Briefings in Bioinformatics, 22(6), 2021, 1–11

https://doi.org/10.1093/bib/bbab223
Case Study

Human host status inference from temporal microbiome changes via recurrent neural networks

Xingjian Chen, Lingjing Liu, Weitong Zhang, Jianyi Yang and Ka-Chun Wong

Corresponding author: Ka-Chun Wong, Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong SAR. Tel: +852 34428618; E-mail: kc.w@cityu.edu.hk

Abstract

With the rapid increase in sequencing data, human host status inference (e.g. healthy or sick) from microbiome data has become an important issue. Existing studies are mostly based on single-point microbiome composition, while it is rare that the host status is predicted from longitudinal microbiome data. However, single-point-based methods cannot capture the dynamic patterns between the temporal changes and host status. Therefore, it remains challenging to build good predictive models as well as scaling to different microbiome contexts. On the other hand, existing methods are mainly targeted for disease prediction and seldom investigate other host statuses. To fill the gap, we propose a comprehensive deep learning-based framework that utilizes longitudinal microbiome data as input to infer the human host status. Specifically, the framework is composed of specific data preparation strategies and a recurrent neural network tailored for longitudinal microbiome data. In experiments, we evaluated the proposed method on both semi-synthetic and real datasets based on different sequencing technologies and metagenomic contexts. The results indicate that our method achieves robust performance compared to other baseline and state-of-the-art classifiers and provides a significant reduction in prediction time.

Key words: longitudinal microbiome; host status inference; deep learning; feature extraction; data preparation

INTRODUCTION

With the development of high-throughput sequencing technologies, the human microbiome has been a focus of study to explore the potential relationships between the microbiome composition and relevant host status [1–3]. The resulting large amount of sequence data motivates researchers to develop machine learning methods to predict patient phenotype [4, 5].

Generally, the microbiome sequencing reads obtained by the sequencing pipeline can be reorganized into operational taxonomic units (OTUs) according to their sequence similarities and the count for each OTU represents the abundance of a particular bacterial taxon [6]. Recent studies also utilize amplicon sequence variants as the standard unit of marker-gene analysis and reporting [7, 8]. Under the assumption that microbiome composition is different between individuals with

Xingjian Chen has been pursuing his PhD degree from the Department of Computer Science, City University of Hong Kong. His research interests include bioinformatics and deep learning.
Lingjing Liu has been pursuing her PhD degree from the Department of Computer Science, City University of Hong Kong. Her research interests include cancer detection and computational biology.
Weitong Zhang has been pursuing her PhD degree from the Department of Computer Science, City University of Hong Kong. Her research interests include cancer detection and bioinformatics.
Jianyi Yang is a Professor with the School of Mathematical Sciences, Nankai University. His research interests include structural bioinformatics, protein function prediction and deep learning.
Ka-Chun Wong is an Assistant Professor with the Department of Computer Science, City University of Hong Kong. He leads the East Asian Bioinformatics and Computational Biology laboratory and conducts high-impact computing research.
Submitted: 24 March 2021; **Received (in revised form):** 21 April 2021

different host statuses [9], the extracted abundance table can be treated as a species-by-subject matrix, combining qualitative labels of host status description to formulate a machine learning task.

Some of the existing studies followed the aforementioned flows and achieved promising prediction performance [5, 10, 11]. By summarizing the related publications, we can discover that the improvements generally fall into two categories: feature engineering and classifier [4, 12]. The first approach is to combine domain knowledge to extract features from raw data via basic data mining techniques. For example, Pasolli *et al.* used a random forest (RF) to select the best features during the model generation phase [13]. Zhu *et al.* introduced a feature selection method based on graph embedding to validate the molecular meaning of microbial markers [14]. Oudah and Henschel proposed a phylogenetic hierarchy-based algorithm for feature engineering to distinguish healthy and disease subjects [15]. Ditzler *et al.* developed Fizzy to select the information-theoretic subsets for biological features [16]. The second approach usually focuses on exploring advanced classifiers tailored for the characteristics of microbial abundance data. Zhu *et al.* introduced deep forest to study for microbiome-disease associations [17]. Johnson *et al.* proposed a k-nearest-neighbor regressor to predict postmortem interval [18]. Ditzler *et al.* developed a recursive neural network to generate a tree representation for metagenomic classification [19]. Reiman *et al.* adapted a convolutional neural network to explore the spatial structures of abundance profiles [20].

Those machine learning-based methods are the existing approaches in the host status inference. However, all of them are based on the single-point microbiome composition. Moreover, most of the underlying tasks are about disease status prediction (i.e. to predict the presence or absence of a disease), without considering other host statuses. Recently, the development of longitudinal microbiome studies enables us to investigate complex dynamic microbiome patterns from multiple time series measurements. In particular, the longitudinal sampling of the microbiome may be more informative and accurate for host status inference. For a longitudinal study, a subject involves several observations collected at different time points, each of which is quantified in multiple OTU features [21]. The number of OTU features can vary dramatically across different studies due to the experimental contexts, and the observations for each subject are often collected at inconsistent time points. Therefore, the time intervals and the number of observations for each subject can be different [22]. This uneven and inconsistent temporal data poses difficulties in building a supervised prediction model [23, 24]. Therefore, most disease prediction studies as aforementioned simply picked out only one observation sampled at the last time point (usually known as single point or endpoint) to represent the whole microbiome composition for a subject, which actually ignored the highly dynamic changes of the human microbiome.

Throughout the related studies in the past several years, there are only two publications retrieved to analyze temporal microbiome data by applying machine learning-based methods. Metwally *et al.* firstly attempted to utilize longitudinal taxonomic profiles to predict food allergy via long-short term memory (LSTM) [22]. However, the performance is not good possibly due to the lack of systematic data preprocessing and feature engineering. In addition, the proposed method in this study is restricted to predict the food allergy. Therefore, its generalization to other tasks remains speculative. On the other hand, microbiome interpretable temporal rule engine (MITRE) was afterward introduced to extract rules from microbiota temporal data and explore the relationship with its host status [21]. Although the performance is better, it takes several days to train the model, even for very small datasets. Consequently, the methods above cannot achieve the desired performance in terms of accuracy and running time at the same time.

In this work, the analysis is conducted in the context of OTU-level data. Since there are already many existing works on disease status prediction, we mainly focus on solving other host status inference tasks. Given the inherent advantages of recurrent neural network (RNN) for time series data, we propose a host status prediction framework based on gated recurrent unit (GRU) neural network [25] to predict the host status from microbiome temporal data. Our work provides the following contributions:

1. For data preprocessing, we introduced a novel average imputation method to deal with the microbiome temporal data with inconsistent sampling, achieving promising performance compared with traditional methods.
2. For feature engineering, we created new features based on the phylogenetic tree, which can integrate the phylogenetic hierarchy relationship to enrich the microbiome feature representation further.
3. The proposed pseudo-subject representation makes it feasible to conduct and compare traditional feature extraction algorithms while keeping temporal information for microbiome data.
4. To the best of our knowledge, this is the first time that a comprehensive deep learning-based framework is proposed for host status inference, combining tailored data preprocessing and feature engineering steps. The experiment results demonstrate a good trade-off between the prediction precision and running time.

