SOFTWARE TOOL ARTICLE

REVISED *microbiomeDASim*: Simulating longitudinal differential abundance for microbiome data [version 2; peer review: 2 approved]

Justin Williams [1,2], Hector Corrada Bravo[3], Jennifer Tom[1*], Joseph Nathaniel Paulson[1*]

[1]Department of Biostatistics, Genentech, Inc, South San Francisco, CA, 94080, USA
[2]Department of Biostatistics, University of California, Los Angeles, Los Angeles, CA, 90095, USA
[3]Department of Computer Science, University of Maryland, College Park, College Park, MD, 24072, USA

[*] Equal contributors

v2 **First published:** 17 Oct 2019, **8**:1769 (https://doi.org/10.12688/f1000research.20660.1)
Latest published: 26 Feb 2020, **8**:1769 (https://doi.org/10.12688/f1000research.20660.2)

Abstract

An increasing emphasis on understanding the dynamics of microbial communities in various settings has led to the proliferation of longitudinal metagenomic sampling studies. Data from whole metagenomic shotgun sequencing and marker-gene survey studies have characteristics that drive novel statistical methodological development for estimating time intervals of differential abundance. In designing a study and the frequency of collection prior to a study, one may wish to model the ability to detect an effect, e.g., there may be issues with respect to cost, ease of access, etc. Additionally, while every study is unique, it is possible that in certain scenarios one statistical framework may be more appropriate than another. Here, we present a simulation paradigm implemented in the R Bioconductor software package microbiomeDASim available at http://bioconductor.org/packages/microbiomeDASim microbiomeDASim. microbiomeDASim allows investigators to simulate longitudinal differential abundant microbiome features with a variety of known functional forms with flexible parameters to control desired signal-to-noise ratio. We present metrics of success results on one particular method called metaSplines.

Keywords

Microbiome, Differential Abundance, Longitudinal, R, Bioconductor

This article is included in the Bioconductor gateway.

Open Peer Review

Reviewer Status ✓ ✓

	Invited Reviewers	
	1	**2**
version 2 (revision) 26 Feb 2020	✓ report	
version 1 17 Oct 2019	? report	✓ report

1 **Leo Lahti**, University of Turku, Turku, Finland

2 **Kris Sankaran**, Montreal Institute for Learning Algorithms (MILA), Montreal, Canada

Any reports and responses or comments on the article can be found at the end of the article.

Corresponding authors: Jennifer Tom (tom.jennifer@gene.com), Joseph Nathaniel Paulson (paulson.joseph@gene.com)

Author roles: Williams J: Data Curation, Formal Analysis, Investigation, Software, Writing – Original Draft Preparation; **Bravo HC**: Investigation, Methodology; **Tom J**: Conceptualization, Investigation, Methodology, Resources, Supervision, Writing – Original Draft Preparation; **Paulson JN**: Conceptualization, Investigation, Methodology, Resources, Supervision, Writing – Original Draft Preparation

Competing interests: JW, JT, and JNP were employed by Genentech, Inc. during the time of this study. JT and JNP have ownership of stock in F. Hoffmann-La Roche Ltd.

Grant information: The author(s) declared that no grants were involved in supporting this work.

How to cite this article: Williams J, Bravo HC, Tom J and Paulson JN. ***microbiomeDASim*: Simulating longitudinal differential abundance for microbiome data [version 2; peer review: 2 approved]** F1000Research 2020, **8**:1769 (https://doi.org/10.12688/f1000research.20660.2)

First published: 17 Oct 2019, **8**:1769 (https://doi.org/10.12688/f1000research.20660.1)

REVISED **Amendments from Version 1**

We have made revisions to the following manuscript and Bioconductor package based on reviewer feedback. Major changes to the manuscript include an additional section "Approximating Observed Microbiome Data" that highlights using the simulator to generate data from a historical microbiome clinical trial along with reproducible code and figures, additional rationale for using the simulator tools in study design, and updated code within the manuscript. Updates to the package software require users to specify an interval of time for simulating longitudinal data with time points sampled uniformly or randomly, additional functions to match observed data, and the ability to convert simulated data into commonly used objects in the metagenomeSeq and phyloseq packages. Individual responses to reviewer comments are available in the Reviewer Report tab.

Any further responses from the reviewers can be found at the end of the article

Introduction

Analysis of the microbiome aims to characterize the composition and functional potential of microbes in a particular ecosystem. Recent studies have shown the gut microbiome plays an important role in various diseases, from the efficacy of cancer immunotherapy to the pathogenesis of inflammatory bowel disease (IBD)[1–4]. While many studies profile static community "snapshots", microbial communities do not exist within an equilibrium[5]. To better understand bacterial population dynamics, many studies are expanding to longitudinal sampling and foregoing cross-sectional or single time-point explorations. With a decrease in sequencing costs, more longitudinal data will be generated for varying communities of interest. While data generation will present fewer difficulties, there remain several statistical challenges involved in analyzing these datasets.

The common approach in the marker-gene survey literature is to perform pairwise differential abundance tests between specific time points and visually confirm, sometimes using smoothing methods like splines, how differences are manifested across time[6]. These methods require that analysts provide one or more specific time points to test, and the statistical inferences derived from these procedures are specific to these pairwise tests. Other standard methods for longitudinal analysis test for global differences across time, sometimes using non-linear methods including splines to capture dynamic profiles across time[7]. Incorporating confounding sources of variability, both biological and technical is essential in high-throughput studies[8] and require statistical methods capable of estimating both smooth functions and sample-specific characteristics.

Simulating marker-gene amplicon sequencing data presents a variety of challenges related to biological and technical limitations when collecting data. We present a framework for simulating data that can be used across multiple methods for estimating longitudinal differential abundance. This simulation framework allows for appropriate comparison between methods while taking into account some of the unique challenges for the marker-gene amplicon sequencing data, including the following:

1. Non-negative restriction
2. Presence of Missing Data/High Number of Zero Reads
3. Low Number of Repeated Measurements
4. Asynchronous Repeated Measures
5. Small Number of Subjects

The first two challenges described above are related to the data generating process itself while the following three represent logistical challenges often faced when collecting the data. In `microbiomeDASim`[9], we attempt to address these data generating challenges through specific simulation mechanisms described in the *Microbiome adaptions* section. Similarly, logistical challenges are addressed by allowing users to specify these values flexibly and investigate the corresponding effects, tailoring the simulation to an appropriate setting.

This package allows investigators to simulate longitudinal differentially abundant microbiome features with a variety of known functional forms along with flexible parameters to control design aspects such as signal to noise ratio, correlation structure, and effect size. This feature simulation paradigm can be used in study design evaluation by either matching previously observed trends from small scale studies or evaluating the power to detect differential abundance with a specified study duration, sample size, effect size, effect shape and sample collection schedule. We highlight the ability of the package to use results from a historical longitudinal study on the human gut microbiome in gnotobiotic mice[10] to simulate differential abundance for a hypothetical large

scale expansion of this study and then demonstrate using the simulation package to evaluate the performance of one particular method of differential abundance estimation across a range of parameter values, metaSplines[11].

Methods

Distributional assumptions

Sequencing data are often non-normal. However, transformations, such as log(·) or arcsinh(·), are often applied to raw marker-gene amplicon sequencing data so that the subsequent data is approximately normally distributed. As such, we generate simulated data from a multivariate normal distribution. Using a multivariate normal is a natural choice in this setting as longitudinal correlation structure can be easily incorporated. The following methods focus on cases where the desired microbiome features following appropriate transformation are approximately normally distributed.

Assume that we have data generated from the following distribution,

$$\mathbf{Y} \sim N\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right),$$

where

$$\mathbf{Y} = \begin{pmatrix} \mathbf{Y}_1^T \\ \mathbf{Y}_2^T \\ \vdots \\ \mathbf{Y}_n^T \end{pmatrix} = \begin{pmatrix} Y_{11} \\ Y_{12} \\ \vdots \\ Y_{1q_1} \\ Y_{21} \\ \vdots \\ Y_{2q_2} \\ \vdots \\ Y_{nq_n} \end{pmatrix},$$

with Y_{ij} representing the i^{th} individual at the j^{th} time point and each individual has q_i repeated measurements with $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, q_i\}$. We define the total number of observations as $N = \sum_{i=1}^{n} q_i$. While this model holds for different choices of q_i, throughout this article we will assume, without loss of generality, that the number of repeated measurements is constant, i.e., $q_i = q \; \forall \; i \in \{1, \ldots, n\}$. This means that the total number of observations simplifies to the expression $N = qn$. Similarly, we split the total patients (n) into two groups, control (n_0) and treatment (n_1), with the first n_0 patients representing the control patients and the remaining $n–n_0$ representing the treatment patients. Subsequently we define the total number of observations in each group as $N_0 = n_0 \cdot q$ and $N_1 = n_1 \cdot q$ respectively. $\mathbf{Y}$ represents a single taxa/feature to be simulated across the N samples. When simulating multiple features as shown later in the `gen_norm_microbiome`, these features are assumed to be independent.

Mean components

Partitioning our observations into control and treatment groups in this way allows us to define the mean vector separately for each group as $\boldsymbol{\mu} = (\boldsymbol{\mu}_0, \boldsymbol{\mu}_1)$ where $\boldsymbol{\mu}_0$ is an $N_0 \times 1$ vector and $\boldsymbol{\mu}_1$ is an $N_1 \times 1$ vector. To generate differential abundance the mean for the control group is held constant $\mu_0 \; \mathbf{1}_{n_0 \times 1,}$ but allow the mean vector for the treatment group to vary as a function of time $\mu_{1ij}(t) = \mu_0 + f(t_j)$ for $i = 1, \ldots, n_1$ and $j = 1, \ldots, q$. The form of $f(t_j)$ will dictate the functional form of the differential abundance. Note that if $f(t_1) = 0$, then both groups have equal mean at baseline.

Polynomial functional forms

We allow $f(t_j)$ to be specified using polynomial basis as

$$f\left(t_j\right) = \beta_0 + \beta_1 t_j + \beta_2 t_j^2 + \cdots + \beta_p t_j^p$$

for a p dimensional polynomial. We restrict the allowed polynomials to be either linear, p=1, quadratic, $p = 2$, or cubic, $p = 3$. For instance, to define a quadratic polynomial one would specify $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)^T$ in the following equation,

$$f\left(t_j\right) = \beta_0 + \beta_1 t_j + \beta_2 t_j^2.$$

Again, it is important to note that if $\boldsymbol{\beta} = \mathbf{0}$, that the treatment group is assumed to have no differentially abundant timepoints. Typically to simulate no differential abundance, a linear trend is chosen with $\beta_0 = \beta_1 = 0$.

