DeepMicroGen: a generative adversarial network-based method for longitudinal microbiome data imputation

Joung Min Choi[1], Ming Ji[2], Layne T. Watson[3], Liqing Zhang[1,*]

BIBS Seminar
24/02/01
Hanbyul Song

서울대학교 통계학과
생물정보통계연구실
BIBS

Bioinformatics, 2023, 39(5), btad286
https://doi.org/10.1093/bioinformatics/btad286
Advance access publication 26 April 2023
Original Paper

OXFORD

Genome analysis

DeepMicroGen: a generative adversarial network-based method for longitudinal microbiome data imputation

Joung Min Choi [1], **Ming Ji**[2], **Layne T. Watson**[3], **Liqing Zhang**[1,*]

[1]Department of Computer Science, Virginia Tech, Blacksburg, VA 24060, United States
[2]College of Nursing, University of South Florida, Tampa, FL 33620, United States
[3]Departments of Computer Science, Mathematics, and Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA 24060, United States

*Corresponding author. Department of Computer Science, Virginia Tech, Blacksburg, VA 24060, USA. E-mail: lqzhang@cs.vt.edu (L.Z.)

Associate Editor: Valentina Boeva

Contents

Introduction

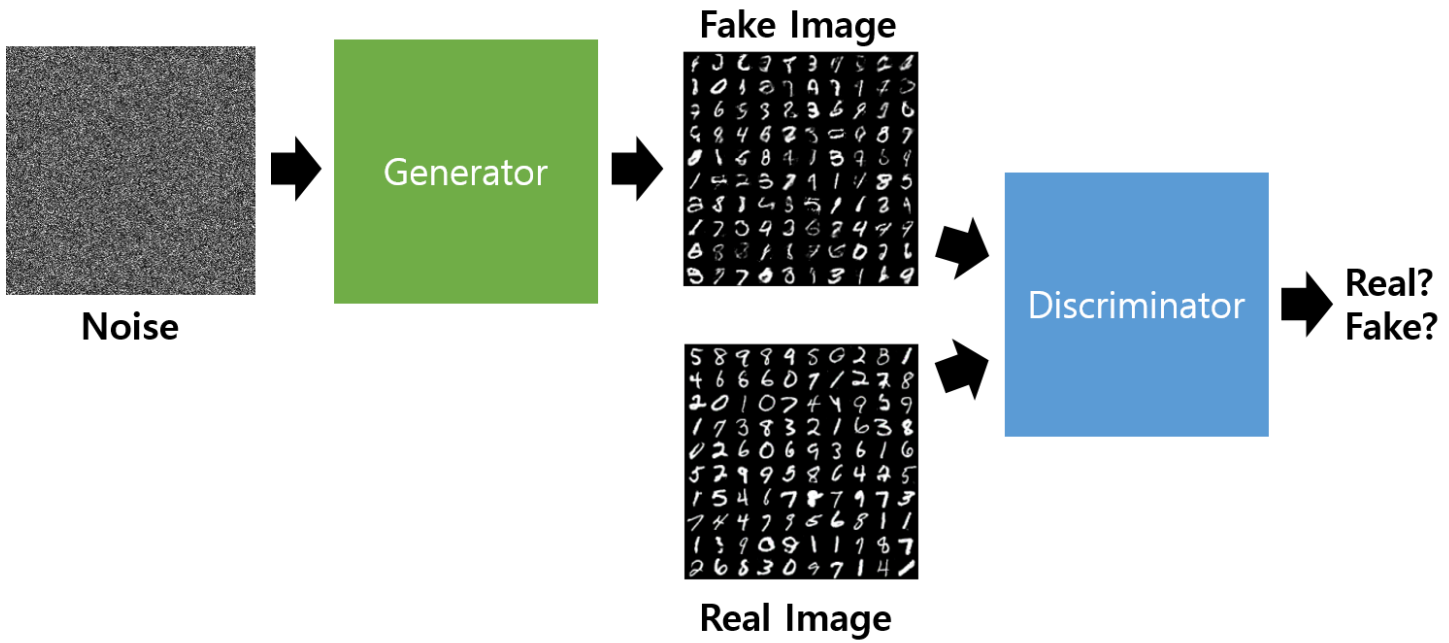
Introduction: Background

- **Microbiome Research in Bioinformatics:**
  - Explores microorganisms in environments for new medical treatments.
  - Crucial in understanding and treating complex diseases like diabetes, cancer, and allergies.
  - Longitudinal studies reveal microbiome dynamics, aiding disease diagnosis and treatment.

- **Challenge in Longitudinal Microbiome Studies:**
  - Uneven and varying number of time points for different subjects.
  - Comprehensive analysis is difficult due to missing samples at some timepoints.
    → Results in a significant amount of unusable data (Ridenhour et al. 2017).