MATERIALS AND METHODS

Datasets

To validate the generalization of our proposed framework, we conduct the experiments on a diverse range of both simulation and real datasets. Following the previous studies [21, 22], we consider the binary classification in this study. For simulation experiments, to keep a fair comparison with previous studies, we follow the same simulation procedures and settings of MITRE [21] to generate semi-synthetic datasets. The semi-synthetic datasets are bootstrapped from [26] to predict the delivery type, and the same as MITRE; we set single-clade perturbation and two-clade perturbation on the phylogenetic tree to simulate the disease subject. The detailed simulation process can be found in Supplementary Method 1.1. After simulations, both new datasets are composed of 32 subjects, and each subject includes 308 OTU features with 18 observations.

The real datasets are selected based on different metagenomic contexts and data sources (16S rRNA amplicon sequencing, shotgun metagenomic sequencing and NIH Human Microbiome Project (HMP)). Based on 16S rRNA amplicon sequencing, we select the first dataset from [27]. They studied the influences of dietary intake type for the human gut microbiome and explored the individual differences between animal-based diet and plant-based subjects. The second dataset is from [26]. They profiled the early-life microbiome development of infants to

Table 1. List of the classification tasks for different studies. The number of total subjects is obtained after the data preprocessing

Study	Dataset code	Variable predicted	Positive subjects	Negative subjects	Total subjects
Simulation (Bokulich *et al.*, 2016)	Single-clade	Cesarean delivery	16	16	32
	Two-clade	Cesarean delivery	16	16	32
David *et al.* (2014)	David	Plant-based or animal-based diet	10	10	20
Bokulich *et al.* (2016)	Bdiet	Formula-dominant diet	11	24	35
	Bdeliv	Cesarean delivery	13	22	35
Vatanen *et al.* (2016)	Knat	Russian nationality	30	83	113
	Kegg	Egg allergy	25	84	109
	Kige	Elevated IgE levels	28	81	109
Brooks *et al.* (2017)	Brooks	Cesarean delivery	10	19	29
Hall *et al.* (2017)	Hall	IBD	13	19	32
Heintz-Buschart *et al.* (2016)	Heitz	T1D	10	10	20
Raymond *et al.* (2016)	Raymond	Cephalosporins	36	35	71
Vincent *et al.* (2016)	Vincent	CDI	90	8	98
Shao *et al.* (2019)	Shao	Cesarean delivery	416	336	752
iHMP	HMPibdmdb	IBD	103	27	130
iHMP	HMPt2d	T2D	20	8	28

illustrate the complexity of its sensitivity to perturbation. The third dataset is from [28]. They followed the gut microbiome development of hundreds of infants from birth until age three to uncover the potential mechanism linking to immune diseases. For Bokulich dataset, we define two tasks to predict infant diet and delivery mode, respectively. For Vatanen dataset, we define three tasks made up of nationality, allergens, and serum IgE (Immunoglobulin E) levels. For David dataset, we define one task to predict the diet type.

In the context of shotgun metagenomic sequencing, we retrived six more datasets from curatedMetagenomicData R package [29]. For BrooksB dataset, we define the task to predict cesarean delivery or normal delivery [30]; for HallAB dataset, we define the task to predict whether the subject has inflammatory bowel disease (IBD) or not [31]; for HeizBA dataset, we define the task to predict whether the subject has type 1 diabetes (T1D) or not [32]; for RaymondF dataset, we define the task to predict whether the subject used antibiotics (Cephalosporins) or not [33]; for VincentC dataset, we define the task to predict whether the subject has *Clostridium difficile* infection (CDI) or not [34]; and for ShaoY dataset, we define the task to predict cesarean delivery or normal delivery [35].

In addition, we entered the HMP website (https://hmpdacc.org/) to download and select two more HMP datasets [36]. For HMPibdmdb dataset, we define the task to predict whether the subject has IBD or not; for HMPt2d dataset, we define the task to predict whether the subject has type 2 diabetes (T2D) or not. The detailed data sources and selection standards for all the datasets are described in Supplementary Method 1.2. We listed the description for all the predicted tasks among different studies in Table 1.

Overview of the framework

Firstly, since unevenly sampling time intervals can increase the difficulty for classification, the raw data are preprocessed and filtered by the recommended data preprocessing measurements (See Section 2.3). Then, the dimensions of the time OTU features are reduced by feature extraction, and essential features are obtained. Finally, the temporal abundance data are sent to GRU neural network to train and validate the model. The pipeline of our method can be described in Figure 1.

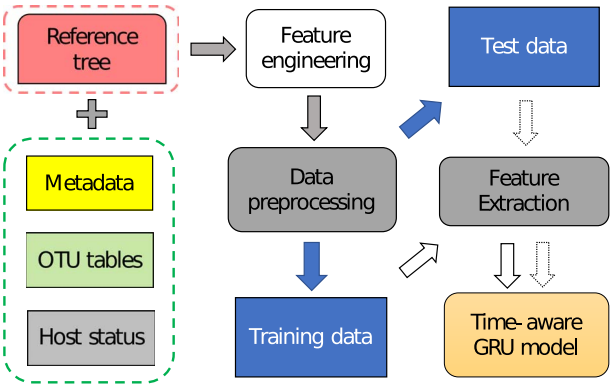

Figure 1. Pipeline of the proposed deep learning-based framework for human host status inference. For each input subject, we need four files, which are Metadata, OTU table, host status as well as reference tree. Metadata is the corresponding time point for each observation in a subject. OTU table is an $M*N$ matrix while M represents the number of observations and N represents the dimension of the OTU abundance features. Host status represents the binary description of the predicted variables. Reference tree is produced by the analysis pipeline (pplacer [37]). The blue arrows mean the data splitting, the solid-line arrows represent the flow of the training data while the dash-line arrows represent the flow of the test data. The dataset size for each dataset can be found in Table 1.

Feature engineering and data preprocessing

In this work, the pipeline takes in the following input: OTU abundance tables measured over time for each host, phylogenetic placements on the reference tree (produced by pplacer [37]) and the relevant host status. Firstly, we need to preprocess the raw sequencing data to obtain the OTU abundance tables. Different pipelines can generate different OTU features, thereby influence the prediction performance. To keep a fair comparison, we directly utilize the abundance tables and phylogenetic tree processed by MITRE [21] under the default parameter setting. MITRE utilized DADA2 1.1.5 [38] and mothur 1.35.1 [39] for denoising, quality filtering, sequence assignment, clustering and taxonomical classification.

For feature engineering, we introduced a phylogenetic aggregation method to generate new features from the phylogenetic tree. Specifically, the relative abundance estimate of ancestral nodes is obtained by summing the relative abundances of its

children for each node in the phylogenetic tree and then added to the end of the temporal abundance table. The procedure is depicted in Algorithm 1.