Oscillating functional forms

While polynomial functions are often natural choices for longitudinal trends, interest also lies in exploring other non-smooth, i.e., non-differentiable, types of trends. One such form we refer to as oscillating functional forms. These trends include types that transition from linearly increasing to linear decreasing at a point, or vice versa from linearly decreasing to linear increasing. One of the most well known trends of this type is the absolute value function. To allow for flexible choices in oscillating type trends, we allow for these non differentiable linearly connected trends to repeat forming what we call M and W trends. From a biological perspective we could think of these trends as representing spikes in a particular feature that may occur immediately after a treatment dose is given, but then decays rapidly to baseline levels followed by a similar spike and decay upon repeated dosing. These functional trends are operationalized as

$$\begin{aligned} f\left(t_j\right) = {} & \beta_0 + \beta_1 I\left(t_j < \mathrm{IP}_1\right) t_j + \left(\beta_0 + \beta_1 \mathrm{IP}_1\right) I\left(\mathrm{IP}_1 \le t_j < \mathrm{IP}_2\right) + \left(\beta_0 + \beta_1 \mathrm{IP}_1\right) I\left(t_j \ge \mathrm{IP}_3\right) \\ & + \frac{\left(-\beta_0 - \beta_1 \mathrm{IP}_1\right)}{\mathrm{IP}_2 - \mathrm{IP}_1} I\left(\mathrm{IP}_1 \le t_j < \mathrm{IP}_2\right)\left(t_j - \mathrm{IP}_1\right) \\ & + \frac{\left(\beta_0 + \beta_1 \mathrm{IP}_1\right)}{\mathrm{IP}_3 - \mathrm{IP}_2} I\left(\mathrm{IP}_2 \le t_3 < \mathrm{IP}_2\right)\left(t_j - \mathrm{IP}_2\right) \\ & + \frac{\left(-\beta_0 - \beta_1 \mathrm{IP}_1\right)}{t_q - \mathrm{IP}_3} I\left(t_j \ge \mathrm{IP}_3\right)\left(t_j - \mathrm{IP}_3\right), \end{aligned}$$

where IP_k for $k = 1, 2, 3$ denotes an inflection point where the linear trend changes from increasing to decreasing or vice versa. Note that for these types of trends that the sign of β_1 determines whether the trend is initially increasing, i.e. M, ($\beta_1 > 0$) or initially decreasing, i.e. W, ($\beta_1 < 0$). By construction, we force the trend line to be exactly zero at IP_2 and by doing so the trend is specified completely as $\boldsymbol{\beta} = (\beta_0, \beta_1)^T$ and $\mathbf{IP} = (\mathrm{IP}_1, \mathrm{IP}_2, \mathrm{IP}_3)^T$. An implicit restriction on the functional trend is that $\mathrm{IP}_3 \neq t_q$. However, we can construct absolute value and inverted absolute value type trends by defining $\mathrm{IP}_1 \in (t_1, t_q)$ and $\mathrm{IP}_2, \mathrm{IP}_3 > t_q$. Again, the key difference for these set of trends is that the inflection points create non-smooth trends.

Hockey stick functional forms

An additional extension to linear functional trends is the family of Hockey Stick functional forms. There are two available families of hockey stick functional forms, which are referred to as L_up and L_down within the package. Both of these trends are designed to create two mutually exclusive regions over the time frame specified. These two regions are defined as $\mathscr{R}_1 = (t_1, \mathrm{IP})$ and $\mathscr{R}_2 = (\mathrm{IP}, t_q)$ where one of the regions $\mathscr{R}_1$ or $\mathscr{R}_2$ has linear differential abundance while the other has no differential abundance and IP denotes the inflection point. In the case of the L_up trend, $\mathscr{R}_1$ is defined as the non-differentially abundant region and $\mathscr{R}_2$ is a linearly increasing region. We can define the functional form as

$$f\left(t_j\right) = \left(-\beta_1 \times \mathrm{IP}\right) I\left(t_j \ge \mathrm{IP}\right) + \beta_1 I\left(t_j \ge \mathrm{IP}\right) t_j$$

Note that with this specification that we do not specify the intercept β_0 and instead only need to specify the slope term β_1 and the appropriate point of change. We restrict the slope term to be positive, i.e., $\beta_1 \in (0, \infty)$ to create the "up" trend.

Conversely, the L_down trend assumes that $\mathscr{R}_1$ is a differentially abundant region that begins with the treatment group higher than the control group and then linearly decreases to the region $\mathscr{R}_2$ where there is no differential abundance. We define this functional form as

$$f\left(t_j\right) = \beta_0 I\left(t_j < \frac{-\beta_0}{\beta_1}\right) + \beta_1 I\left(t_j < \frac{-\beta_0}{\beta_1}\right) t_j$$

Note that in this case we do not specify the point of change directly, but rather it is implicitly implied by the choice of β_0 and β_1, i.e. $\mathrm{IP} = -\beta_0/\beta_1$. To ensure that the trend in $\mathscr{R}_1$ is properly specified, we place additional restrictions on the parameters so that $\beta_0 \in (0, \infty)$ and $\beta_1 \in (-\infty, 0)$ to ensure the trend is decreasing and check that the choice of β_0 and β_1 are appropriately defined so that $\mathrm{IP} \in (t_1, t_q)$.

Example trends are shown in Figure 1 generated using the `mean_trend` function.

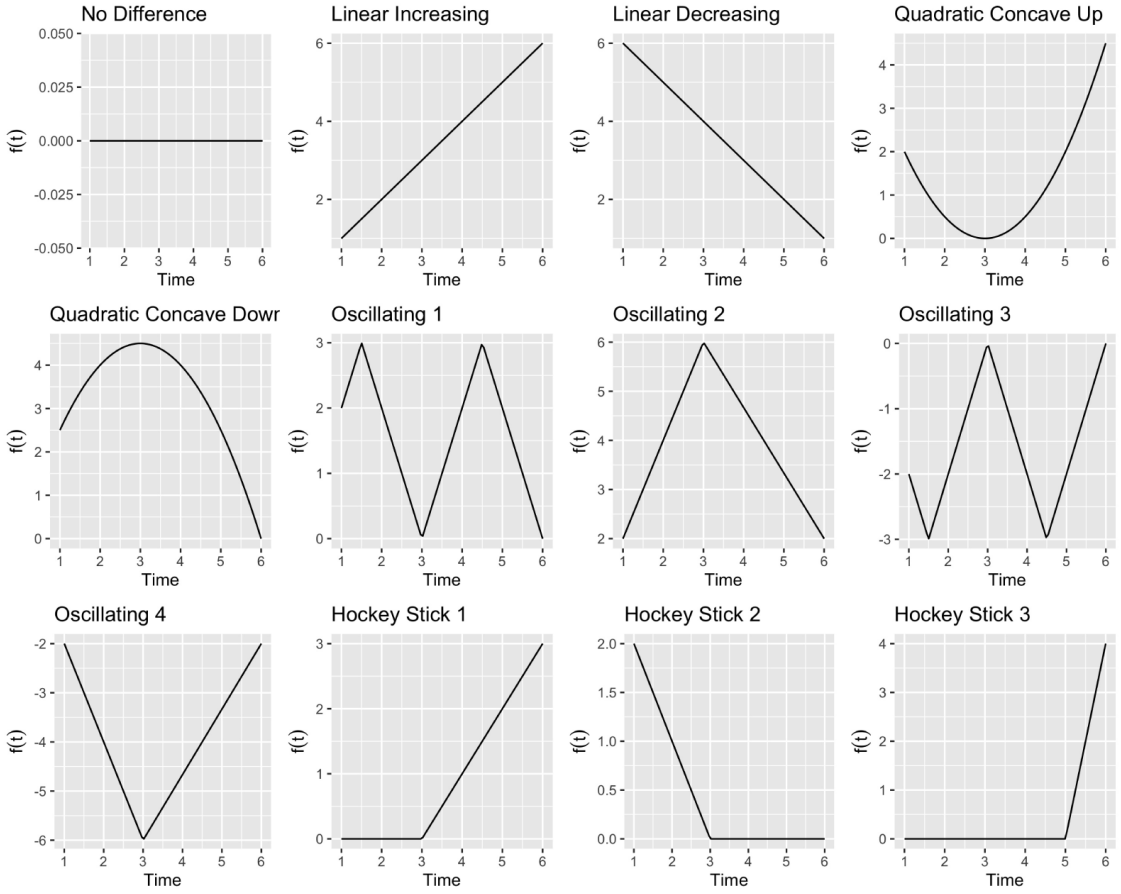


Figure 1. Different functional forms available using the `mean_trend()` function.

Covariance components

As discussed in the *Introduction*, the multivariate normal is a natural choice for longitudinal simulation due to the ease with which dependency of repeated measures is specified. To encode this longitudinal dependency observations within an individual are assumed to be correlated, i.e. $\text{Cor}(Y_{ij}, Y_{ij'}) \neq 0 \ \forall j \neq j'$ and $i \in \{1, \ldots, n\}$, but observations between individuals are assumed independent, i.e. $\text{Cor}(Y_{ij}, Y_{i'j'}) = 0 \ \forall i \neq i'$ and $j \in \{1, \ldots, q_i\}$. To accomplish this we define the block diagonal matrix $\boldsymbol{\Sigma}$ as $\boldsymbol{\Sigma} = \text{bdiag}(\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_n)$, where each Σ_i is a $q \times q$ covariance matrix for individual i and bdiag$(\cdot)$ indicates that the matrix is block diagonal with all off diagonal elements not in $\boldsymbol{\Sigma}_i$ equal to zero. For each individuals covariance matrix, we assume a global standard deviation parameter and correlation component ρ, i.e. $\boldsymbol{\Sigma}_i = \sigma^2\boldsymbol{\Omega}(\rho)$.

For instance, if we want to specify an autoregressive correlation structure for individual i the covariance matrix is defined as

$$\Sigma_i = \sigma^2 \begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{|1-q|} \\ \rho & 1 & \rho & \cdots & \rho^{|2-q|} \\ \rho^2 & \rho & 1 & & \vdots \\ \vdots & & & \ddots & \vdots \\ \rho^{|q-1|} & \rho^{|q-2|} & \cdots & \cdots & 1 \end{bmatrix}$$

In this case we are using the first order autoregressive definition and therefore will refer to this as AR(1).

Alternatively, for the compound correlation structure for an individual i' we define the covariance matrix as

$$\Sigma_{i'} = \sigma^2 \begin{bmatrix} 1 & \rho & \rho & \cdots & \rho \\ \rho & 1 & \rho & \cdots & \rho \\ \rho & \rho & 1 & & \vdots \\ \vdots & & & \ddots & \vdots \\ \rho & \rho & \cdots & \cdots & 1 \end{bmatrix}$$

Finally, we allow the user to specify an independent correlation structure for an individual i'', which assumes that repeated observations are in fact uncorrelated and is defined as

$$\Sigma_{i''} = \sigma^2 \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & & \vdots \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 1 \end{bmatrix}$$

Each of these correlation structures are referred as AR(1), compound, and independent respectively.