Introduction: Existing methods

- **Use of Generative Adversarial Networks (GANs):**
  - GANs proposed to address missing data in longitudinal studies.
  - Widely adopted in fields like image synthesis and text generation.
  - Effective for data augmentation, reducing overfitting in prediction/classification tasks.



https://hyeongminlee.github.io/post/gan001_gan/

Introduction: Existing methods

- **Innovations in GAN Applications:**
  - GAN combined with **Recurrent Neural Network (RNN)** for imputing missing multivariate time series data.
  - Luo et al. (2018, 2019) used **Gated Recurrent Unit (GRU)** in GAN for modeling temporal irregularities and reconstructing incomplete datasets.
  - Gupta and Beheshti (2020), Zhang et al. (2021) introduced **bidirectional RNN** and **GRU** with GAN for predictive classification and regression tasks.

Introduction: Existing methods

- **GAN-Based Approaches in Microbiome Data:**
  - **MB-GAN** for learning latent spaces and generating simulated microbial abundances (Rong et al. 2021).
  - **DeepBioGen** for generating realistic human gut microbiome profiles and generalizing classifiers for type 2 diabetes (Oh and Zhang 2021).

  - Current methods focus on single time point data augmentation; longitudinal data imputation remains unaddressed.

Introduction: Proposed method

- **Method Overview:**
  - **DeepMicroGen** is a deep generative method specifically designed for longitudinal microbiome data.
  - Utilizes multiple operational taxonomic units (OTUs) from the input dataset.

- **Technical Features:**
  - Extracts features incorporating phylogenetic relationships between taxonomies.
  - Uses **Convolutional Neural Network (CNN)** modules for feature extraction.
  - Employs a **bidirectional RNN-based GAN** model.

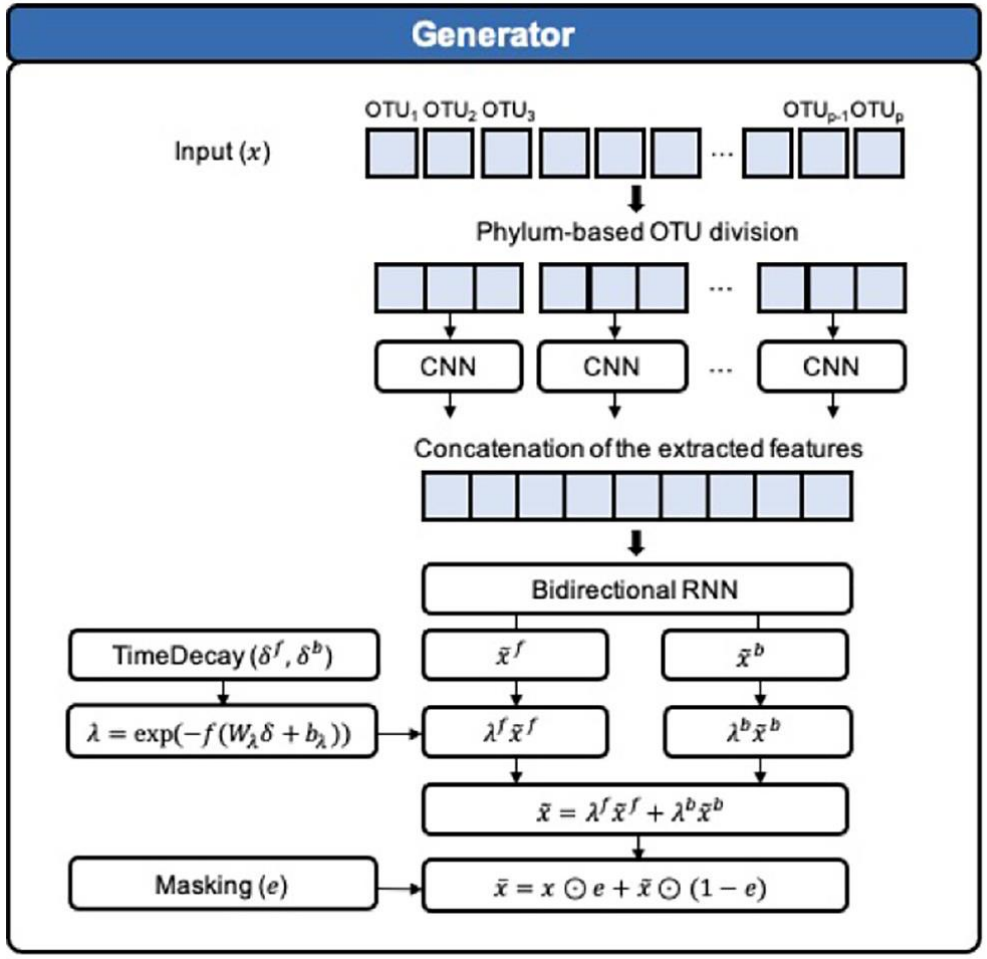
Introduction: Proposed method

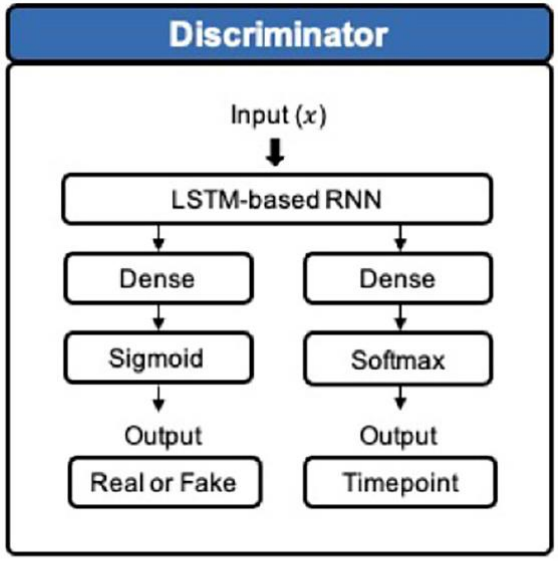
- **Functionality:**
  - Generates imputed values by learning temporal dependencies between observations at different time points.