Algorithm 1: Phylogenetic aggregation for feature engineering.

Data: Taxa abundance vector X and a phylogenetic tree $G = \{V, E\}$ with the maximum depth of L
Result: New taxa abundance vector U

Initialization $l = L, U = X$;
while $l \geq 0$ **do**
  for each node v in layer l **do**
    if the label of v is an OTU in vector X **then**
      assign the abundance of the OTU from vector X to node v;
    else if v has any children **then**
      add its children's abundances to the abundance of v;
      $U \leftarrow Append(U, v)$;
    end
  end
  $l \leftarrow l - 1$;
end
return U

The operation 'Append' means to add the new elements into the vector. For more details, please see Supplementary Method 1.3. In addition, since the subjects in the datasets can have different numbers of time points and the sampling density is also different, we need to implement data preprocessing and filtering steps for the raw time series data to decrease the noise and keep the input consistency. The following procedures are taken for each dataset:

1) Drop the sequences/OTUs with the sequencing read count (summing across all the observations) less than the platform-specific threshold (i.e. 5000 for HiSeq/MiSeq data [21]) and remove the inner nodes that are not needed to maintain the phylogenetic tree topology (refer to [20]);
2) Discard all the observations where data summed over OTUs are too low because of the shallow sequencing depth (typically less than 10 [4]). For each subject, the data are added together along the OTU axis and only those time points greater than the threshold are kept;
3) Drop the subjects with inadequately dense temporal sampling [21]. Specifically, we drop the subjects with the observations less than three time points;
4) Discard all observations with the time points before the start of the experiment and after the end of the experiment for time window consistency (e.g. the observations before day -4 in [27]), since such observations are not available outside that period for most subjects [21]).

In addition, since the distribution of the abundance data is highly skewed, we converted the OTU counts of each observation into relative abundances by adding a pseudo-count of one and normalized the data using the logarithmic transformation. Besides, as the sampling inconsistency for each subject in the study can increase the difficulty for classification, and some subjects are missing features at some time steps, we utilized an adaptive imputation approach to sample the time steps from small time patches to keep the dynamic patterns to the greatest extent. Specifically, we averaged the abundance data across all time points for each subject, as defined by dividing the experiment into certain amounts of equal pieces and taking any certain consecutive intervals as a valid time window. Parameters are chosen to ensure each divided period contains at least one observation for each subject and maximizes the temporal resolution. After selecting the window size w and sliding interval d, the algorithm procedure for average imputation is depicted in Algorithm 2. The operation 'Mean' means to average all the elements between $X[i * d]$ and $X[i * d + w]$. To further validate the performance of our imputation method, we compared different imputation methods and demonstrate that the average imputation works the best on the datasets of interest. The detailed imputation approach and the comparison results can be found in Supplementary Method 1.4 and 1.5.

Algorithm 2: Average data imputation.

Data: Taxa abundance vector X, window size w and sliding interval d
Result: New taxa abundance vector U

Initialization $X \in \mathbf{R}^n, i = 0$;
while $w + i * d \leq n$ **do**
  $v = Mean(X[i * d] : X[i * d + w])$;
  $U \leftarrow Append(U, v)$;
  $i \leftarrow i + 1$;
end
return U

Pseudo-subject-based feature extraction

Since the dimension of OTU abundance features for each subject can be extremely high, it is of great importance to select the best features to reduce the computational complexity and increase the generalization of model [16, 40]. Although there are many commonly used feature selection and dimension reduction algorithms in microbiome studies such as RF, auto encoder (AE), linear discriminant analysis (LDA), principal component analysis (PCA), etc. However, there is not any systemic framework to integrate and validate these methods for microbiome longitudinal studies. Moreover, most of the existing methods can merely accept the input of flattening or single-point-based microbiome data. For a subject composed of several observations at different time points, flattening or choosing the last observation over the time window to get a one-dimensional representation can lead to the deficiency in temporal information. For deep learning-based methods such as convolutional autoencoder, which can accept flexible inputs such as a sequence or an image, the performance cannot be guaranteed since it trains without labels. Therefore, for feature selection, we propose a pseudo-subject-based representation for each observation in a subject. Specifically, in a microbiome longitudinal study, the host status of a specific subject usually keeps invariable from the beginning of the experiment till the end. This fact allows us to focus on studying the dynamic changes of the microbiome composition with fixed host status without worrying about changes in the host status in a time window. Consequently, we consider treating each observation (time point) in a subject as a pseudo-subject and give these observations the same labels as their relevant subjects. Based on that, we can compress the OTU feature dimensions while keeping the same temporal dimensions as before.

To further verify the rationality such that the proposed approach indeed does not affect the performance of the model, we compare the proposed feature extraction approach with functional PCA (FPCA), and a series of traditional feature extraction methods based on flattening vectors. Moreover,

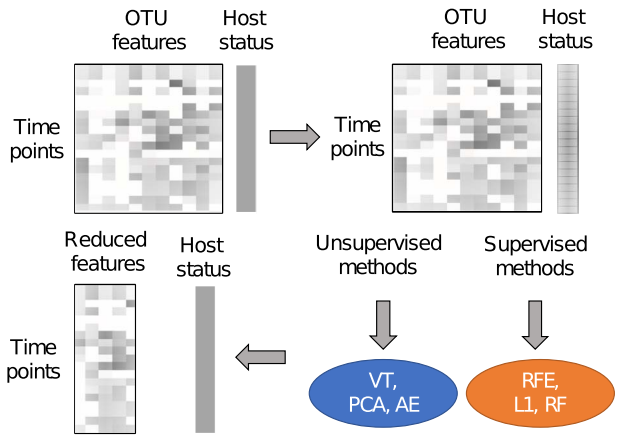

Figure 2. Flow diagram of the pseudo-subject-based feature extraction. The upper right corner represents the raw abundance profile and its host status, then for each observation in the raw subject, the feature selection methods are conducted. Finally, we obtain the reduced OTU features as well as keeping the previous temporal dimension and host status label.

we conducted a series of comparison experiments based on two directions, feature selection and dimension reduction, to compare the effects of different feature extraction algorithms. Feature selection based methods include variance threshold (VT), recursive feature elimination (RFE), L1-based feature selection (L1) and RF. Dimension reduction based methods include PCA and AE. In particular, we added a sparsity constraint (L1 Regularization) based on AE to construct a sparse autoencoder (SAE) to get the compressed representation of the input. The detailed parameter settings and the implementations can be found in Supplementary Method 1.6. Particularly, for feature extraction methods utilized in this study, no matter for supervised methods (RFE, L1, RF) or unsupervised methods (SAE, VT, PCA), the experiments were strictly conducted in each cross-validation iteration and we only trained the feature extraction models based on the training data. Even for supervised methods (which use subject labels), we make sure that the model is trained only with the labels of the training data to avoid data leakage and overestimated bias. After the feature selection step, we aggregated all the observations in each subject as a matrix, according to the initial orders of their corresponding sampling time points. The matrix is then sent to the RNN to train the classifier. The flow diagram of the pseudo-subject-based feature extraction is depicted in Figure 2.

