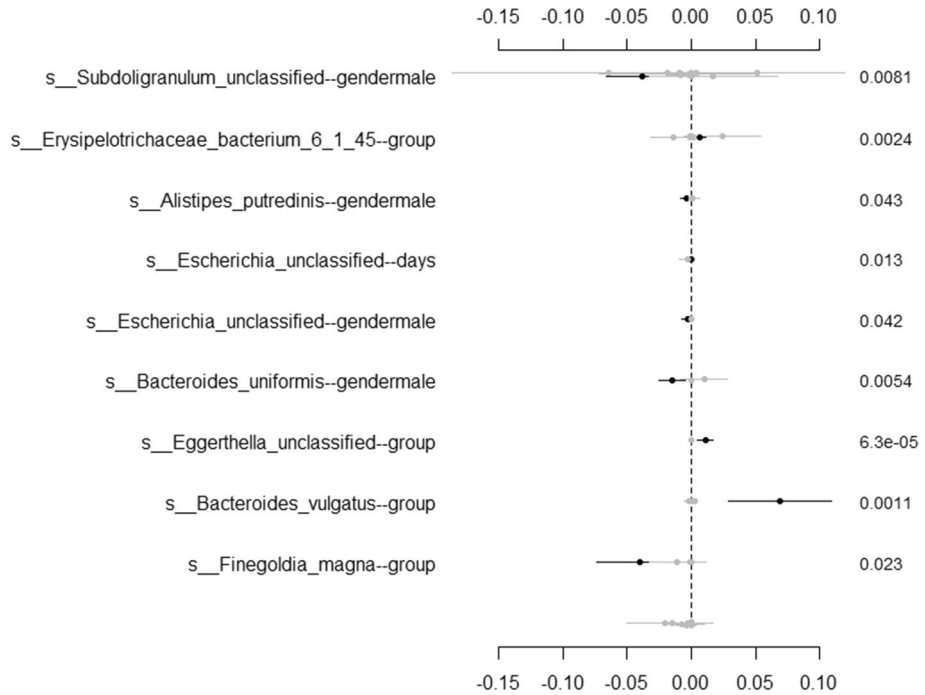

Fig. 5 Effects between significant taxa and all covariates for Vincent et al. [21] using ZIGMMs. The left labels the taxa name and corresponding covariate. The right labels the *p* value

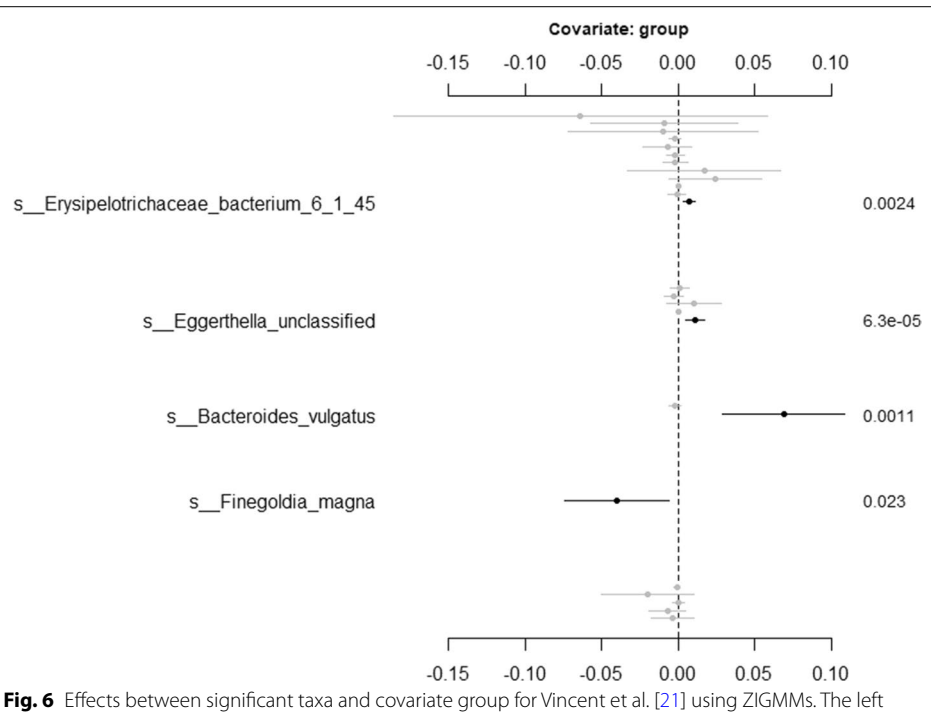

Fig. 6 Effects between significant taxa and covariate group for Vincent et al. [21] using ZIGMMs. The left labels the taxa name and corresponding covariate. The right labels the *p* value