Microbiome adaptions

As discussed in the *Introduction*, simulating microbiome data presents a variety of unique challenges. In particular there are two data generating restrictions, 1. non-negative restriction and 2. presence of missing data/high number of zero reads, that must be addressed when simulating this data. In this section we will outline some of the specific adaptions of the simulation framework designed to address these issues.

1. Non-negative restriction. One of the most relevant challenges faced with microbiome data, is the restriction of the domain to non-negative values. To assure that the simulated normalized counts are non-negative, one solution is to simply replace the multivariate normal distribution with a multivariate *truncated* normal distribution. The new data generating distribution is now

$$\mathbf{Y} \sim \text{TN}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, a\mathbf{1}_N),$$

where TN indicates the multivariate truncated normal distribution and a is the left-truncation value. To impose zero truncation it is assumed that $a = 0$. Values from the multivariate truncated normal are drawn using the package `tmvtnorm`[12]. Note that the default method for drawing observations from this distribution is rejection sampling which proceeds by first drawing from a multivariate normal and then for all values that fall below a to reject the observed sample and re-sample. This procedure works well when the majority of the distribution falls above the truncation point, but can be computational intensive when the probability of acceptance, $p_{\text{acpt}} = P(\mathbf{Y} > a\mathbf{1}_N)$, is low. In our simulation design if the value of $\boldsymbol{\mu}$ is sufficiently close to a then rejection sampling is not feasible. In the case there the $p_{\text{acpt}} \leq 0.1$, non-negative restriction is imposed by censoring negative values and using point imputation with the truncation value a as shown below

$$\mathbf{Y}^* \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}),$$
$$Y_{ij} = \begin{cases} Y_{ij}^* & \text{if } Y_{ij}^* \geq 0, \\ 0 & \text{if } Y_{ij}^* < 0. \end{cases}$$

To remove the non-negative restriction there is an option in the function `mvrnorm_sim` which can be used to turn-off the domain restriction, but by default the zero truncation is imposed. Note that an alternative option to using the multivariate truncated normal is to use the Johnson translation system which can allow samples to be drawn from a multivariate log normal distribution via an appropriate translation function[13]. The current implementation uses only the multivariate truncated normal distribution for drawing samples via the `zero_trunc` option within the `mvrnorm_sim()` and `gen_norm_microbiome()` functions.

2. Presence of missing data/high number of zero reads. The second major data generating challenge when simulating microbiome data is the presence of missing data along with a high percentage of features with zero counts. Based on technical limitations when amplifying and sequencing microbiome data, certain features may be present but remain undetected. To approximate this potential for missing features that are truly present, options within `mvrnorm_sim` allow the user to specify: 1) the percent of individuals to generate missing values from (`missing_pct`), 2) the number of measurements per individual to assign as missing (`missing_per_subject`), and 3) the value to impute for missing observations (`miss_val`). Sample IDs are randomly chosen without replacement across all n units and for each selected ID measurements are randomly selected without replacement from $\{t_2, \ldots, t_q\}$ until the specified number of measurements per individual is achieved. For each missing measurement selected the observed value is replaced with the user specified missing value. Typically the missing value is specified as 0 or as *NA* with the first case representing a situation where the feature was not included due to technical limitations and the second representing an individual whose data was not collected for a particular time point. The initial value t_1 cannot be assigned as missing since it is assumed that all individuals have baseline values collected.

Implementation

The current version of the R Bioconductor software package `microbiomeDASim`[9] can be installed in R with the following executable code:

```
if(!requireNamespace("BiocManager", quietly = TRUE)){
   install.packages("BiocManager")
}
BiocManager::install("microbiomeDASim")
```

Alternatively, a development version is available from GitHub and can be accessed at the following repository williazo/microbiomeDASim. The developmental version may contain additional features that are being developed before they are officially introduced into the Biocondutor version. The developmental version can be installed using the following code:

```
if(!requireNamespace("devtools", quietly = TRUE)){
   install.packages("devtools")
}
devtools::install_github("williazo/microbiomeDASim")
```

For a guided introduction into using the functions see either the package vignette for a static example of how to set up and interact with various options for simulating data or for a dynamic guide see mvrnorm_demo.ipynb, a Jupyter notebook on the GitHub page under the inst/script directory. This notebook can be loaded using Google Collab allowing the code to be run without installing Jupyter locally.

Operation

`microbiomeDASim`[9] is compatible with major operating systems including Mac OS, Windows and Linux. Package dependencies and system requirements are outlined in the documentation available at GitHub.

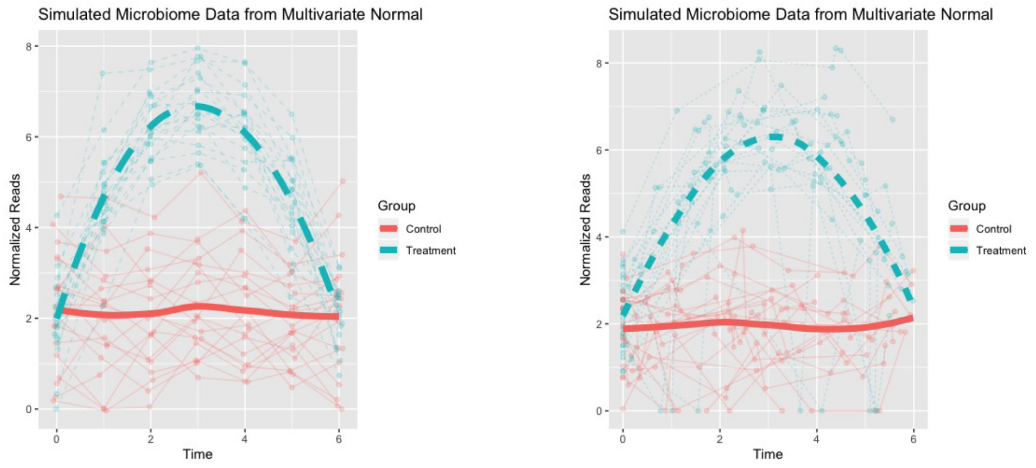
Use cases

Data generating procedure

The primary mechanism for simulating data in the `microbiomeDASim` package[9] is the function `mvrnorm_sim`. Through this function, the number of subjects in each group is specified along with the necessary parameters, i.e $\boldsymbol{\beta}$, σ^2, ρ, and $\mathbf{IP}$, to generate $\boldsymbol{\mu}$ and Σ. Below is an example of generating differential abundance using a quadratic trend. This type of example could be part of an initial attempt to understand the effects of proposed sample sizes per group, hypothetical functional forms for differential abundance, and sensitivity to signal to noise ratios. In this case there may be a sparsity of empirical evidence and many possible simulation designs can be tested, or on the other end of the spectrum the ecological process could be well understood and the parameter values are well known with emphasis focused on constraints such as collection timepoints and sample size.

```
> library(microbiomeDASim)
> sim_dt <- mvrnorm_sim(n_control=20, n_treat=20, control_mean=2, sigma=1,
+                       num_timepoints=7, t_interval=c(0, 6), rho=0.7,
+                       corr_str="compound", func_form="quadratic",
+                       beta=c(0, 3, -0.5), missing_pct=0, missing_per_subject=0,
+                       asynch_time=FALSE, dis_plot=TRUE)
> typeof(sim_dt)
[1] "list"
> names(sim_dt)
[1] "df"        "Y"         "Mu"        "Sigma"     "N"         "miss_data" "Y_obs"
> head(sim_dt$df)
          Y ID time   group     Y_obs
1 0.2132845  1    0 Control 0.2132845
2 0.7784994  1    1 Control 0.7784994
3 1.6464264  1    2 Control 1.6464264
4 1.6283489  1    3 Control 1.6283489
5 0.8769442  1    4 Control 0.8769442
6 0.7625660  1    5 Control 0.7625660
> head(sim_dt$miss_data)
[1] miss_id
<0 rows> (or 0-length row.names)
```

The output of the simulation function is a list with 7 total objects. The main object of interest is `df`, which is a data.frame that contains the complete outcome, `Y`, IDs for each subject $i = 1, \ldots, n$, the corresponding time for each observation t_j, a group variable indicator, and the outcome with missing data, `Y_obs`. The time interval of interest must be specified as a parameter in `t_interval`, and by default timepoints are drawn at equidistant points along this interval. Both the complete and missing data vectors are also returned as independent objects, `Y` and `Y_obs`, respectively, along with the complete mean, $\boldsymbol{\mu}_{N\times 1}$ = `Mu`, and covariance matrix, $\boldsymbol{\Sigma}$=`Sigma`. The function also includes a data.frame `miss_data` which lists any IDs and time points for which missing data was induced. Finally, the function also returns the total number of observations, `N`=$\Sigma_i\, q_i$. The option dis_plot is used to automatically generate a time-series plot tracking each individuals trajectory along with group mean trajectories. The corresponding plot for this data is shown in Figure 2a.



(a) Quadratic Trend without Missing Data and Equadistant Sampling

(b) Quadratic Trend with Induced Missing Data and Asyhcnronous Time Measurements

Figure 2. Simulating a quadratic differential abundance trend with compound correlation structure and parameters: $\beta = (0, 3, -0.5)^T$, $\rho = 0.7$, $\sigma = 1$, $n_0 = n_1 = 20$, $q = 6$. Missing data in Figure 2b is generated with 20% of subjects randomly selected to have missing values and for each of these subjects to have 2 non-baseline times randomly selected to be missing with the missing observations imputed as 0.

One important thing to note about the example above is that we generated no missing observations as both missing_pct and missing_per_subject were set to 0. Therefore `miss_data` was empty. We can compare this to the case below where we induce missingness into the data.

```
> sim_dt <- mvrnorm_sim(n_control=20, n_treat=20, control_mean=2, sigma=1,
+                       num_timepoints=7, t_interval=c(0, 6), rho=0.7,
+                       corr_str="compound", func_form="quadratic",
+                       beta=c(0, 3, -0.5), missing_pct=0.2,
+                       missing_per_subject=2, miss_val=0, asynch_time=TRUE)
> head(sim_dt$miss_data[order(sim_dt$miss_data$miss_id, sim_dt$miss_data$miss_time),])
   miss_id miss_time
11       6         5
12       6         6
13      13         2
14      13         4
10      16         3
9       16         4
> head(sim_dt$df[sim_dt$df$ID %in% sim_dt$miss_data$miss_id, ])
           Y ID      time group       Y_obs
36 1.7663067  6 0.0000000 Control 1.7663067
37 0.5918298  6 0.1084354 Control 0.5918298
38 2.0162980  6 1.8372196 Control 2.0162980
39 1.7626451  6 1.8900365 Control 1.7626451
40 2.0873529  6 3.2812129 Control 0.0000000
41 2.1775117  6 5.2906868 Control 0.0000000
```

In this case we see that for t_5 and t_6 for subject 6 that our outcome with missing data, Y_obs, is now set as 0 which was specified as our missing value while the complete data has the original value before inducing missingness. Another feature demonstrated in this second example is using the `asynch_time` option. When this variable is set to true, timepoints are randomly drawn from a uniform distribution over the interval $[t_0, t_q]$. By construction it is assumed that all individuals have a baseline measurement recorded at t_0, but all remaining timepoints are drawn at random. The corresponding plot of the outcome `Y_obs` for this simulation which contains the induced missing observations and asynchronous time measurements is shown in Figure 2b.