Recurrent neural network

Since the OTU abundance features are substantially multivariate time series data and existing studies have proved that RNN is skilled to explore the temporal information [41, 42], we utilize the RNN-based model to complete the prediction of host status in this study. RNN can accept the input variables at different time points and keep the features from the previous time points. During the training, RNN has a different state at different time t, and the output of the hidden layer at time $t-1$ will be applied to the hidden layer at time t, which can be described as Equation 1 and 2.

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \tag{1}$$

$$y_t = \sigma_y(W_y h_t + b_y) \tag{2}$$

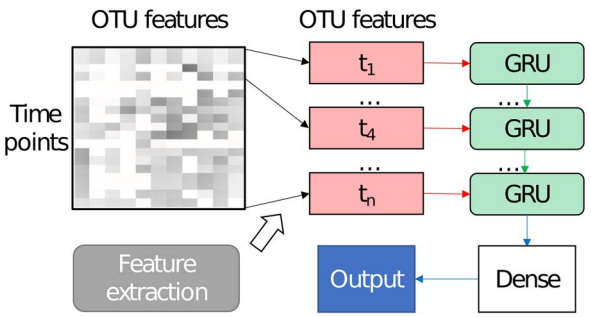

Figure 3. GRU framework for microbiome time series data. The input time series data are the OTU matrix measured at several time points. Each row in the matrix represents an observation at a specific time point. The temporal data are then sent to the GRU unit to train the classifier.

where h_t is the hidden state of the current node and h_{t-1} contains the information from the previous node. x_t is the current input, and y_t is the output. W_h and U_h are the weight matrices connecting the units of input layer and hidden layer. σ_h and σ_y are the activation functions, b is the bias. Mathematically, this structure can utilize the past states to infer the status of the current host in our study. However, the basic RNN has two main problems, the disappearance of the gradient and the explosion of the gradient. Therefore, LSTM was proposed to avoid long-term dependencies of RNN [43]. LSTM relies on the gate structures to allow information to selectively affect the current state. In this study, we applied GRU, which can be regarded as a variant of LSTM, as the deep learning classifier. Compared with LSTM, GRU can achieve a considerable effect. It is easier to train and significantly improve the training efficiency by combining the forget gate and the input gate into one update gate [44]. The GRU cell can be described as Equations 3 to 6.

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{3}$$

$$r_t = \sigma(W_t x_t + U_t h_{t-1}) \tag{4}$$

$$\tilde{h}_t = \tanh(W_{x_t} + U(r_t * h_{t-1})) \tag{5}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{6}$$

where z_t is the update gate that replaces the forget gate and the input gate of LSTM. The update gate merges the cell state c_t and the hidden state h_t. In addition, the calculation of new information at the current moment is different from LSTM. Here r_t is the reset gate, h_t is the final hidden state and $\tilde{h}_t$ is the candidate state at time t. In this study, the input for each subject is a time-species matrix in which each row means each observation and each column means the time points. The proposed GRU framework in our study is showed in Figure 3.

As seen in Figure 3, the state of the last observation will be determined by all the previous observations. Compared with conventional machine learning algorithms, our deep learning model can utilize all the observations in a time window to train the model without losing any temporal information. In our study, the initial number of the hidden nodes for the GRU model is set to be 512. Dropout and weight decay are utilized to prevent overfitting and ensure generalization performance. Especially, we applied early stopping to avoid the model over-trained. The patience is set to be 10 and the tolerance is set to be 0.00001 for all the datasets. Besides, We employs stochastic gradient descent as

the optimizer, and the momentum is set to be 0.9 to speed up the convergence. Cross-entropy error is utilized as the loss function to train the model. ReLU is used as the activation function for all the layers. The detailed basic learning rate, maximal training epoch and the batch size are selected based on the experimental experience (Supplementary Method 1.7).

EXPERIMENTS AND RESULTS

In our work, we used Keras as the deep learning implementation for its accessibility and flexibility. The computing environment is Intel Core i7-9700K CPU, 3.60GHz frequency, GTX 2080Ti graphics card, 64G memory and 1T hard disk. The operating system is Ubuntu 18.04, and Anaconda platform is used for Python development. In addition, to speed up the training process of our GRU model, we utilized the GPU acceleration technology of CUDA.

Evaluation metrics

In order to keep a fair comparison, we utilized the same cross-validation standard for all the methods utilized in this study. Specifically, the final results are obtained by averaging 10-fold cross-validations for 10 times. For each fold in each single 10-fold cross-validation, we ensure that the test set is independent and not used in the training process. When splitting the training and test data, we utilized the stratified K-fold function in scikit-learn [45] to obtain the same fraction of positive and negative samples in training and test data. Folds are stratified by preserving the percentage of samples for each class. Besides, we treat each subject as independent and split the subjects rather than the observations in subjects to make sure that the same subject cannot be present in both training and testing folds. In addition, since random initialization is involved in specific methods (e.g. random sample/feature split in a random forest, random weight initialization in a deep neural network), it can lead to different results. Therefore, we repeatedly run the experiments 10 times for each dataset to ensure reproducible results. The final result is obtained by the average of 100 runs on independent test data. The area under the ROC curve (AUC) is selected as the metric in our work for robust and fair comparisons according to the previous literature [46–49].

Results based on 16S rRNA amplicon sequencing data

Comparisons with different DNN architectures

In order to further improve the prediction performance and illustrate the superiority of our deep learning model, we designed and compared different deep learning models based on the model architectures with different training tricks such as dropout, L2 regularization and batch normalization. The detailed model configurations are shown in Table 2. MLP is used as the benchmark classifier of deep learning. Compared with our RNN-based model (i.e. LSTM and GRU), the input of MLP is the same as traditional methods, which are the flattening vector of multivariate time series data. In addition, we keep all the other parameters such as learning rate and batch size to be the same as our RNN-based models for a fair comparison.

As we can see from Figure 4, our GRU network has remarkable improvement on seven of eight datasets comparing with the performance of the benchmarking classifier MLP, which proves the superiority of the RNN-based model for time series data.

Table 2. Model configurations for deep neural networks

Model	Architecture
MLP	Dense 1(512) → Dense 2(256)
LSTM	LSTM 1(512) → Dense 1(256)
GRU1	GRU 1(512) → Dense 1(256)
GRU2	BiGRU 1(512) → Dense 1(256)
GRU3	BiGRU 1(512) → BiGRU 2(256) → Dense 1(128)
GRU4	GRU3 with dropout after three layers (0.2, 0.5, 0.5)
GRU5	GRU4 with L2 regularization (0.0002 for all the layers)
GRU6	GRU3 with batch normalization (for all the layers)

Notes: MLP represents the multilayer perceptron; Dense represents the dense layer; LSTM means the basic one-directional LSTM layer; GRU means the basic one-directional GRU layer; BiGRU represents the bi-directional GRU, which is a combination of forward GRU and backward GRU. For backward GRU, the state of h_t will be determined by h_{t+1}. For all the tricks used to relieve the overfitting, we only add them after or on the layers listed.