- **Performance and Advantages:**
  - Demonstrates the lowest mean absolute error (MAE) compared to standard baseline methods.
  - Shows improved performance in **simulated** and **real datasets.**
  - Enhances prediction performance for allergy outcomes by providing a complete longitudinal dataset through imputation**.**

Introduction: Proposed method

- **Illustration of DeepMicroGen**





$$Loss_{discriminator} = loss_D + loss_T$$

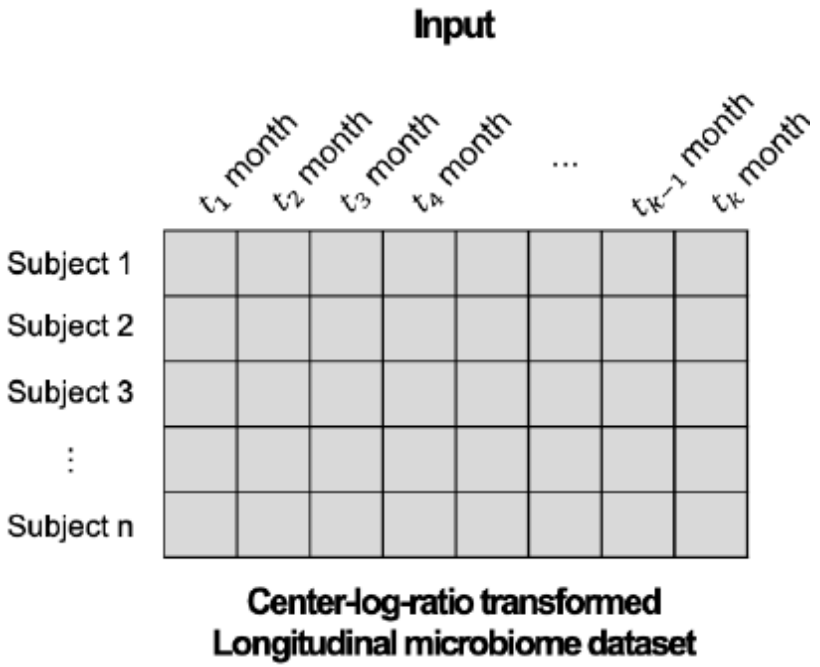
$$Loss_{generator} = loss_G + loss_R + loss_C$$

Materials and Methods

Materials and Methods: Preprocessing of longitudinal microbiome data

- **Input Data:**
  - Uses species-level relative abundance (RA) profiles.
  - RA profiles consist of real values in the range [0,1], representing species percentages of the total observed species.

- **Data Transformation:**
  - Added a **pseudo-count** to zero values (minimum RA divided by two).
  - Applied **centered log ratio (clr)** transformation.
    → To account for compositionality.



Materials and Methods: Data representation and preparation

- **Taxa Number (n):**
  - Represents the count of different taxa in the dataset.
- **Time Sequence $\left(T = (t_1, t_2, \ldots, t_k)\right)$:**
  - A sequence of time points, increasing in value, capturing the temporal aspect of the data.

- **Observations $(x_i$ in $R^n)$:**
  - These are the real-valued observations at each time t_i.
    → Where $x_i = 0$ indicates a missing observation at time t_i.
- **Matrix Representation $(x = (x_1, \ldots, x_k) \in R^{n*k})$:**
  - Forms a matrix of all observations x_i, enabling analysis across multiple time points.

Materials and Methods: Data representation and preparation

- **Mask ($e \in \{0,1\}^k$):** A binary mask where $e_i = 0$ if x_i is missing and 1 otherwise.
  → This helps in identifying missing data points.

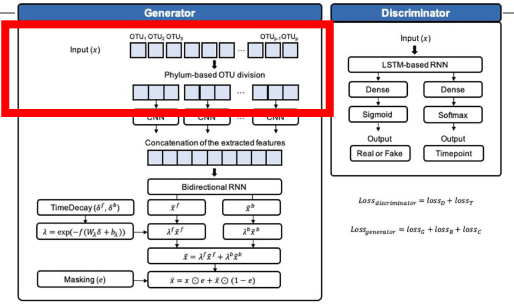
$$e_i = \begin{cases} 0, & \text{if } x_i \text{ is missing,} \\ 1, & \text{otherwise,} \end{cases}$$

- **Time Gap Vectors ($\delta^f \in R^k$):** Represent the time lag between the current and previous values in the forward direction.

$$\delta_i^f = \begin{cases} t_i - t_{i-1}, & \text{if } e_{i-1} = 1,\ i > 1, \\ \delta_{i-1}^f + t_i - t_{i-1}, & \text{if } e_{i-1} = 0,\ i > 1, \\ 0, & \text{if } i = 1. \end{cases}$$

- δ^b **(Backward Time Gap):** Similar to δ^f but calculated in the reverse direction, reflecting time differences with the next observed value.