As mentioned in the *Distributional assumptions* section, data are generally generated one feature at a time. However, we may want to simultaneously create data with similar patterns across a number of features with certain features experiencing differential abundance while others have no differential abundance patterns. To do this we can use the function `gen_norm_microbiome` which lets users specify the number of total features to simulate, `features`, and the number of total features to be differentially abundant, `diff_abun_features`. In the example below 10 total features are generated with 4 features having longitudinal differential abundance with an L_down hockey stick type trend.

```
> bug_gen <- gen_norm_microbiome(features=10, diff_abun_features=4, n_control=20,
+                                n_treat=20, control_mean=2, sigma=1,
+                                num_timepoints=5, t_interval=c(0,10), rho=0.7,
+                                corr_str="compound", func_form="L_down",
+                                beta=c(2, -0.5), missing_pct=0.2,
+                                missing_per_subject=2, miss_val=0)
Simulating Diff Bugs

  |++++++++++++++++++++++++++++++++++++++++++++++++++| 100% elapsed=05s
Simulating No-Diff Bugs
```

```
  |++++++++++++++++++++++++++++++++++++++++++++++++++| 100% elapsed = 07s
> head(bug_gen$bug_feat)
         ID time   group Sample_ID
Sample_1  1  0.0 Control  Sample_1
Sample_2  1  2.5 Control  Sample_2
Sample_3  1  5.0 Control  Sample_3
Sample_4  1  7.0 Control  Sample_4
Sample_5  1 10.0 Control  Sample_5
Sample_6  1  0.0 Control  Sample_6
> bug_gen$Y[, 1:5]
              Sample_1 Sample_2 Sample_3  Sample_4   Sample_5
Diff_Bug1   2.78721292 3.034923 2.448909 3.4472145 2.01708421
Diff_Bug2   2.17420076 2.126378 1.875765 2.1224031 1.45393399
Diff_Bug3   3.00420764 2.667490 2.919144 2.5646103 1.98241611
Diff_Bug4   2.79533312 2.526658 2.254584 3.4089330 3.46269243
NoDiffBug_1 1.91089105 2.000122 1.265382 1.1625345 0.97581881
NoDiffBug_2 3.20731129 3.446508 3.389278 3.1057941 4.21898174
NoDiffBug_3 0.05647967 0.000000 1.553321 0.2595184 0.08792209
NoDiffBug_4 2.08900175 1.566923 1.917273 1.4543443 1.34799811
NoDiffBug_5 0.03105152 2.350758 2.139133 1.5934641 0.58829093
NoDiffBug_6 2.24743076 3.082808 2.526052 1.8868046 2.33309314
```

There are two objects returned in this function, `bug_feat` and `Y`. The object `bug_feat` contains all of the sample specific information including Subject ID, timepoint t_j, an indicator for group assignment and the Sample_ID which ranges from Sample_1 up to Sample_N. The other object `Y` is the typical OTU (operational taxonomic unit) table with rows corresponding to features and column to samples that are commonly used for analysis in packages such as `metagenomeSeq`[14,15] and `phyloseq`[16]. There are two additional helper functions that will convert the simulated data into MRexperiment or phyloseq objects respectively to allow practitioners to use simulated data in either of these familiar environments.

```
> # convert to MRexperiment object
> MR_bug_gen <- simulate2MRexperiment(bug_gen)
> MR_bug_gen
MRexperiment (storageMode: environment)
assayData: 10 features, 200 samples
  element names: counts
protocolData: none
phenoData
  sampleNames: Sample_1 Sample_2 ... Sample_200 (200 total)
  varLabels: ID time group Sample_ID
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation:
> head(pData(MR_bug_gen))
         ID time   group Sample_ID
Sample_1  1  0.0 Control  Sample_1
Sample_2  1  2.5 Control  Sample_2
Sample_3  1  5.0 Control  Sample_3
Sample_4  1  7.5 Control  Sample_4
Sample_5  1 10.0 Control  Sample_5
Sample_6  2  0.0 Control  Sample_6
>
```

```
> # convert to phyloseq object
> phylo_bug_gen <- simulate2phyloseq(bug_gen)
> phylo_bug_gen
phyloseq-class experiment-level object
otu_table()   OTU Table:         [ 10 taxa and 200 samples ]
sample_data() Sample Data:       [ 200 samples by 4 sample variables ]
> head(sample_data(phylo_bug_gen))
Sample Data:        [6 samples by 4 sample variables]:
         ID time   group Sample_ID
Sample_1  1  0.0 Control  Sample_1
Sample_2  1  2.5 Control  Sample_2
Sample_3  1  5.0 Control  Sample_3
Sample_4  1  7.5 Control  Sample_4
Sample_5  1 10.0 Control  Sample_5
Sample_6  2  0.0 Control  Sample_6
```

Approximating observed microbiome data

Another important goal of the simulation software is the ability to closely approximate real data from longitudinal experiments where sequencing was performed. To demonstrate this ability using `microbiomeDASim` we will approximate observed data from a longitudinal study on the human gut microbiome in gnotobiotic mice[10]. This data file is available within the `metagenomeSeq` package, and is particularly interesting to simulate for several reasons. The experiment was performed with a total of 12 mice, 6 in each treatment arm, to test the effect of a low-fat, plan polysaccharide-rich diet (BK) versus a high-fat, high-sugar (Western) diet. This small scale study showed promising results and may warrant a larger scale clinical design to investigate the robustness of the effect of diet on the gut microbiome. As such, we can use the simulation tools to generate hypothetical results for this large scale trial assuming that we observe either the same functional trend as the original study or any of the possible hypothetical functional trends at our disposal, including no differential abundance. We will show how to generate hypothetical data for a large scale version of this experiment by increasing the sample size by five fold and replicating the observed functional trend for a particular feature of interest.

As a first step we need to identify a particular feature of interest at an appropriate taxonomic level. The original data contains sequenced counts on over 10,000 OTUs with the majority of these being extremely low frequency features. Since the total sample size (n=12) is too small for central limit theory approximations to be valid, we aggregate counts to the genus level for modelling. We further filter genus level features by imposing a minimum depth of 1000 and presence of 10, leaving a set of 35 features. Of these 35 features, we select one at random which we will want to replicate using our simulation framework. In our case we select the genus *Sutterella*. The raw sequencing counts are then log normalized using the default procedure available in the `metagenomeSeq` package which will serve as our primary outcome of interest. We plot these results over time as shown in Figure 3.

There is significant variability between the groups with a marked decrease in both groups prior to the implementation of the intervention. We see a bounce back effect to baseline levels occurring in the BK diet group while the Western diet group have significantly lower values across the remainder of the study period. We see that the measurement timepoints for each individual vary slightly and are not equally spaced over the entire study window.

As the primary interest lies in the difference between the diet groups across time, we develop our simulation model by re-scaling the BK reference group to a constant level across time and allowing theWestern group to vary. To obtain an initial estimate of this treatment functional trend we use the `metagenomeSeq`[15] package to fit a Gaussian smoothing spline ANOVA (SS-ANOVA) shown in Figure 4.

We see that over the initial 21 days that the 95% confidence intervals for the differential abundance overlaps zero, and that after the intervention begins that the Western group is significantly lower than the BK group. While the estimated trend is non-linear, we may expect that this is a function of small sample size noise and that the true functional trend is a linearly decreasing trend. We therefore construct our hypothetical functional form using the `L_up` designation assuming there is no differential abundance over the interval $t \in [0, 21]$ followed by a linearly decreasing trend over the interval $t \in (22,80]$. We show this chosen functional form alongside the estimated differential abundance in Figure 5.

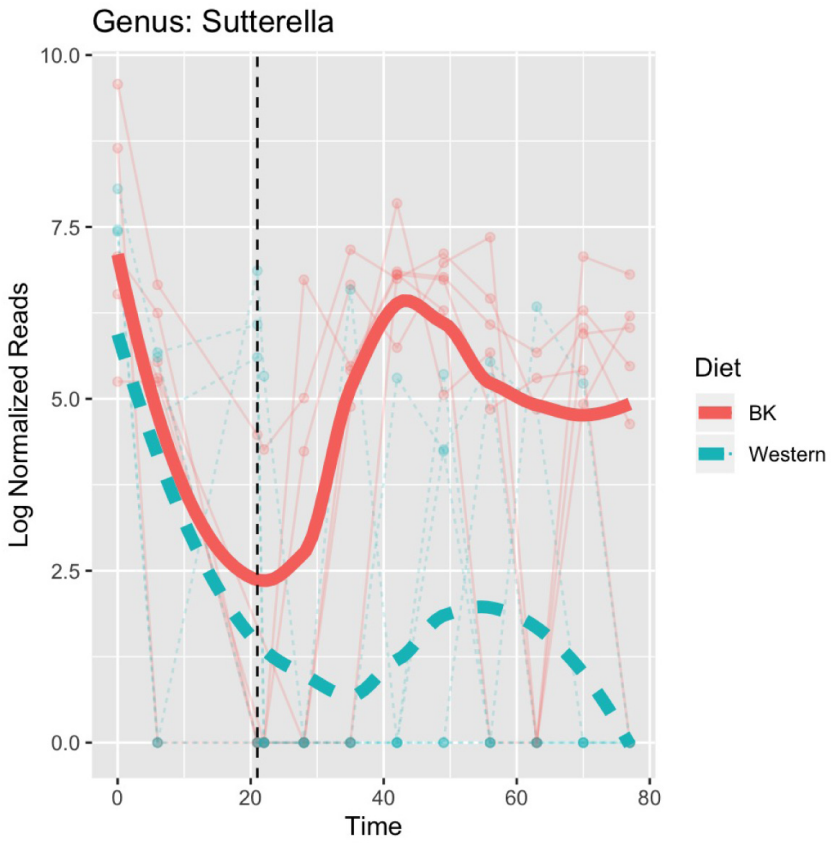


Figure 3. Observed longitudinal trends for the two diet groups in Turnbaugh *et al.*[10] study for the Genus *Sutterella* with estimated LOESS curves for each group. Note that both groups had equivalent diets over the first 21 days with half of the mice switching to the Western diet at this point marked with the vertical dotted line.