The performance is also better than LSTM model. Comparing with the performance of GRU1 and GRU2, it is evident that the bi-directional GRU has distinct advantages since it integrates both the information at the previous time and the succeeding time. For the comparisons of the performance of GRU3, GRU4 and GRU5 with GRU2, the AUCs obviously increase on six of eight datasets. We can conclude that dropout and L2 regularization can effectively relieve the overfitting problems, thereby increasing the final prediction performance. Notably, there is a considerable amount of existing research proving that batch normalization has better generalization than dropout or L2 regularization. However, in experiments, the performance on seven of eight datasets dropped when we utilized batch normalization in GRU6 to replace dropout and L2 regularization in GRU3, GRU4 and GRU5, and it means that batch normalization may not be suitable for longitudinal microbiome data. We also noticed that the GRU performance on Kegg dataset decreases compared with the simple MLP. We attribute the overall bad performance of this task to the noisy features and feature irrelevance between the abundance data and labels, comparing to other tasks. Besides, we further discussed the potential of using transfer learning to conduct the fine-tuning based on the model trained on Knat dataset to test its generalization on small datasets. Although the performance is relatively not good, we cannot deny the potential of transfer learning in metagenomics. We think the reason is that the Knat dataset is still too small to extract the potential patterns from abundance times data for generalization although its size is already the biggest in our study. The comparison results of fine tuning can be seen in Supplementary Figure S7.

Comparisons with different feature extraction methods

Since the average performance of GRU4 is relative better on all the datasets, we further compare the performance of different kinds of feature extraction algorithms using GRU4 as the basic classifier (as Baseline in Figure 5). As we can see in Figure 5, for both single-clade simulation and two-clade simulation datasets, L1 achieves the best performance among all the feature selection algorithms. For real datasets, L1 performs the best on David and Kegg datasets; RF performs the best on Bdiet and Kige datasets; VT performs the best on Bdeliv dataset; PCA performs the best on Knat dataset. For other comparison algorithms, we noticed that the overall performance of supervised methods is slightly better than unsupervised methods, we consider the reason may be that the former utilizes label information. In addition, the best performance of L1 also supported that, since L1 is a supervised

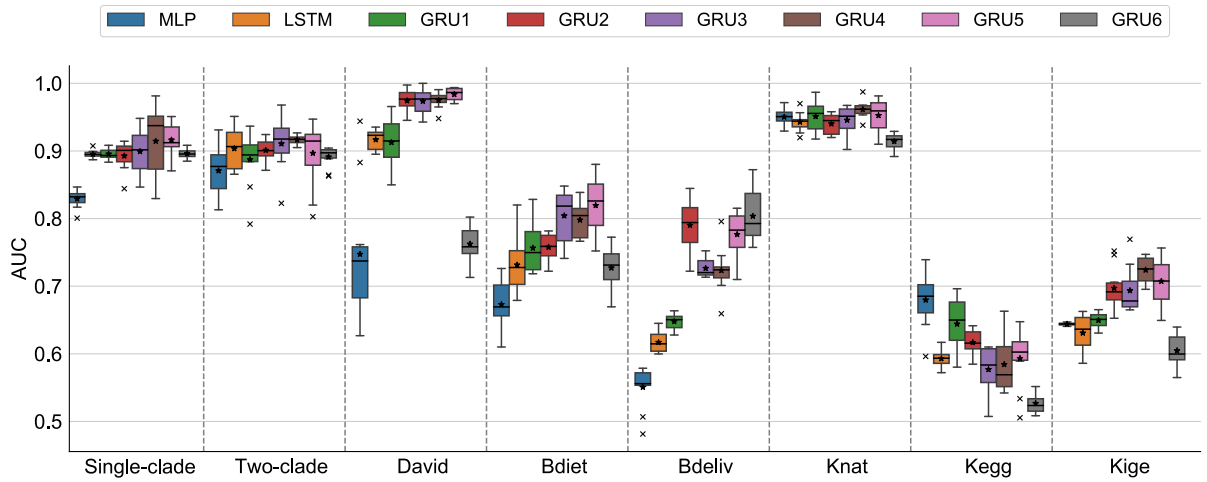

Figure 4. The comparisons of different deep learning model configurations on both simulation and real datasets. The detailed model architectures are shown in Table 2. The implementation details can be found in Supplementary Method 1.7.

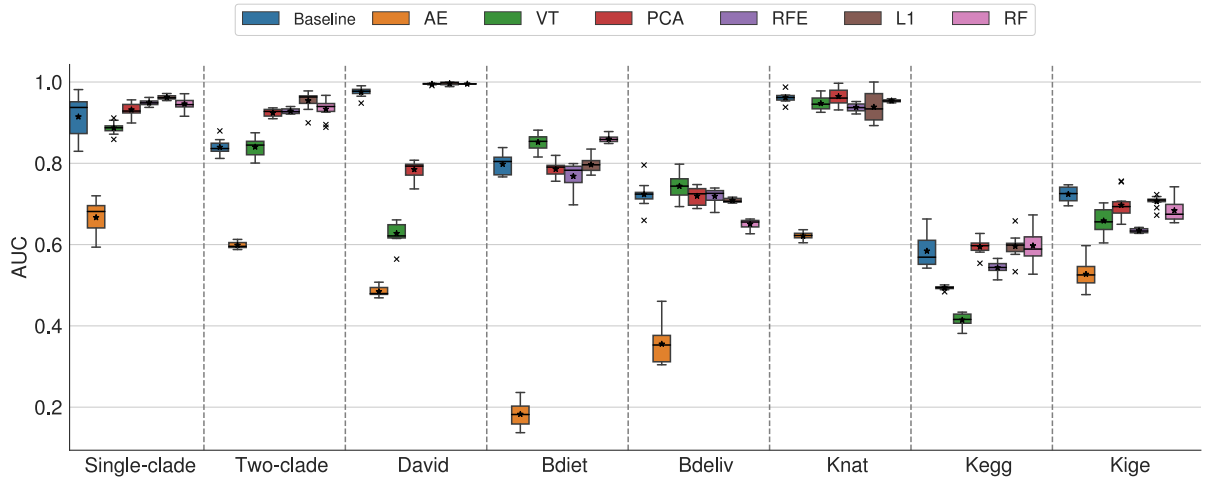

Figure 5. Comparison results with different feature extraction algorithms on both simulation and real datasets. All the feature extraction algorithms are based on GRU4 classification model. Baseline represents the results obtained by raw features without feature extraction; AE represents the auto encoder; VT represents the variance threshold; PCA represents the principal components analysis; RFE represents the recursive feature elimination; L1 represents the L1-based feature selection; RF represents the random forest. See Supplementary Method 1.7 for more implementation details.

method as well. AE achieves the worst AUCs on seven of eight datasets among all the comparison algorithms. Especially for Bdiet dataset, the average AUC is even less than 0.2. Although we utilized sparse AE and decreased the reconstructed loss as much as possible, the performance still cannot be guaranteed. The figures of the reconstructed losses for all the datasets can be seen in Supplementary Figure S8. We reason the bad performance of AE is an inherent issue with microbiome temporal data since the performance is not good on all datasets. We observe that, although AE can flexibly reduce the feature dimension, it can also lose the essential information that may be important for classification. In addition, since AE is an unsupervised method training without labels, it may be more suitable for dimension reduction and feature visualization rather than the prediction tasks. By comparing all the feature extraction methods, we can conclude that there is no best feature extraction method that can be generalized to all the studies. However, feature extraction does improve the prediction accuracy and can decrease the prediction time and information required from subjects for disease prediction.