Materials and Methods: CNN module

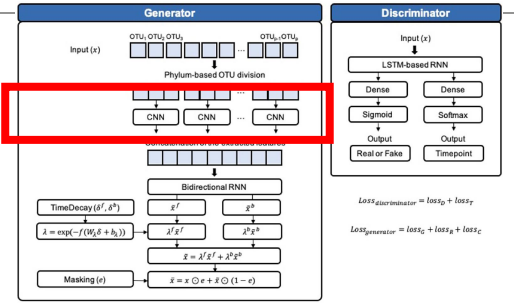


- **Capture Phylogenetic Relationship:**
  - ➢ Features are extracted to represent the evolutionary relationships among taxa using CNN modules.
- **Clustering and Correlation Measurement:**
  - ➢ OTUs are divided into clusters based on phylum. Within each cluster, the Spearman rank correlation between OTUs is computed, forming a matrix.
- **Geometric Mean Calculation:**
  - ➢ For each row in the Spearman rank coefficient matrix, the geometric mean of correlation coefficients is calculated using the formula:

$$\rho_{\mathrm{OTU}_j} = \sqrt[p]{\left|\rho_{\mathrm{OTU}_{j1}} \cdot \rho_{\mathrm{OTU}_{j2}} \cdot \ldots \cdot \rho_{\mathrm{OTU}_{jp}}\right|}, \quad 1 \leq j \leq p,$$

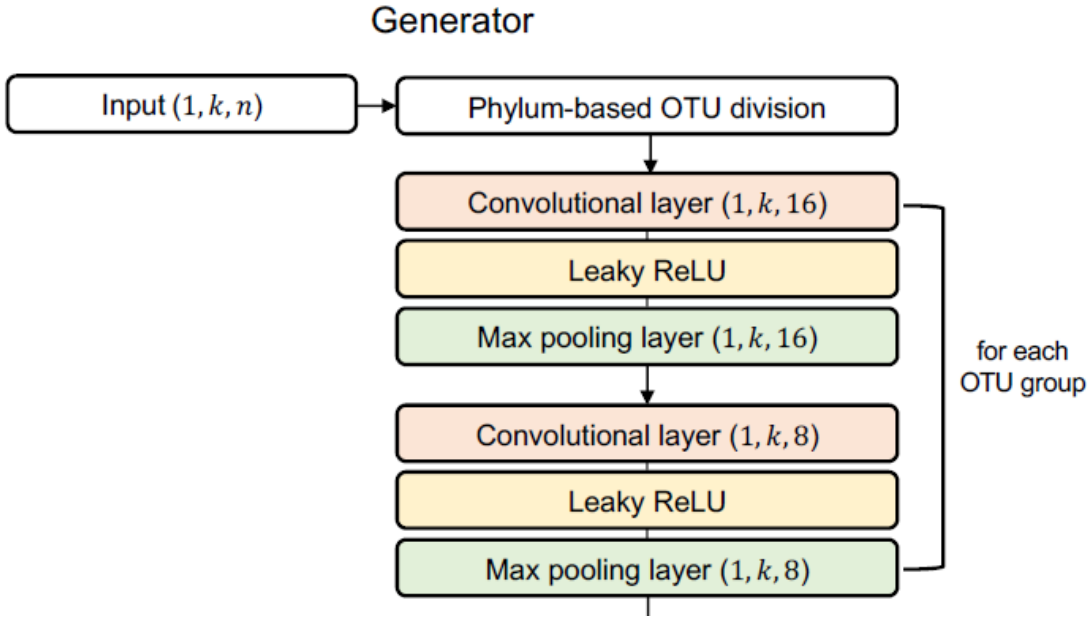
→ Where $\rho_{OTU_{jp}}$ is the Spearman correlation between OTU_j and OTU_p.

Materials and Methods: CNN module

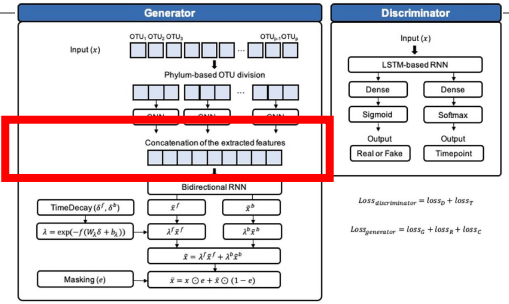


- **Sorting for CNN Input:** OTUs are sorted by the geometric mean of correlation coefficients, setting up for effective feature capture by the CNN.

- **CNN Module Configuration:** Comprises two 1D-CNN layers with specific kernel sizes and filters, each followed by Leaky ReLU activation and a max-pooling layer.
  → Extracted features from each cluster were concatenated and transferred to the biRNN module



Materials and Methods: Bi-directional RNN module



- **Functionality:** Captures both forward and backward temporal relations between observations.