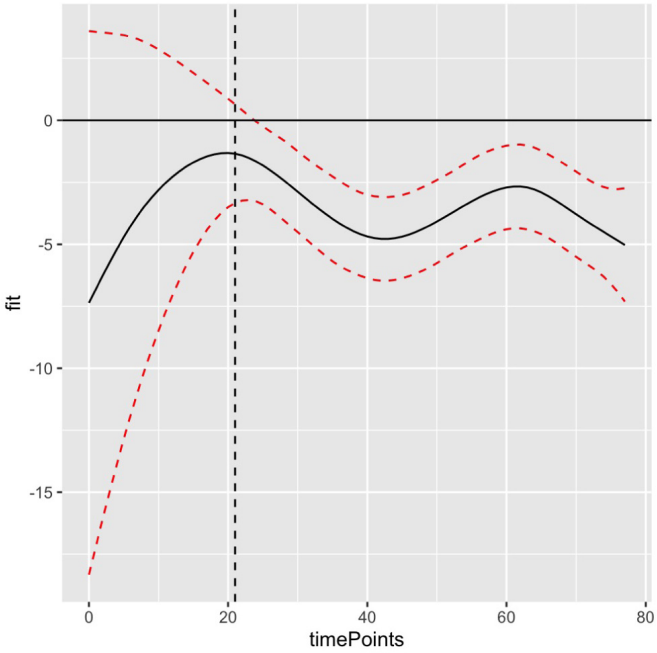


Figure 4. Estimated functional form of the longitudinal differential abundance for the Western diet group from Turnbaugh *et al.*[10] study for the Genus *Sutterella*. The black line represents the point estimate with the dashed red lines corresponding to 95% confidence intervals fit using Gaussian smoothed spline ANOVA.

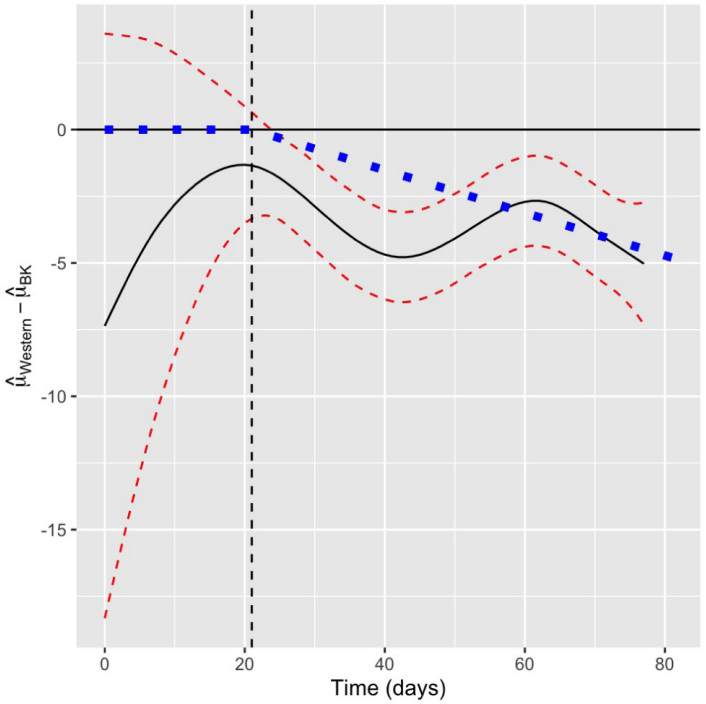


Figure 5. Estimated functional form of the longitudinal differential abundance for the Western diet group from Turnbaugh *et al.*[10] study for the Genus Sutterella with the corresponding functional form chosen for the simulation shown in blue. The black line represents the point estimate with the dashed red lines corresponding to 95% confidence intervals fit using Gaussian smoothed spline ANOVA.

In general our hypothetical trend is contained within the estimated bounds of the smoothed fig, and we may believe that it is an ecologically valid representation of the expected change over time.

With `microbiomeDASim` we can use the observed times for each ID, and replicate each subject five times creating a total sample size of n=60 with 30 mice in each treatment arm. We use the data to obtain estimates for `sigma` and `control_mean` along with the functional form chosen above to generate the simulated data using the `mvrnorm_sim_obs()` function with an AR1 correlation structure. The results for the simulated data are shown in Figure 6.

This simulated data could then be used to conduct power analyses of detecting differential abundance at time $t \in [t_0, t_q]$ or this process could be repeated multiple times to generate feasible bounds for what the trend may look like in this larger sample. Alternatively, the observed data could be altered to change the planned time point measurements to see the effect of collecting fewer samples during the follow-up period. In addition, as mentioned earlier in this section multiple functional forms could be tested including situations where no differential abundance is observed to determine the likelihood of committing Type 1 errors. Further details and code for this example are available on GitHub at inst/script/mouse_microbiome_approximation.pdf

Longitudinal differential abundance estimation

Next, we want to use our simulation design to test some of the available methods to estimate longitudinal differential abundance. We will examine properties of the estimation method available in the `metagenomeSeq`[15] package to fit a Gaussian smoothing spline ANOVA (SS-ANOVA) model[11,17,18] referred

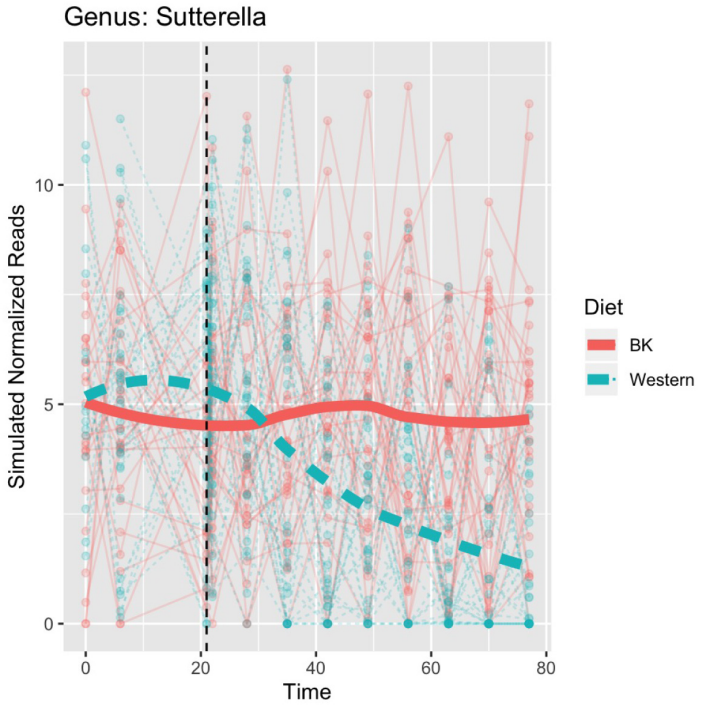


Figure 6. Estimated functional form of the longitudinal differential abundance for the Western diet group from Turnbaugh *et al.*[10] study for the Genus *Sutterella* with the corresponding functional form chosen for the simulation shown in blue. The black line represents the point estimate with the dashed red lines corresponding to 95% confidence intervals using metaSplines to fit smoothed spline ANOVA.

to here after as the metaSplines method. We start by generating our simulated data. In this example we will fix parameters to have $q = 10$ repeated measurements on each individual with $n_0 = n_1 = 30$ individuals per arm.

```
> #generating the simulated data
> out_sim <- mvrnorm_sim(n_control = 30, n_treat = 30, control_mean = 2, sigma = 1,
+                        num_timepoints = 10, t_interval=c(1, 10),
+                        rho = 0.8, corr_str = "compound",
+                        func_form = "L_up", beta = 0.5, missing_pct = 1,
+                        missing_per_subject = 2, IP = 5)
>
> #capturing the true mean values for the specified functional form
> true_mean <- mean_trend(timepoints=seq_len(10), form = "L_up", beta = 0.5, IP = 5)
>
> MR_mvrnorm <- simulate2MRexperiment(out_sim)
> MR_mvrnorm
MRexperiment (storageMode: environment)
assayData: 1 features, 600 samples
  element names: counts
protocolData: none
phenoData
  sampleNames: 1 2 ... 600 (600 total)
  varLabels: ID time group
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
Annotation:
```

After generating the simulated data, we fit the model. Note that one can fit either the outcome with the complete data or the outcome with imputed missing data. In this example we use the complete data. To use the induced missing data when creating the MRexperiment object we would set the missing variable in `simulate2MRexperiment` to TRUE.

```
> #fitting the metaSplines model with random intercept
> metasplines_mod <- fitTimeSeries(obj = MR_mvrnorm, formula = abundance ~ time*class,
+                                  id = "ID", time = "time", class = "group",
+                                  feature = 1, norm = FALSE, log = FALSE, B = 1000,
+                                  random = ~ 1|id)
Loading required namespace: gss
[1] 100
[1] 200
[1] 300
[1] 400
[1] 500
[1] 600
[1] 700
[1] 800
[1] 900
[1] 1000
```

Now we can display the estimated interval of differential abundance

```
> metasplines_mod$timeIntervals
     Interval start Interval end     Area       p.value
[1,]              6           10 6.457622   0.000999001
```

We compare the estimated trend $\hat{f}(t_j)$ to the truth $f(t_j)$ as shown in Figure 7. We observe that the metaSplines estimate falls closely to the true functional form. Further, the confidence intervals for the functional form completely contain the true trend reflecting that the variability in estimation is accurately reflected.

Evaluating estimation procedures

In the example for metaSplines above we looked at performance using a visual inspection for a single choice of parameter values. Using our simulation framework we can expand our investigation of performance. By knowing the true underlying functional form we can quantify how accurate a particular estimation method captures the truth as a function of sample size per group, number of repeated observations, signal-to-noise strength, type of functional form etc. In order to use the simulated data to compare different longitudinal methods for estimating differential abundance we need to define performance metrics that quantify how accurate an estimate is to the truth. We propose four different performance metrics that can be used when comparing methods.