Comparisons with recently published predictors and baseline classifiers

To evaluate the performance of our deep learning models, we further compare them with the latest predictor (MITRE), the deep learning model (RNN) proposed by [22] and a series of traditional baseline classifiers (SVM, KNN, LR and RF). The input data and the cross-validation standard are set to be consistent in all the predictors for strictly fair evaluations. For the inputs of traditional methods (SVM, KNN, LR and RF), we directly flatten the matrix to be a one-dimensional vector. In this way, the input explicitly contains the time information, thereby ensuring the fairness of the comparison. For all basic classifiers, we use grid search and internal cross-validation to choose the best parameters (e.g. for KNN, the k value was chosen between 2 and

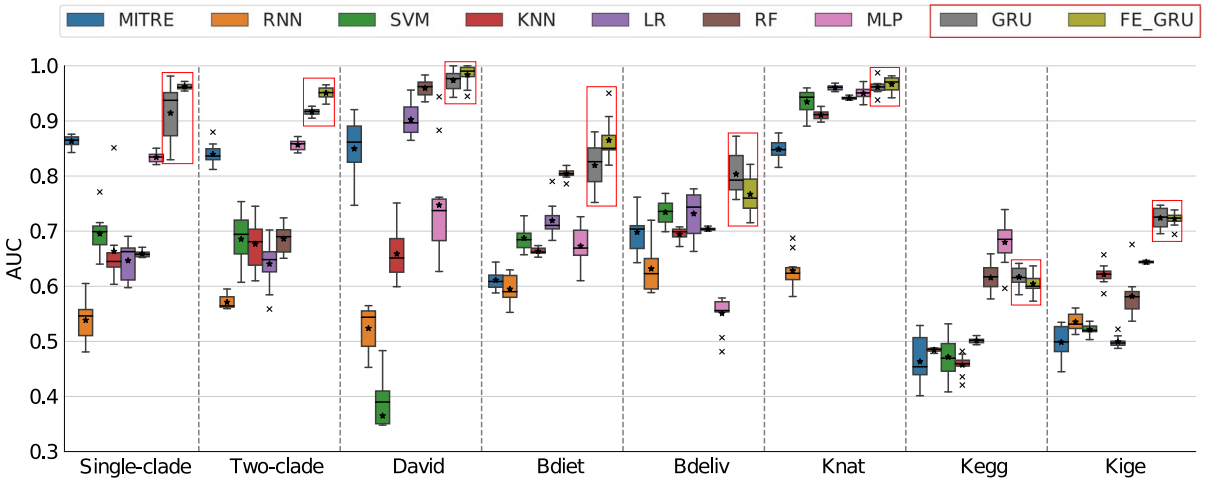

Figure 6. Comparison results recently published predictors and baseline classifiers on both simulation and real datasets. MITRE is based on Bayesian algorithm with rule-based prior knowledge. RNN is based on LSTM model proposed to predict the food allergy. SVM represents the support vector machine, KNN represents K-nearest neighbor, LR represents L1-regularized logistic regression, RF represents the random forest. The architecture of MLP is described in Table 2, GRU and FE_GRU represent the best results obtained by our GRU neural network and its combination with feature extraction, which are described in 3.3. See Supplementary Method 1.7 for more implementation details.

the sample size for minority class). The comparison results are shown in Figure 6.

As we can see that our deep learning models (GRU, FE_GRU), no matter before feature extraction or after feature extraction, practically outperform all the comparison predictors on both simulation and real datasets, especially on Kegg and Kige datasets. Although the final prediction results are relatively not higher than the performance on other datasets, SVM, KNN and LR basically cannot classify them. Thus, it can be concluded that the flattening vectors failed to distinguish the host status without temporal information. Especially, all of our deep learning methods outperform the recently published predictor MITRE on all the datasets, which further shows that our methods can capture the dynamic patterns among those classification tasks, consequently improving the prediction performance. Besides, our method also gives a substantial boost in the performance compared to the similar time series model RNN by efficient data preprocessing and model designs. For David dataset, the AUC of our model achieves perfect performance, demonstrating that our framework has not misclassified subject on this dataset. We attribute to the observation that the dataset for this study was sampled at more consistent and well-distributed time points, as evidenced by the distribution of the observations in Supplementary Figure S4. The performance of our methods on both simulation datasets (singleclade, two clade) is better than the real dataset (Bdeliv), although the original dataset and the new simulated datasets have a different number of subjects and time points. In addition, comparing with the improvements on real datasets, the increases on the simulation datasets tend to be more remarkable, which means that the generated data may be cleaner and the data distribution is evener than the raw dataset.

Results based on shotgun metagenomic sequencing and HMP data

The comparison results of different methods on shotgun metagenomic sequencing and HMP datasets can be seen from Supplementary Method 1.8. As we can see in Table S4, our FE_GRU still achieves the best prediction performance among all the comparison predictors on shotgun metagenomic and HMP datasets. Besides, it can be noticed that even for our single model GRU, comparing with the recently published method MITRE, we still have a 2%–10% improvement. In Table S5, we conducted different feature extraction methods based on GRU4 classifier. We observed that L1 still performs the overall best performance in all the feature extraction algorithms, and this also proved our previous conclusion that L1 may be the best feature extraction algorithms for OTU abundance data since the same results also appeared on 16S rRNA amplicon sequencing data. The prediction results indicate that the proposed pipeline can also be successfully extended to HMP and shotgun metagenomic data.

Execution time

The running time for different feature extraction algorithms and training GRU deep learning models depends on the size and complexity of the dataset, the number of epochs and the batch size for each study. The detailed running time statistics can be found in Supplementary Method 1.9. The data preprocessing time for David, Bokulich and Karelia studies takes 4 minutes, 3 minutes and 8 minutes, respectively; the running time for all the feature extraction algorithms is less than 1 minute for all the datasets; the running time for training the GRU model for David, Bokulich and Karelia studies takes about 5 minutes, 7 minutes and 10 minutes, respectively. Compared with the latest predictor MITRE, which costs about 45 to 65 h to classify the same datasets, or RNN proposed by [22], which is faster than MITRE but the performance cannot be guaranteed, our method has significantly reduced the prediction time with 1-100X faster while keeping considerable prediction performance.