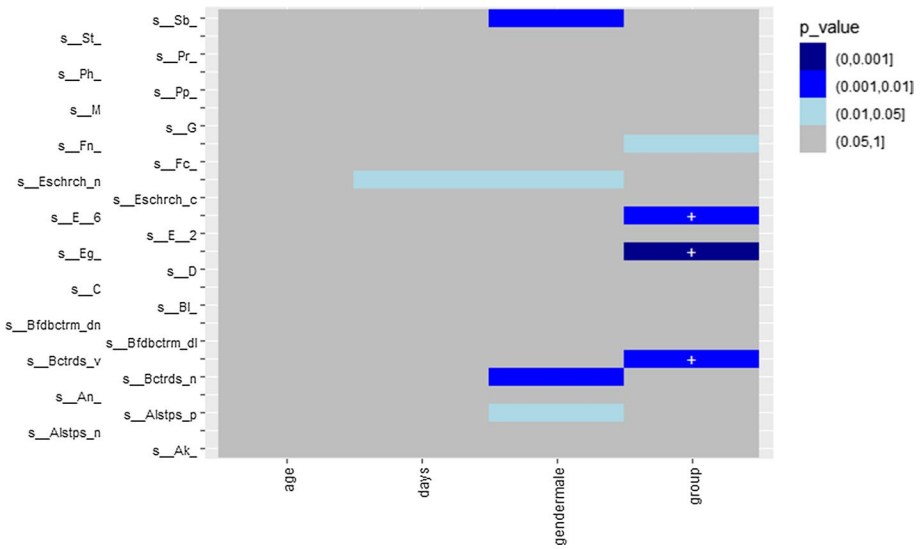

Fig. 7 A heat map for *p* values fitted with ZIGMMs for Vincent et al. [21]. The taxa on y-axis are abbreviated. The sign "+" indicates a positive effect

```
out = fixed(f16)$dist
out = out[out[,2]!="(Intercept)", ]
res = out[, 3:5]

par(mfrow = c(1, 1), cex.axis = 1, mar = c(2, 14, 4, 4))
plot.fixed(res=res, threshold=0.05, gap=500, col.pts=c("black",
"grey"), cex.axis=1, cex.var=1)
```

```
res = get.fixed(f16, part="dist", vr.name="group", sort=F)
par(mfrow = c(1, 1), cex.axis = 1, mar = c(2, 14, 4, 4))
plot.fixed(res, threshold=0.05, gap=300, main="Covariate: group",
cex.axis=1, cex.var=1)
```

```
out = fixed(f)$dist
out = out[out[,2]!="(Intercept)", ]
heat.p(df=out, zigzag=c(T,F), abbrv=c(T,F), margin=c(1.5,10),
y.size=8)
```

Conclusions

We have developed a freely available R package **NBZIMM** to address some of the analytic challenges in complex microbiome/metagenomic studies. Although we emphasize microbiome/metagenomic data, the package and the methods are general and can be used to analyze other over-dispersed and zero-inflated count data with multilevel designs. **NBZIMM** package is under continual development. Our mixed models adopt a classical framework that is not appropriate to jointly analyze multiple correlated covariates. We will extend the mixed models by incorporating weakly informative prior distributions for the fixed effects that allow us to obtain more reliable and stable inferences [23]. We also plan to develop mixed models for jointly analyzing multiple taxa.

Moreover, we found ZIGMMs had inflated false positive rate similarly as the R package **metagenomeSeq** [9]. Our plan for solution is to develop analyzing methods under Bayesian framework using MCMC algorithm to possibly address the current fitting issues.

Availability and requirements

Project name: NBZIMM
Project home page: https://github.com/nyiuab/NBZIMM
Operating system(s): Platform independent
Programming language: R
Other requirements: none
License: MIT
Any restrictions to use by non-academics: none

Abbreviations
CDI: Clostridium difficile infected; EM: Expectation–maximization; EM-IWLS: Expectation–maximization-Iteratively reweighted least squares; NBZIMM: Negative binomial and zero-inflated mixed models; NBMMs: Negative binomial mixed models; OTU: Operational taxonomic units; ZIBR: Zero-inflated Beta regression; ZIGMMs: Zero-inflated Gaussian (linear) mixed models; ZINBMMs: Zero-inflated negative binomial mixed models.

Acknowledgements
None.

Authors' contributions
NY developed the statistical method and the software. XZ performed real data analysis. XZ and NY drafted and revised the manuscript. Both authors read and approved the final manuscript.

Funding
None.

Availability of data and materials
The datasets used and analyzed during the current study are publicly available from Romero et al. [6] and Vincent et al. [21]. The package with manual is freely available from the public GitHub repository https://github.com/nyiuab/NBZIMM.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1] Department of Statistics and Analytical Sciences, Kennesaw State University, Kennesaw, GA, USA. [2] Department of Biostatistics, University of Alabama at Birmingham, Birmingham, AL 35294, USA.