1. Sensitivity/Specificity $\in [0, 1]$
2. Cosine Similarity $\frac{\hat{f}(\mathbf{t})^T f(\mathbf{t})}{\|\hat{f}(\mathbf{t})\| \cdot \|f(\mathbf{t})\|} \in [-1,1]$
3. Euclidean Distance $\|\hat{f}(\mathbf{t}) - f(\mathbf{t})\| \in [0,\infty]$
4. Normalized Euclidean Distance $\|\frac{\hat{f}(\mathbf{t})}{\|\hat{f}(\mathbf{t})\|} - \frac{f(\mathbf{t})}{\|f(\mathbf{t})\|}\| \in [0,2]$

To ensure robustness, for each set of parameter values simulated multiple repetitions, B, are required. Sensitivity is defined as the number of repetitions where **any** differential abundance at any value $t_j \in \{t_1, \ldots, t_q\}$ is detected over the total number of repetitions given that the functional form had some true differential abundance over time, i.e. $f(t_j) \neq 0\ \forall t_j \Leftrightarrow \boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_0$. Likewise, specificity is defined as the number of repetitions where no differential abundance was detected across **all** timepoints over the total number of repetitions given that the function form had no true differential abundance over time, i.e., $f(t_j) = 0\ \forall t_j$. The other remaining metrics are

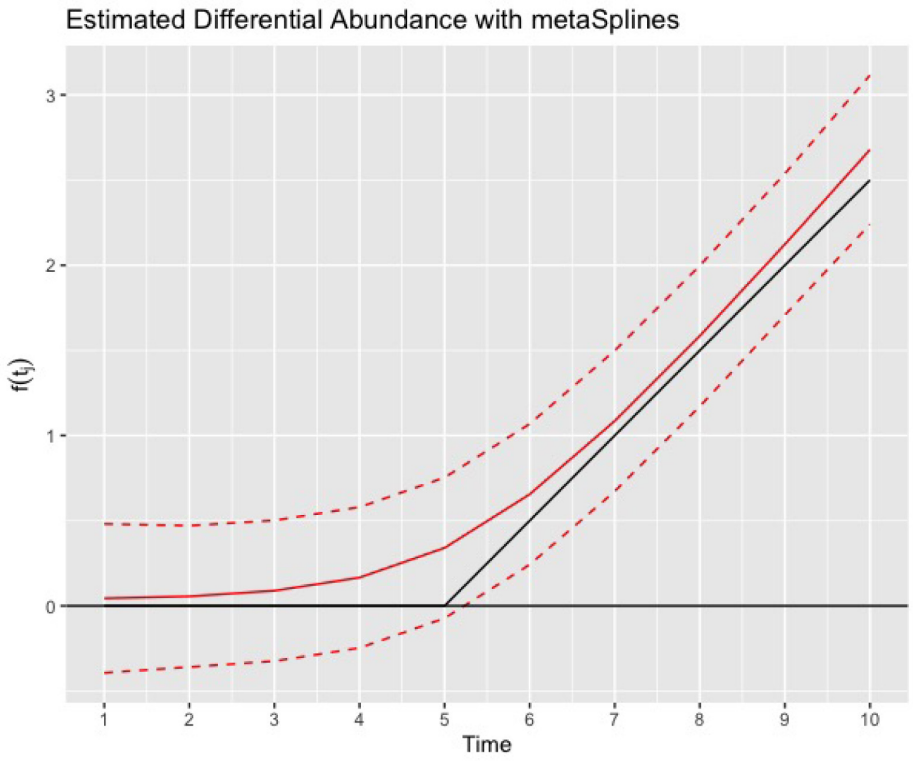


Figure 7. Comparison of the estimated functional form for the metaSplines method, in red, to the truth, in black.

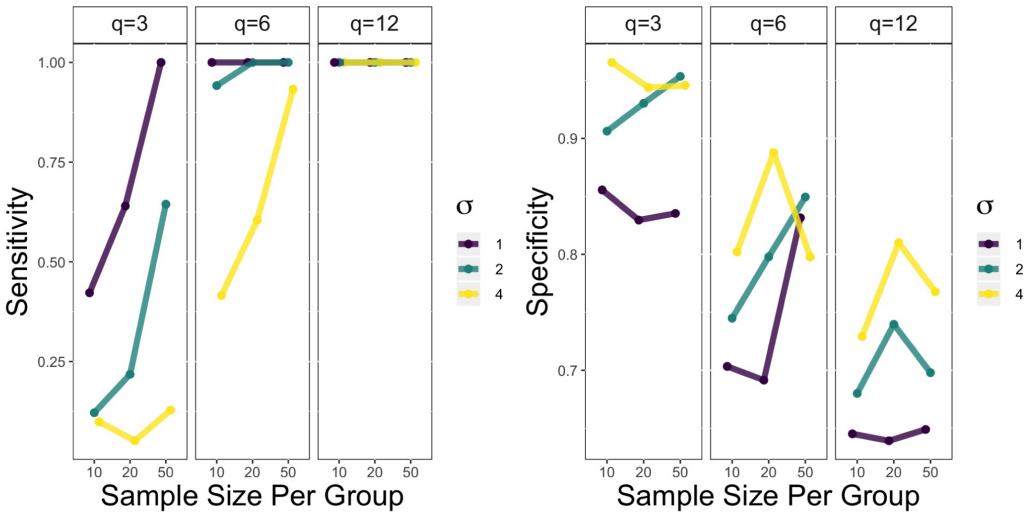
continuous values that look to compare how closely the estimated mean trend is to the true trend at a set of points $t_j \in \{t_1, \ldots, t_q\}$. Cosine similarity is comparable across different lengths of **t**, but is not particularly discriminant especially near the boundaries around –1 and 1. The Euclidean distance quantifies how far apart each point is but the length of **t** is highly influential. Therefore, to make the Euclidean distance comparable across different lengths of repeated observations we can use the normalized Euclidean distance which first transforms the estimated and true functional form into unit vectors and then calculates the distance between these unit vectors.

Sensitivity and specificity results

Using these performance metrics we simulated data across a range of different parameters settings and then estimated the functional form of the trend using the metaSplines procedure described earlier for a total of 100 repetitions for each parameter setting. Below we show the performance results for a simulation where the functional form was fixed as L_up with an AR(1) correlation structure, $\rho = 0.7$, and varied the sample size per group, standard deviation, and timepoints from small, medium, and large respectively. The corresponding sensitivity and specificity results are shown in Figure 8a and Figure 8b.

Looking at Figure 8a, in general the sensitivity decreases as σ increases for a fixed sample size and q. For example when $n_0 = n_1 = 10$ and $q = 6$ the estimation procedure is perfectly sensitive (100%) when $\sigma = 1$ but has lower sensitivity (42%) when $\sigma = 4$. Also as the sample sizes increases for a fixed q and σ, sensitivity generally increases. Likewise, as the number of repeated observations increase, i.e. q increases, the sensitivity increases quite dramatically. This figure suggests that 6 repeated measurements is sufficiently large to detect differential abundance for strong ($\sigma = 1$) or medium ($\sigma = 2$) signals regardless of the sample size per group. On the other hand, we can look at the specificity in Figure 8b to see that these trends are no longer monotonic. In general we note that as q increases the specificity decreases and that as σ increases the specificity tends to increase. However, the trend for sample size is more nuanced and may variable due to the number of repetitions that were estimable. Using the metaSplines method there were cases with small sample size and repeated observations that the method returned no estimate.

The sensitivity results shown above were for a single choice of functional form, but this is another potential parameter of interest to test. We ran a similar set of parameter combinations for 7 other functional forms shown in Table 1 below to compare the sensitivity as a function of the type of trend. In this table we can see that the



(a) Sensitivity of metaSplines Estimation (b) Specificity of metaSplines Estimation

Figure 8. Sensitivity and specificity results for L_up Hockey Stick type trend for an AR(1) correlation structure with parameters: β = 1, IP = (t_q +1)/2, ρ = 0.7. Remaining parameters were varied to create 27 different combinations of repeated measurements, sample size per group, and σ. Points plot are the average result of B = 100 repetitions.

Table 1. Estimated sensitivity from metaSplines method for data simulated from each respective functional form for a total of 100 repetitions across 27 different parameter settings fixing the correlation structure to be AR(1) with ρ = 0.7. Parameter values used: $\sigma \in \{1, 2, 4\}$, $n_0 = n_1 \in \{10, 20, 50\}$, $q \in \{3, 6, 12\}$. Note that the Total Non-Missing Observations is less than the Total Observations.

Functional Form	Sensitivity	Total Repetitions	Non-Missing Estimates
Linear Increasing	1.00	2700	2686
Linear Decreasing	0.97	2700	2634
Quadratic: Concave Up	0.91	2700	2154
Quadratic: Concave Down	0.95	2700	2600
Oscillating 1	0.96	2700	2614
Oscillating 2	0.84	2700	2501
Hockey Stick 1	0.78	2700	2261
Hockey Stick 2	0.77	2700	2280

non-differential trends, Oscillating, and variable trends, Hockey Stick, had lower average sensitivity while the linear and quadratic trends tended to perform the best.

Continuous performance results

The continuous performance metrics for the cosine similarity, Euclidean distance and normalized Euclidean distance are shown in Figure 9 for the L_up trend with AR(1), ρ = 0.7. From this figure we see similar trends as the sensitivity results. Starting from the left most panel we see that the cosine similarity is highest when σ is small, q, n_0, n_1 are large. The spread of cosine similarity scores when q = 12 are very tightly clustered around 1 while the spread of values when q = 3 or q = 6 is larger. The center plot illustrates that using raw Euclidean distances with a small number of repeated measurements tend to have smaller distances, but this trend is not seen with normalized Euclidean distance in the last panel. Within each value of q in this middle panel there is a consistent trend that as the sample size per group increases the distance generally decreases. Finally moving to the last panel we have the normalized Euclidean distance, which can now be used to compare across different repeated measurement panels. We see a similar trend to the cosine similarity where the distance decreases, meaning better performance, for small σ and large q and $n_0 = n_1$.