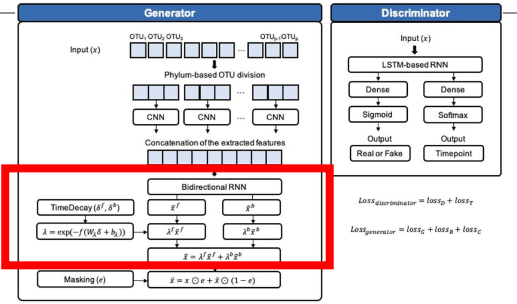
- **RNN Configuration:** Utilizes a one-layer biRNN with tanh activation and a fully connected layer.

- **RNN Cell Definition:** The RNN cell is defined by the equation:

$$h_i = \tanh(W_h h_{i-1} + W'_h x_i + b_h)$$

→ Where W_h and W'_h are weight matrices, b_h is the bias, and h_{i-1} is the previous time point's hidden state

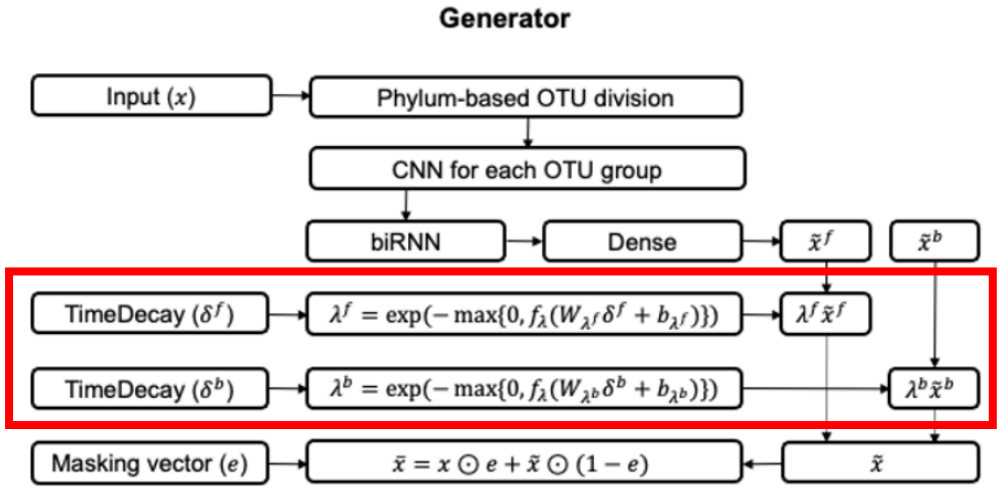
Materials and Methods: Output Generation and Combination



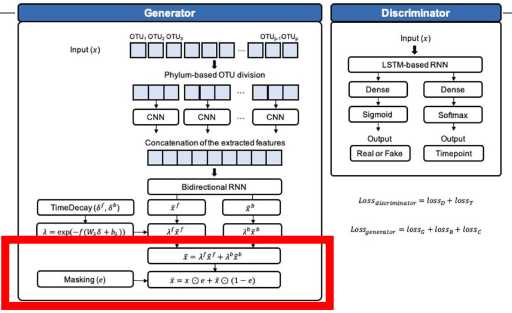
- **Forward and Backward Outputs:** Two outputs ($\tilde{x}_f, \tilde{x}_b$) are generated, representing imputed values from both directions.
- The outputs are weighted by combination factors λ_f, λ_b calculated as:

$$\lambda^f = \exp(-f_\lambda(W_{\lambda^f}\delta^f + b_{\lambda^f})),$$
$$\lambda^b = \exp(-f_\lambda(W_{\lambda^b}\delta^b + b_{\lambda^b})),$$

→ These factors, based on time gaps, modulate the influence of the forward and backward outputs.



Materials and Methods: Handling actual and imputed values



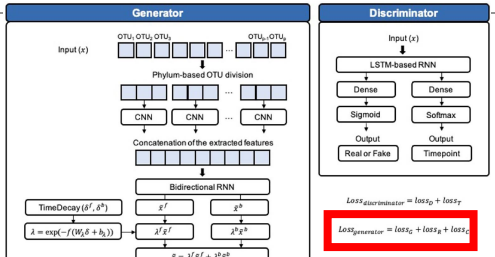
- **Final Output:**
  - The final output matrix x is calculated as a weighted sum of the forward and backward outputs: $\bar{x} = \delta^f \tilde{x}^f + \lambda^b \tilde{x}^b$.
  - This matrix is of size $\mathrm{n} * k$, corresponding to the number of taxa and time points.

- If actual values exist in the input data, they replace the corresponding generated output in the final imputation.