DISCUSSION

In this study, we propose a deep learning-based framework to infer features from microbiome temporal data linked to its host status. To the best of our knowledge, there are few studies using data science to predict the host status in microbiome longitudi-

nal studies, especially for other host status predictions rather than the human disease. Our framework is the first systemic tool that integrates all the alternative steps such as data preprocessing, feature engineering, feature extraction and classifiers for human host status inference. For data preprocessing, we introduced a novel average imputation method to address the issue of missing data in microbiome longitudinal studies. For feature engineering, we applied a phylogenetic tree to embed the hierarchy relationship to enrich the microbiome feature representation. For feature extraction, we proposed pseudo-subject representation to select the best features on OTU-level data, which makes it possible for traditional feature extraction algorithms to keep the temporal information. Compared with other conventional machine learning algorithms and the recently published methods, our solution achieves the competitive and the best performance with taking the least running time. For existing studies in disease status inference, single-point-based methods can cause the loss of temporal information. Therefore, it is imperative to design a method to address this issue using deep learning algorithms. RNN has always been remarkable for its robust ability to process time series and the sequence data. Therefore, we conduct the classification experiments using the RNN-based model, which can take the observations at different time points as a whole OTU matrix, and find the temporal relationship among them.

We implemented a series of experiments on different studies, and the results demonstrate that our algorithm can be well generalized on different datasets without complicated parameter settings or optimizations on specific studies. Throughout the overall prediction results and the data distribution from different studies, we can see that the final prediction precision partly depends on the dataset integrity, and the data preprocessing is very important. For instance, the time distribution and the sampling intervals (Figure S4) for the study of [27] are the most even and perfect; thus, we can obtain the best prediction results (Figure 5) on this dataset. In contrast, for the study of [26], since the dataset is less relevant to certain classification tasks, and the sampling period is very long, as well as the inconsistent sampling, it is really difficult to improve the prediction precision based on the existing methods. There is a noticeable phenomenon that, for the classification tasks of Kegg and Kige, the traditional classifiers of interest basically cannot classify them. In contrast, our single GRU model can achieve relatively good prediction performance RNN architecture (with AUCs all more than 0.6). We reason that GRU architecture can keep the temporal information of the microbiome data, enabling the abundance features easy to be classified. For the performance of feature extraction methods, there is only a slight improvement comparing with the single GRU model. Therefore, we attribute the performance gain to the RNN while feature selection has a relatively low contribution. Furthermore, the comparison results with conventional classifiers demonstrated that our GRU model could capture the temporal relationship to infer dynamic patterns among those classification tasks, consequently achieving the best prediction performance. This can be instructive and meaningful to the microbiome longitudinal research, especially facing the extremely high-dimensional and noisy time series abundance data.

Admittedly, there are also some limitations to this study. Although our algorithm can achieve a relatively superior prediction performance, the black-box property of deep learning model cannot help us to obtain biological insights such as the feature importance offered by random forest. Since the most superior part for deep learning is to learn the features rather than directly generate the decision boundary from the raw features, it is hard to achieve a good trade-off between the model interpretability and prediction performance. Such a direction may be undoubtedly worthy of future investigations. In addition, the existing datasets in longitudinal microbiome studies are still small due to the sequencing cost. However, deep learning is well known to achieve better results with large amount of data. We believe that, in the future, the performance can be improved considerably through the use of large and rich datasets along with the development of sequencing technology.

CONCLUSION

We present that deep neural network is able to address the issue of predicting the human host status from the inconsistent time series abundance data. Comparing with conventional algorithms and the recently published predictors, our prediction model achieves the best prediction performance and the least prediction time. There are few relevant works on deep neural networks for host status prediction, especially for inconsistent time series abundance data. We believe that our algorithm can contribute to the microbiome longitudinal research and help to explore the dynamic patterns among the abundance time series data and its host status, paving the foundation for personalized medicine in the future.

Key Points

- This is the first comprehensive deep learning-based framework for human status inference from longitudinal microbiome data.
- The proposed novel data imputation and feature engineering can be a standard data preparation pipeline for longitudinal microbiome data.
- The tailored pseudo-subject representation makes traditional feature selection methods applicable to longitudinal microbiome data.
- The framework demonstrates a good trade-off between the prediction precision and running time. On top of that, it can be generalized to different microbiome contexts.

AVAILABILITY

Datasets and source code are publicly available at https://github.com/Microbiods/Meta_GRU.

SUPPLEMENTARY INFORMATION

Supplementary materials are available online at *Briefings in Bioinformatics*.

ACKNOWLEDGMENTS

The two anonymous reviewers are thanked for their time and efforts, improving numerous aspects of the current study.

FUNDING

Research Grants Council of the Hong Kong Special Administrative Region (CityU 11200218); Health and Medical Research Fund; Food and Health Bureau; The Government

of the Hong Kong Special Administrative Region (07181426); Hong Kong Institute for Data Science at City University of Hong Kong; City University of Hong Kong (CityU 11202219, CityU 11203520); National Natural Science Foundation of China (32000464); Shenzhen Research Institute, City University of Hong Kong.