Similar to the sensitivity performance metrics shown in Table 1, we can also compare the average value of the continuous performance metrics based on functional form. This is shown in Table 2. Similar trends appear in this

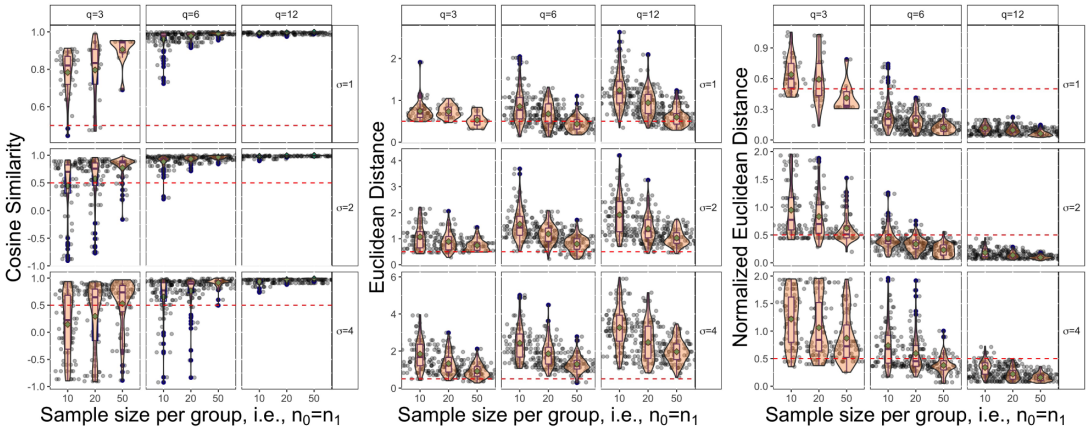


Figure 9. Estimated values of performance metrics including cosine similarity, Euclidean distance, and normalized Euclidean distance based on 100 repetitions for an L_up Hockey Stick trend with AR(1) correlation structure, ρ = 0.7, simulated across multiple settings varying repeated measurements q, sample size per group, n_0 and n_1 and σ. Note that the red dashed line serves as a reference point at 0.5 and the green dot in each panel represents the mean value across the 100 repetitions

Table 2. Average continuous performance metrics from metaSplines method for data simulated from each respective functional form for a total of 100 repetitions across 27 different parameter settings fixing the correlation structure to be AR(1) with ρ = 0.7. Parameter values used: $\sigma \in \{1, 2, 4\}$, $n_0 = n_1 \in \{10, 20, 50\}$, $q \in \{3, 6, 12\}$. Note that the Total Non-Missing Observations is less than the Total Observations.

Functional Form	Total Repetitions	Non-Missing Estimates	Avg. Cosine Similarity	Avg. Euc. Distance	Avg. Norm. Euc. Distance
Linear Increasing	2700	2686	0.99	1.26	0.07
Linear Decreasing	2700	2634	0.98	1.27	0.09
Quadratic: Concave Up	2700	2154	0.94	1.60	0.23
Quadratic: Concave Down	2700	2600	0.97	1.55	0.15
Oscillating 1	2700	2614	0.97	1.69	0.14
Oscillating 2	2700	2501	0.88	1.71	0.35
Hockey Stick 1	2700	2261	0.84	1.35	0.40
Hockey Stick 2	2700	2280	0.84	1.38	0.38

table with the linear trends having the highest average cosine similarity scores and lowest average normalized Euclidean distance and non-differentiable trends peforming worse.

Conclusions

With an increasing emphasis on understanding the dynamics of microbial communities in various settings, longitudinal sampling studies are underway. There remain many statistical challenges when dealing with longitudinal data collected from marker-gene amplicon sequencing. In order to validate and compare methods of estimation for longitudinal differential abundance a unified simulation framework is needed. Currently available simulation tools include R packages `seqtime`[19] and `untb`[20]. These packages focus primarily on simulation from the perspective of ecological processes aimed to capture the entire community dynamics. With `microboimeDASim` package[9] we instead provide the tools to simulate various functional forms for longitudinal differential abundance with added flexibility to control important factors such as the number of repeated measurements per subject, the number of subjects per group, within subject correlation, sequencing of time measurements,

etc. for a specific feature of interest. We have shown the benefit of these simulation tools by constructing a simulation design based on real microbiome data and showed the utility in methods evaluation using the metaSplines estimation procedure to compare the performance across a wide range of different parameter settings. In this manner the `microbiomeDASim` helps meet an important need in the research community to help in study design and compare existing methods as well as validate potentially novel methods.

Data availability

All data underlying the results are available as part of the article and no additional source data are required.

Software availability

microbiomeDASim is available at: http://bioconductor.org/packages/microbiomeDASim.

Source code available from: https://github.com/williazo/microbiomeDASim

Archived source code at time of publication: https://doi.org/10.5281/zenodo.3458563[9].

License: MIT.

Author contributions

JW performed analyses, implemented software and wrote first draft of article. HCB contributed to analysis and article review. JT and JNP oversaw analyses and designed experiment.

Acknowledgments

Authors would like to acknowledge Jane Fridlyand and Christina Rabe for helpful discussions and support.

References

1. Gopalakrishnan V, Spencer CN, Nezi L, *et al.*: **Gut microbiome modulates response to anti-PD-1 immunotherapy in melanoma patients.** *Science.* 2018; **359**(6371): 97–103.
PubMed Abstract | Publisher Full Text | Free Full Text
2. Routy B, Le Chatelier E, Derosa L, *et al.*: **Gut microbiome influences efficacy of PD-1-based immunotherapy against epithelial tumors.** *Science.* 2018; **359**(6371): 91–97.
PubMed Abstract | Publisher Full Text
3. Matson V, Fessler J, Bao R, *et al.*: **The commensal microbiome is associated with anti-PD-1 efficacy in metastatic melanoma patients.** *Science.* 2018; **359**(6371): 104–108.
PubMed Abstract | Publisher Full Text | Free Full Text
4. Sivan A, Corrales L, Hubert N, *et al.*: **Commensal Bifidobacterium promotes antitumor immunity and facilitates anti-PD-L1 efficacy.** *Sci Transl Med.* 2015; **350**(6264): 1084–9.
PubMed Abstract | Publisher Full Text | Free Full Text
5. Yatsunenko T, Rey FE, Manary MJ, *et al.*: **Human gut microbiome viewed across age and geography.** *Nature.* 2012; **486**(7402): 222–27.
PubMed Abstract | Publisher Full Text | Free Full Text
6. Kostic AD, Gevers D, Siljander H, *et al.*: **The dynamics of the human infant gut microbiome in development and in progression toward type 1 diabetes.** *Cell Host Microbe.* 2015; **17**(2): 260–73.
PubMed Abstract | Publisher Full Text | Free Full Text
7. Morris A, Paulson JN, Talukder H, *et al.*: **Longitudinal analysis of the lung microbiota of cynomolgous macaques during long-term SHIV infection.** *Microbiome.* 2016; **4**(1): 38.
PubMed Abstract | Publisher Full Text | Free Full Text
8. Leek JT, Scharpf RB, Bravo HC, *et al.*: **Tackling the widespread and critical impact of batch effects in high-throughput data.** *Nat Rev Genet.* 2010; **11**(10): 733–9.
PubMed Abstract | Publisher Full Text | Free Full Text
9. Williams J, Bravo HC, Tom J, *et al.*: **williazo/microbiomeDASim: Tools to simulate longitudinal differential abundance for microbiome data (v0.99.2)**. 2019.
http://www.doi.org/10.5281/zenodo.3458563
10. Turnbaugh PJ, Ridaura VK, Faith JJ, *et al.*: **The effect of diet on the human gut microbiome: a metagenomic analysis in humanized gnotobiotic mice.** *Sci Transl Med.* 2009; **1**(6): 6ra14.
PubMed Abstract | Publisher Full Text | Free Full Text
11. Paulson JN, Talukder H, Bravo HC: **Longitudinal differential abundance analysis of microbial marker-gene surveys using smoothing splines.** *bioRxiv.* 2017.
Publisher Full Text
12. Wilhelm S, Manjunath BG: **tmvtnorm: Truncated Multivariate Normal and Student t Distribution.** 2015.
Reference Source
13. Johnson NL: **Systems of frequency curves generated by methods of translation.** *Biometrika.* 1949; **36**(Pt. 1–2): 149–76.
PubMed Abstract | Publisher Full Text
14. Paulson JN, Stine OC, Bravo HC, *et al.*: **Differential abundance analysis for microbial marker-gene surveys.** *Nat Methods.* 2013; **10**(12): 1200–2.
PubMed Abstract | Publisher Full Text | Free Full Text
15. Paulson JN, Pop M, Bravo HC: **metagenomeSeq: Statistical analysis for sparse high-throughput sequncing**. Bioconductor package. 2013.
Reference Source
16. McMurdie PJ, Holmes S: **phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data.** *PLoS One.* 2013; **8**(4): e61217.
PubMed Abstract | Publisher Full Text | Free Full Text
17. GU C: **Smoothing spline anova models: R package gss.** *J Stat Softw.* 2014; **58**(5): 1–25.
Publisher Full Text
18. GU C: **Smoothing spline ANOVA models.** Springer, New York, 2nd edition, 2013.
Publisher Full Text
19. Faust K, Bauchinger F, Laroche B, *et al.*: **Signatures of ecological processes in microbial community time series.** *Microbiome.* 2018; **6**(1): 120.
PubMed Abstract | Publisher Full Text | Free Full Text
20. Hankin RKS: **Introducing untb, an R package for simulating ecological drift under the unified neutral theory of biodiversity.** *J Stat Softw.* 2007; **22**(12).
Publisher Full Text

Open Peer Review

Current Peer Review Status: ✔ ✔

Version 2

Reviewer Report 26 February 2020

https://doi.org/10.5256/f1000research.24837.r60557

✔ **Leo Lahti**
Department of Future Technologies, University of Turku, Turku, Finland

The authors have responded to my review comments appropriately.

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Microbiome bioinformatics.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Reviewer Report 06 November 2019

https://doi.org/10.5256/f1000research.22722.r55802

✔ **Kris Sankaran**
Montreal Institute for Learning Algorithms (MILA), Montreal, QC, Canada

Contributions

The authors have developed an R package to simulate longitudinal microbiome time course data, especially where there are difference in trajectories between treatment and control groups. This can be used to address,

1. Experimental design: Simulations can guide power analysis, to see whether a proposed study will be well-powered, as a function of assumptions on the generating mechanisms.

2. Methods comparisons: The effectiveness of different methods will depend on the structure of the data, and simulations provide ground truth from which to make assessments.

They simulate data one species at a time. Both treatment and control groups are assumed to have gaussian data, truncated below at 0 to reflect transformed counts. Control data are assumed to be drawn from some common mean, but with specified correlation structure over time. Treatment data are assumed to have a mean that deviates from the control according to some function f(), but have the same correlation structure. The authors provide an interface for simulating a few patterns of f() that are believed to be common in real data (e.g., oscillating, quadratic, and linear shapes).

The authors share code to display simulated data. They also describe a study evaluating the power of a particular method, 'metaSplines', as simulation parameters are changed.

Evaluation

Strengths:

- I like the idea of formalizing simulation-based power analysis. In the microbiome setting, simulations make more sense than theory, but have two issues (1) they are potentially labor-intensive and (2) they can be ad hoc, and never published. By preparing a package, the authors lower the barrier to entry to / introduce a more formal standard for this work, hopefully enabling simulation-based power analysis in the field.

- The paper is generally technically sound, and reads well. Code is available publicly, is clearly documented, and written in a professional style.