- The final imputation output x is calculated as:

$$\bar{x} = x \odot e + \tilde{x} \odot (1 - e)$$

Materials and Methods: Generator and losses



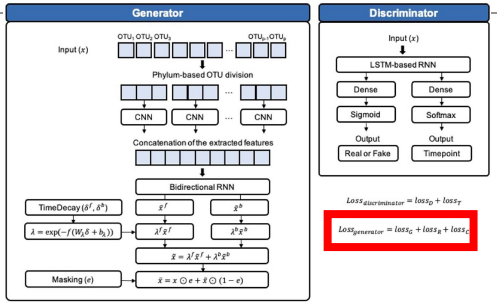
- During training phase, the generator was trained to minimize the loss composed of three different losses:

$$\text{loss}_{\text{generator}} = \text{loss}_G + \text{loss}_R + \text{loss}_C$$

- **Classification Loss ($loss_G$):**
  - Represents the classification loss where the generator is trained to maximize the probability $D(\bar{x} \odot (1 - e))$ that the discriminator D classifies the fake instances as actual values.

$$\text{loss}_G = -\log(D(\bar{x} \odot (1 - e)))$$

Materials and Methods: Generator and losses



- **Reconstruction Loss ($loss_R$):**
  - Aims to ensure the generated output $(\tilde{x}_i)$ is close to the actual values (x_i).

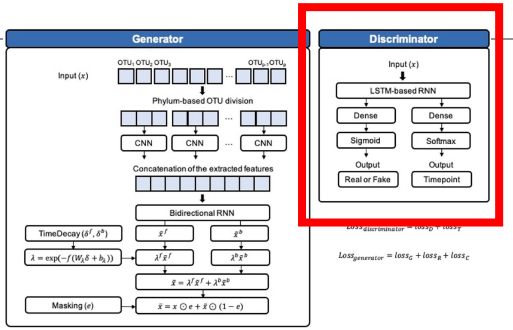
$$\text{loss}_R = \sum_{i=1}^{k} \|(x_{\cdot i} - \tilde{x}_{\cdot i})e_i\|_1 / ||e||_1$$

  - Calculated as the sum of absolute errors between actual and generated values across all time points:

- **Consistency Loss ($loss_C$):**
  - Minimizes the difference between the imputed outputs from the forward and backward directions of the biRNN.

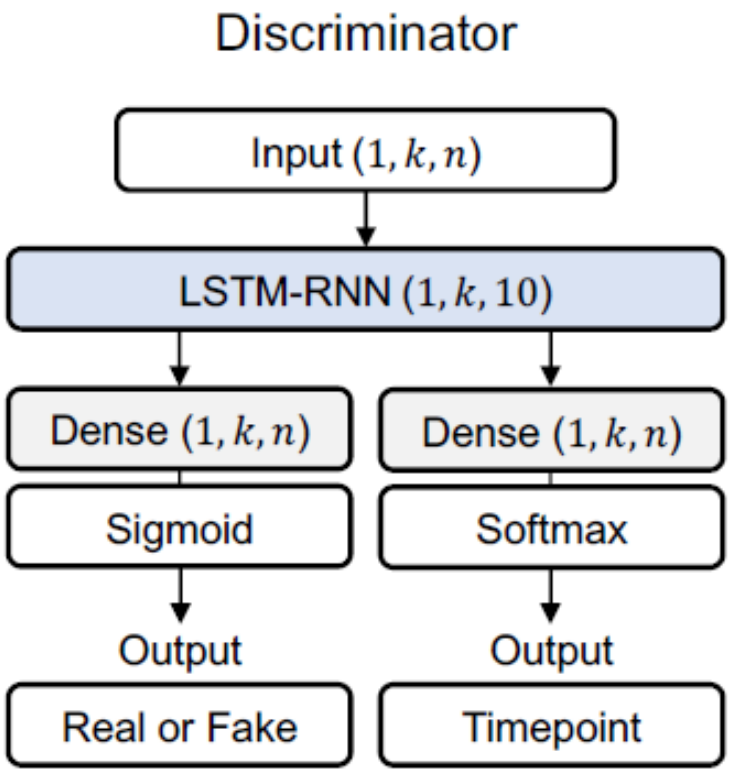
$$\text{loss}_C = \frac{1}{k} \sum_{i=1}^{k} \|\tilde{x}^f_{\cdot i} - \tilde{x}^b_{\cdot i}\|_1$$

Materials and Methods: Discriminator architecture and function

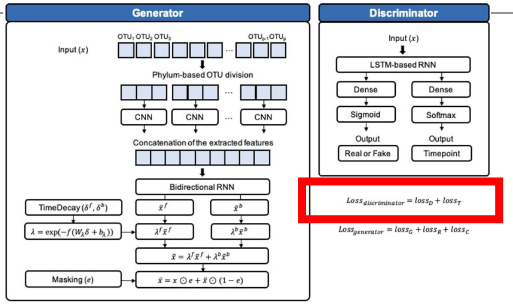


- **Structure of Discriminator (D):**
  - ➢ One-layer RNN module
  - ➢ long short-term memory (LSTM) cells (10 units)
  - ➢ tanh (hyperbolic tangent) activation function.

- **Output Processing and Tasks:**
  - Two separate feed-forward neural networks take RNN outputs for different tasks:
  - ➢ **First Network Tasks:** Classifies inputs as real or generated (sigmoid function).
  - ➢ **Second Network Tasks:** Predicts the time point for each value (softmax function).