Received: 13 May 2020 Accepted: 8 October 2020
Published online: 30 October 2020

References

1. Gilbert JA, Quinn RA, Debelius J, Xu ZZ, Morton J, Garg N, Jansson JK, Dorrestein PC, Knight R. Microbiome-wide association studies link dynamic microbial consortia to disease. Nature. 2016;535(7610):94–103.
2. Jovel J, Patterson J, Wang W, Hotte N, O'Keefe S, Mitchel T, Perry T, Kao D, Mason AL, Madsen KL, et al. Characterization of the gut microbiome using 16s or shotgun metagenomics. Front Microbiol. 2016;7:459.
3. Ursell LK, Metcalf JL, Parfrey LW, Knight R. Defining the human microbiome. Nutr Rev. 2012;70(Suppl 1):S38-44.
4. Zhang X, Mallick H, Yi N. Zero-inflated negative binomial regression for differential abundance testing in microbiome studies. J Bioinform Genom. 2016;2:2.

5. Zhang X, Mallick H, Tang T, Zhang L, Cui X, Benson AK, Yi N. Negative binomial mixed models for analyzing microbiome count data. BMC Bioinform. 2017;18:4.
6. Romero R, Hassan SS, Gajer P, Tarca AL, Fadrosh DW, Nikita L, Galuppi M, Lamont RF, Chaemsaithong P, Miranda J, et al. The composition and stability of the vaginal microbiota of normal pregnant women is different from that of non-pregnant women. Microbiome. 2014;2(1):4.
7. Pinheiro JC, Bates DC. Mixed-effects models in S and S-PLUS. New York: Springer; 2000.
8. Venables WN, Ripley BD. Modern applied statistics with S. New York: Springer; 2002.
9. Paulson JN, Stine OC, Bravo HC, Pop M. Differential abundance analysis for microbial marker-gene surveys. Nat Methods. 2013;10(12):1200–2.
10. Brooks ME, Kristensen K, van Benthem KJ, Magnusson A, Berg CW, Nielsen A, Skaug HJ, Machler M, Bolker BM. glmmTMB balances speed and flexibility among packages for zero-inflated generalized linear mixed modeling. R J. 2017;9(2):378–400.
11. Rizopoulos D. GLMMadaptive: generalized linear mixed models using adaptive Gaussian quadrature. R package version 06-0 2019.
12. Rigby RA, Stasinopoulos DM. Generalized additive models for location, scale and shape. J R Stat Soc C Appl. 2005;54:507–44.
13. Wood SN, Pya N, Safken B. Smoothing parameter and model selection for general smooth models. J Am Stat Assoc. 2016;111(516):1548–63.
14. Zeileis A, Kleiber C, Jackman S. Regression models for count data in R. J Stat Softw. 2008;27(8):1–25.
15. Chen EZ, Li H. A two-part mixed-effects model for analyzing longitudinal microbiome compositional data. Bioinformatics. 2016;32(17):2611–7.
16. ZIBR (Zero-Inflated Beta Random Effect model). https://github.com/chvlyl/ZIBR.Accessed 23 Oct 2020.
17. Zhang X, Pei YF, Zhang L, Guo B, Pendegraft A, Zhuang W, Yi N. Negative binomial mixed models for analyzing longitudinal microbiome data. Front Microbiol. 2018;9:1683.
18. Zhang X, Yi N. Fast zero-inflated negative binomial mixed modeling approach for analyzing longitudinal metagenomics data. Bioinformatics. 2020;36:2345–51.
19. Lloyd-Price J, Mahurkar A, Rahnavard G, Crabtree J, Orvis J, Hall AB, Brady A, Creasy HH, McCracken C, Giglio MG, et al. Strains, functions and dynamics in the expanded Human Microbiome Project. Nature. 2017;550(7674):61–6.
20. Integrative HMPRNC. The Integrative Human Microbiome Project: dynamic analysis of microbiome-host omics profiles during periods of human health and disease. Cell Host Microbe. 2014;16(3):276–89.
21. Vincent C, Miller MA, Edens TJ, Mehrotra S, Dewar K, Manges AR. Bloom and bust: intestinal microbiota dynamics in response to hospital exposures and Clostridium difficile colonization or infection. Microbiome. 2016;4:12.
22. Pasolli E, Schiffer L, Manghi P, Renson A, Obenchain V, Truong DT, Beghini F, Malik F, Ramos M, Dowd JB, et al. Accessible, curated metagenomic data through ExperimentHub. Nat Methods. 2017;14(11):1023–4.
23. Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB. Bayesian data analysis. 3rd ed. New York: Chapman & Hall; 2014.

Publisher's Note