Weaknesses:

- The simulated data are never properly evaluated -- this is my reason for the "partly" response in my report. Of course, any simulation is only an approximation of reality, but it would be nice to know along which dimensions the approximation is close, and along which it is poor. This would also set the stage for studying whether the conclusions that you're aiming for (study design or methods choices) are substantially affected by / robust to these deviations in real data. Something in the spirit of graphical inference could be quite interesting here.[1]

Missed Opportunities:

- The 'metaSplines' analysis ends somewhat abruptly, because it's not clear what actual conclusions would be drawn from it. I think it would be interesting if you compared another method against it, because you'd be getting at something like the relative efficiency of the approaches (you could also measure their robustness to particular assumptions).

- The functional forms seem somewhat restrictive, though I see their value for people who don't want to spend time writing code. Could you define some kind of interface that makes it easier for people to specify classes of alternatives? E.g., maybe you could let people draw functions interactively, or use as input some examples of microbiome series they see in real data.

Discussion

- I have trouble believing in any kind of i.i.d. assumption across species. First, the scale of abundance across species tends to differ by orders of magnitude. Second, many species exhibit very similar behavior.

- Among the controls, couldn't some species also vary over time, because of factors in that individual that change which are not specifically treatment?

- Setting missing data to 0 is generally bad practice, because then you can't distinguish true zeros from missingness. You should either do proper missing data imputation, or recommend methods that explicitly model the missign values / don't require measurements at equal timepoints.

- The different correlation structures you propose reflect an equispaced sampling design. It wouldn't be too hard to change the correlation structure to allow for unevenly spaced sampling, and it would address your point (4, "Asynchronous repeated measures").

- Could you create an interactive notebook? E.g., using binder: https://mybinder.org/v2/gh/krisrs1128/microbiome_dasim_example/master. This would make it easier for people (esp. nonexperts) to get acquainted with your work, without having to install jupyter etc.

- For dosage effects, I'd find a (reversed) sawtooth or wavelet-style spike more believable than an oscillating function. But again, this is related to the point of letting people choose their own alternatives.

Minor Comments

- The caption in Figure 5 seems deprecated.

- I don't think you ever defined "OTU".

- The library load should say "microbiome" not "microbime".

- There are still a few typos here and there (e.g., "differential abundant" features and "metrics of success results"), so I recommend another careful read.

References

1. Wickham H, Cook D, Hofmann H, Buja A: Graphical inference for Infovis.*IEEE Trans Vis Comput Graph* . **16** (6): 973-9 PubMed Abstract | Publisher Full Text

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: statistics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 19 Feb 2020

Justin Williams, Genentech, Inc, South San Francisco, USA

Thank you for your careful review of the manuscript and suggestions. Responses to issues raised are shown below for specific points raised.

Weakness
In the vein of evaluating the robustness of the simulation in approximating reality we have included an additional section “Approximating Observed Microbiome Data” that aims to show how the current package could complement real-world microbiome data. Some of the implications and thought processes for using the simulation package in this setting are discussed within the details of this section.
Missed Opportunities

1. We thank the reviewer for this comment. The metaSplines analysis that is included in the manuscript is meant to serve as an illustration of how the simulator could be used to evaluate longitudinal differential abundance methods. In the interest of focusing this software tools manuscript on the simulator package itself, a full comparison of different methods was not investigated. However, this would be a valuable avenue to explore in more depth in a subsequent write-up.
2. Presently we are not aware of any interface within R that would dynamically allow users to draw functions. This would be highly useful and we would like to continue adding in different functional forms within the package. The currently available forms were an initial foray into some potentially relevant types of trends that might be observed. Users with R expertise can modify the mean_trend function to create alternative functional forms, but allowing full user specification may create an unintended burden for many practitioners. In the future, we will consider some alternative options that allow for higher flexibility while maintaining usability.

Discussion

- In our simulation design we are restricting to a single feature of interest when generating data and therefore are inherently ignoring variability across species. This feature simulation can be tailored for individual species of interest and would be run separately in each case.
- The control group could also vary over time, but from a simulation perspective we are treating the design as if the sample has been norm referenced across time for the control group. Since the main goal of estimation is calculating the difference between the treatment

and control group over time, restricting the control group to be invariant over time simplifies the user input and maintains the primary goal of estimation.

- By default when inducing missingness in the data, the values are treated as NA rather than 0. However, we included the option to specify the value of the missing data to represent cases where there may be some true non-zero occurrence but due to technical limitations such as read depth the values do not appear. The process of generating missingness is meant to align with some of the typical issues such as loss to follow-up when conducting these types of longitudinal designs.
- Thank you for this comment - as a result we have decided to expand the functionality to allow for asynchronous sampling over a specified interval (using asynch_time=TRUE) or alternatively to have the user specify discrete sampling times for each individual with the mvrnorm_sim_obs function. An example of using each of these asynchronous sampling schemes have been included in the updated manuscript. The compound and independent correlation structures remain unchanged in this unevenly spaced sampling design, but the AR(1) correlation structure now incorporates the amount of time between each sample as |t_{i}-t_{j}|.
- Thank you for this suggestion. The original instructions for installing and running Jupyter with an R kernel were indeed cumbersome. To make the notebook easily interactive, we have re-compiled the materials using Google Colab with a simple badge on top that will allow users to run the code without requiring local installation and setup of Jupyter.
- Thank you for pointing out these possible functional forms. We will work to expand the functional forms available to include these types of trends in the future. As mentioned earlier the ability to define the mean trend has a natural tradeoff between flexibility and useability.

Minor Comments

Caption texts, grammatical errors, and typos pointed out have been corrected. Additional read throughs have also been performed to minimize these types of mistakes in the latest draft.

Competing Interests: No competing interests were disclosed.

Reviewer Report 05 November 2019

https://doi.org/10.5256/f1000research.22722.r55801

?

Leo Lahti

Department of Future Technologies, University of Turku, Turku, Finland

This manuscript introduces a new method for simulating longitudinal differential abundance for microbiome data. The method is implemented as an R/Bioc package. The proposed package allows the user to simulate longitudinal microbiome data based on various assumptions, and allows the tuning of key design aspects such as signal-to-noise ratio, correlation structure, effect size and zero inflation. One of the available methods is validated with benchmarking comparisons.

The manuscript is technically sound and written in a fluent and easily understandable English. Experiments and statistical analyses have been conducted rigorously. The source code and experiments are openly available via Github but I have not tried to replicate the analysis.

Realistic simulations are valuable for study design, and help to address questions about sample size, density of time points, experimental costs, etc. The work provides pragmatic solutions to a topical problem in microbiome bioinformatics.

Major comments:

1. The simulator provides versatile options to tune signal shape, correlations, and noise. However, I am left wondering how well the simulations correspond to real microbiome data. In particular, it is not clear nor validated how the time series shape and correlation structures correspond to known processes in microbial ecology, such as neutral process, competition models (such as generalized Lotka-Volterra), compositionally aware naive models (Dirichlet-Multinomial), mean-reversing processes (Ornstein-Uhlenbeck). All of these have ecological interpretations and have been visible in recent microbiome time series literature. These models are motivated by known ecological processes, rather than technical modifications on the signal shape; it would be relevant to know how large impact the chosen modeling assumptions might have on the results. Can we expect that the proposed simulator will yield qualitative similar conclusions, even if the connection to ecological mechanisms might be weak?

2. The proposed model does not (explicitly) account for heteroschedasticity or overdispersion, and its performance has not been demonstrated with recently popular models of differential abundance, such as DESeq2. It could be true that longitudinal testing of differential abundance requires different methodology. But longitudinal simulators can be also used to simulate cross-sectional data, which is always a snap-shot of longitudinal data. I wonder if the simulator would perform well with standard methods for cross-sectional data; or if it can be shown to yield similar overall distributions. This could provide some additional support for the simulations as the feasibility of the modeling assumptions and their impact on the conclusions remains open.

Minor comments:

1. Other simulators for microbiome data and time series are available. One that I am aware of is the seqtime package (https://github.com/hallucigenia-sparsa/seqtime), although that is only available as an R package (and not formally published), but there may be other recent simulators. I did not find other simulation works being cited, it would be good to check if other simulators can be identified in the recent literature, and how they relate to this work.

2. Lack of integration with phyloseq is a weakness, as this class structure is now very popular among the microbiome R users, and many tools build directly on that class structure. It would be useful addition to the package if the simulations could be made available in a phyloseq format.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Microbiome bioinformatics.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 19 Feb 2020

Justin Williams, Genentech, Inc, South San Francisco, USA

Thank you for your review of the manuscript and suggestions for improvement. Both the manuscript and package have been updated to reflect issues raised above. In the following we address point wise specific comments raised from Version 1 of the manuscript.

Major Comments:
(1) Thank you for this comment. As an additional step to address the ability of the simulator to reflect real microbiome data we have provided an example of approximating clinical data with longitudinal microbiome data in mice from Turnbaugh et. al, 2009. This section was added to the manuscript under “Approximating Observed Microbiome Data” with further details about how the simulator can be used to complement and expand clinical efforts.

In particular, we outline some of the steps to consider when constructing a simulated dataset to approximate a real-world study. Although our simulation design does not explicitly account for ecological processes as mentioned, the focus on the underlying distributional assumption defines the scope of problems which can be addressed.

The simulator looks to construct values for a single feature (aggregated at the taxonomic level of interest) and thus does not incorporate correlation between features or compositional constraints. By focusing on only single features of interest we expect that the simulator will yield similar conclusions to those observed in clinical experiments, and thus offers practitioners a useful tool when designing or expanding a longitudinal microbiome study.

(2) During the construction of the simulator the variance between both groups is held constant, partly in order to reduce the burden of parameter specification on the user. This choice also reflects a belief that the two groups differ only in their mean trend over time, which is often an appropriate default assumption without particular beliefs about how the heteroskedasticity may differ by group over time. However, it is worthwhile to consider adding a heteroskedastic option to the simulator to incorporate potential differences in noise between groups. While the goal of the

simulator focuses on longitudinal designs, it is worthwhile to explore its applicability to cross-sectional data. The simulator function can simulate cross-sectional data by setting num_timepoints=1. Further evaluation of the performance in these cases is merited, but falls outside the scope of this initial software tools manuscript.

Minor Comments:
(1) We thank the reviewer for pointing to these additional simulator packages. A further investigation of the literature returned multiple packages including seqtime, untb, and WrightFisher with similar goals for simulating longitudinal trends. These packages however focus on simulations from a compositional perspective rather than at a single feature level, and lack some of the documentation and formal publication that accompanies our present package. I have updated the manuscript to include references to these additional packages and note some of the differences in the conclusion.

(2) Thank you for this comment. We have added additional conversion functions simulate2MRexperiment and simulate2phyloseq that format simulated data into the respective objects of interest for the metagenomeSeq and phyloseq packages. We have also added details about using these functions within the manuscript.

Competing Interests: No competing interests were disclosed.