Materials and Methods: Discriminator architecture and function



- The discriminator is trained to optimize two losses combined as:

$$\text{loss}_{\text{discriminator}} = \text{loss}_D + \text{loss}_T$$

- **Binary cross-entropy loss ($loss_D$):**
  - Measures the discriminator's ability to correctly identify actual and generated values.

$$\text{loss}_D = -\log(D(\overline{x} \odot e)) - \log(1 - D(\overline{x} \odot (1 - e)))$$

- **Cross entropy loss ($loss_T$):**
  - Measures the discriminator's ability to correctly predict the time point of each sample

$$\text{loss}_T = -\sum_{i=1}^{k} y(i) \log(\hat{y}(i)),$$

Materials and Methods: Training and optimization of DeepMicroGen

- DeepMicroGen was trained with the adaptive optimization algorithm **Adam** with a **learning rate** of 10^{-3}.
- Implemented a **dropout rate** of 0.7 in CNN layers to enhance model robustness.
- DeepMicroGen was built using the Tensorflow library (Version 1.8.0).

Results

Results: Real and simulated datasets

- **Real Study Datasets:**
  - **DIABIMMUNE dataset (16S rRNA):** 1064 samples from 133 subjects across Finland, Estonia, and Russia, focusing on type 1 diabetes and allergies. Samples collected at specific months post-birth.
  - **BONUS-CF dataset (WGS):** 452 samples from 113 subjects with cystic fibrosis. Includes whole genome shotgun sequencing data.

- **Simulated Datasets:**
  - Based on **DIABIMMUNE dataset**, simulating 200 subjects. Added noise to OTUs to create variability.
  - Followed the longitudinal microbiome simulation approach from Sharma and Xu (2021) & ensured total RAs add up to one in simulations.

Results: Imputation methods for comparison

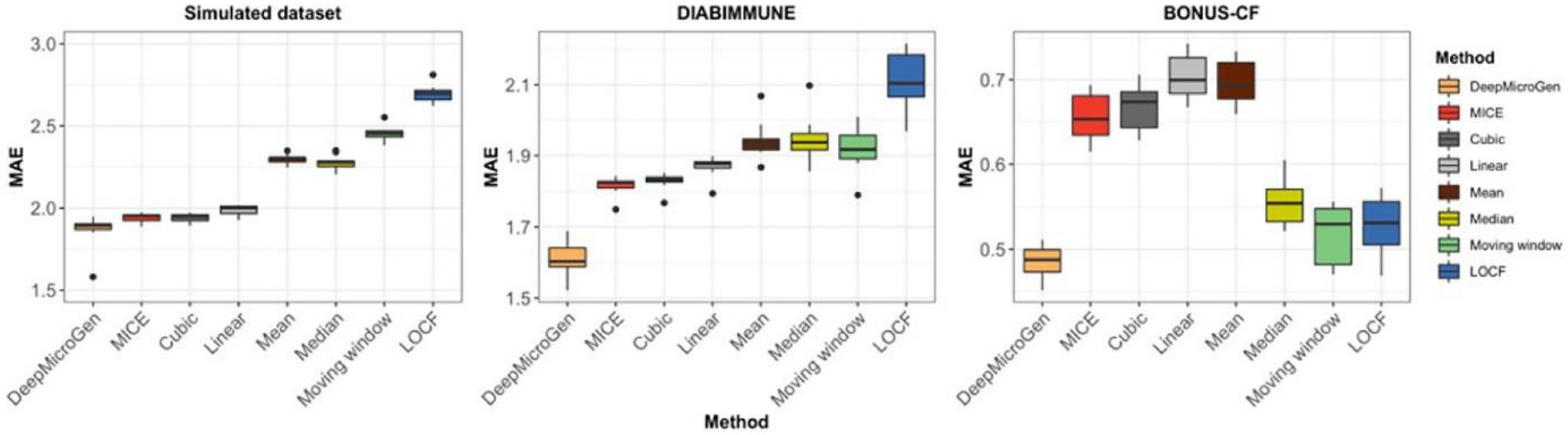
- **Imputation Methods for Comparison:**

  - **Simple imputation:** Mean, Median**.**
  - **Time-series imputation:** Linear curve fitting, cubic curve fitting, moving-window-based (window-size=3).
  - **Longitudinal dataset methods:** Multiple imputation by chained equations (MICE), last observation carried forward (LOCF).

- **Performance Evaluation:**
  - **Mean Absolute Error (MAE)** for missing samples using clr-transformed real and imputed values.