References

1. AE Livanos, TU Greiner, P Vangay, *et al*. Antibiotic-mediated gut microbiome perturbation accelerates development of type 1 diabetes in mice. *Nat Microbiol*, **1**(11): 16140, 2016.
2. YJ Huang, BJ Marsland, S Bunyavanich, *et al*. The microbiome in allergic disease: current understanding and future opportunities—2017 PRACTALL document of the American Academy of Allergy, Asthma & Immunology and the European Academy of Allergy and Clinical Immunology. *J Allergy Clin Immunol*, **139**(4): 1099–110, 2017.
3. SF Rahman, MR Olm, MJ Morowitz, *et al*. Machine learning leveraging genomes from metagenomes identifies influential antibiotic resistance genes in the infant gut microbiome. *MSystems*, **3**(1): e00123–17, 2018.
4. Zhou Y-H, Gallins P. A review and tutorial of machine learning methods for microbiome host trait prediction. *Front Genet* 2019; **10**:579.
5. Maltecca C, Lu D, Schillebeeckx C, *et al*. Predicting growth and carcass traits in swine using microbiome data and machine learning algorithms. *Sci Rep* 2019; **9**(1): 6574.
6. S Schmitt, P Tsai, J Bell, *et al*. Assessing the complex sponge microbiota: core, variable and species-specific bacterial communities in marine sponges. *ISME J*, **6**(3): 564–76, 2012.
7. Callahan BJ, McMurdie PJ, Holmes SP. Exact sequence variants should replace operational taxonomic units in marker-gene data analysis. *ISME J* 2017; **11**(12): 2639–43.
8. Caruso V, Song X, Asquith M, *et al*. Performance of microbiome sequence inference methods in environments with varying biomass. *MSystems* 2019; **4**(1).
9. LaPierre N, Chelsea J-TJ, Zhou G, *et al*. A critical evaluation of deep learning and machine learning in metagenome-based disease prediction. *Methods* 2019; **166**:74–82.
10. Gilbert JA, Blaser MJ, Caporaso JG, *et al*. Current understanding of the human microbiome. *Nat Med* 2018; **24**(4): 392–400.
11. Bolyen E, Rideout JR, Dillon MR, *et al*. Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2. *Nat Biotechnol* 2019;**37**(8):852–7.
12. P Vangay, BM Hillmann, D Knights. Microbiome Learning Repo (ML Repo): a public repository of microbiome regression and classification tasks. *GigaScience*, **8**(5), 2019. giz042.
13. E Pasolli, DT Truong, F Malik, *et al*. Machine learning meta-analysis of large metagenomic datasets: tools and biological insights. *PLoS Comput Biol*, **12**(7), 2016.
14. Zhu Q, Jiang X, Zhu Q, *et al*. Graph embedding deep learning guide microbial biomarkers' identification. *Front Genet* 2019; **10**:1182.
15. Oudah M, Henschel A. Taxonomy-aware feature engineering for microbiome classification. *BMC Bioinform* 2018; **19**(1): 227.
16. Ditzler G, Morrison JC, Lan Y, *et al*. Fizzy: feature subset selection for metagenomics. *BMC Bioinform* 2015; **16**(1): 358.
17. Q Zhu, M Pan, L Liu, *et al*. An ensemble feature selection method based on deep forest for microbiome-wide association studies. In: *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 248–53. IEEE, 2018.
18. HR Johnson, DD Trinidad, S Guzman, *et al*. A machine learning approach for using the postmortem skin microbiome to estimate the postmortem interval. *PloS One*, **11**(12), 2016.
19. Ditzler G, Polikar R, Rosen G. Multi-layer and recursive neural networks for metagenomic classification. *IEEE Trans Nanobiosci* 2015; **14**(6): 608–16.
20. D Reiman, A Metwally, Y Dai. Using convolutional neural networks to explore the microbiome. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4269–72. IEEE, 2017.
21. Bogart E, Creswell R, Gerber GK. Mitre: inferring features from microbiota time-series data linked to host status. *Genome Biol* 2019; **20**(1): 186.
22. Metwally AA, Yang J, Ascoli C, *et al*. Metalonda: a flexible r package for identifying time intervals of differentially abundant features in metagenomic longitudinal studies. *Microbiome* 2018; **6**(1): 32.
23. JNPaulson, OC Stine, HC Bravo, *et al*. Differential abundance analysis for microbial marker-gene surveys. *Nat Methods*, **10**(12): 1200–2, 2013.
24. D Luo, S Ziebell, L An. An informative approach on differential abundance analysis for time-course metagenomic sequencing data. Bioinformatics, **33**(9): 1286–92, 2017.
25. K Cho, B Van Merriënboer, D Bahdanau, *et al*. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches, Proceedings of SSST–8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation. 2014:103–11.
26. NA Bokulich, J Chung, T Battaglia, *et al*. Antibiotics, birth mode, and diet shape microbiome maturation during early life. *Sci Transl Med*, **8**(343): 343ra82–343ra82, 2016.
27. LA David, CF Maurice, RN Carmody„ *et al*. Diet rapidly and reproducibly alters the human gut microbiome. *Nature*, **505**(7484): 559–63, 2014.
28. T Vatanen, AD Kostic, E d'Hennezel, *et al*. Variation in microbiome LPS immunogenicity contributes to autoimmunity in humans. *Cell*, **165**(4): 842–53, 2016.
29. E Pasolli, L Schiffer, P Manghi„ *et al*. Accessible, curated metagenomic data through experimenthub. *Nat Methods*, **14**(11): 1023, 2017.
30. Brooks B, Olm MR, Firek BA, *et al*. Strain-resolved analysis of hospital rooms and infants reveals overlap between the human and room microbiome. *Nat Commun* 2017; **8**(1): 1–7.
31. AB Hall, M Yassour, J Sauk„ *et al*. A novel ruminococcus gnavus clade enriched in inflammatory bowel disease patients. *Genome Med*, **9**(1): 1–12, 2017.
32. A Heintz-Buschart, P May, CC Laczny„ *et al*. Integrated multi-omics of the human gut microbiome in a case study of familial type 1 diabetes. *Nat Microbiol*, **2**(1): 1–13, 2016.
33. Raymond F, Ouameur AA, Déraspe M, *et al*. The initial state of the human gut microbiome determines its reshaping by antibiotics. *ISME J* 2016; **10**(3): 707–20.
34. Vincent C, Miller MA, Edens TJ, *et al*. Bloom and bust: intestinal microbiota dynamics in response to hospital exposures and *Clostridium difficile* colonization or infection. *Microbiome* 2016; **4**(1): 1–11.
35. Shao Y, Forster SC, Tsaliki E, *et al*. Stunted microbiota and opportunistic pathogen colonization in caesarean-section birth. *Nature* 2019; **574**(7776): 117–21.
36. Integrative HMP. 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.
37. Matsen FA, Kodner RB, Armbrust EV. pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree. *BMC Bioinform* 2010; **11**(1): 538.
38. Callahan BJ, McMurdie PJ, Rosen MJ, *et al*. Dada2: high-resolution sample inference from illumina amplicon data. *Nat Methods* 2016; **13**(7): 581–3.
39. PD Schloss, SL Westcott, T Ryabin„ *et al*. Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl Environ Microbiol*, **75**(23): 7537–41, 2009.
40. K Qu, F Gao, F Guo, *et al*. Taxonomy dimension reduction for colorectal cancer prediction. *Comput Biol Chem*, **83**:107160, 2019.
41. Che Z, Purushotham S, Cho K, *et al*. Recurrent neural networks for multivariate time series with missing values. *Sci Rep* 2018; **8**(1): 1–12.
42. Karim F, Majumdar S, Darabi H, *et al*. LSTM fully convolutional networks for time series classification. *IEEE Access* 2017; **6**:1662–9.
43. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* 1997; **9**(8): 1735–80.
44. K Cho, B Van Merriënboer, C Gulcehre, *et al*. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014:1724–34.
45. Pedregosa F, Varoquaux G, Gramfort A, *et al*. Scikit-learn: machine learning in python. *J Mach Learn Res* 2011; **12**:2825–30.
46. M Yazdani, BC Taylor, JW Debelius, *et al*. Using machine learning to identify major shifts in human gut microbiome protein family abundance in disease. In *2016 IEEE International Conference on Big Data (Big Data)*, pp. 1272–80. IEEE, 2016.
47. Honglong Wu, Lihua Cai, Dongfang Li, *et al*. Metagenomics biomarkers selected for prediction of three different diseases in Chinese population. *BioMed Res Int*, 2018.
48. Nakano Y, Suzuki N, Kuwata F. Predicting oral malodour based on the microbiota in saliva samples using a deep learning approach. *BMC Oral Health* 2018; **18**(1): 128.
49. E Asgari, K Garakani, AC McHardy, et al. Micropheno: predicting environments and host phenotypes from 16s RRNA gene sequencing using a k-mer based representation of shallow sub-samples. *Bioinformatics*, **34**(13): i32–i42, 2018.