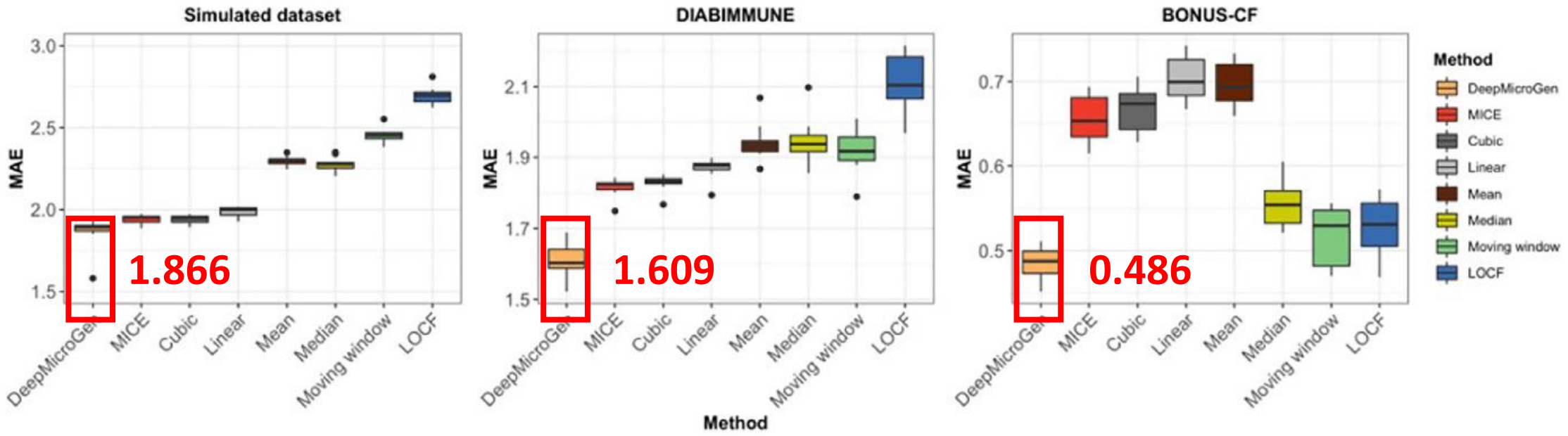
Results: Imputation evaluation

- Performance comparison of DeepMicroGen with the baseline methods based on 10-fold cross validation.



Results: Imputation evaluation

- Performance comparison of DeepMicroGen with the baseline methods based on 10-fold cross validation.



Results: Effectiveness of each component in DeepMicroGen

- To see the effect of different components on imputation performance, evaluated the different component of DeepMicroGen
  - **RNN** (eliminates both the discriminator and the CNN-based feature extraction)
  - **biGAN** (removes the CNN-based feature extraction)
  - **AE+biGAN** (replaces the CNN with the autoencoder)
  - **MDeep+biGAN** (replaces the CNN with the features encoding the phylogenetic correlation based on the method presented in MDeep)
- Performed 10-fold cross-validation using the **DIABIMMUNE dataset** and the average MAE was measured for performance evaluation .

Results: Effectiveness of each component in DeepMicroGen

- Average MAE results under different neural network architectures for imputation performing 10-fold cross-validation

Dataset	RNN	biGAN	MDeep+biGAN	AE+biGAN	DeepMicroGen
DIABIMMUNE	1.672	1.662	1.884	2.759	**1.609**
BONUS-CF	0.535	0.521	0.559	0.732	**0.474**

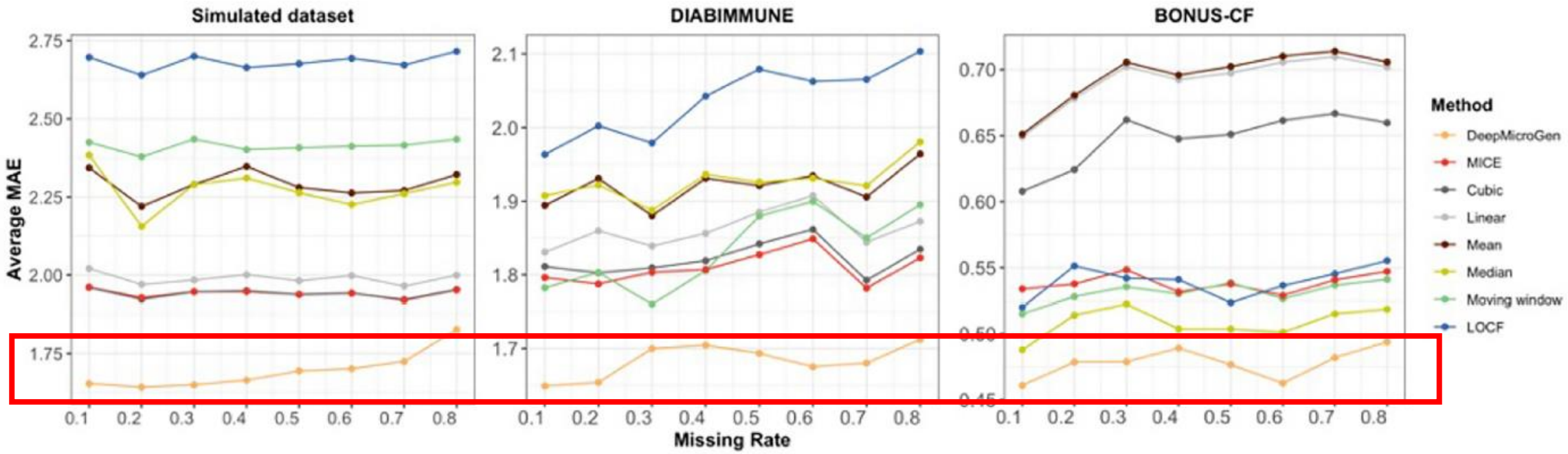
The best performance value for each experiment was bolded.

Results: Effect of different data missing rates and mechanisms

- To examine how the imputation performance changes with different **missing rates**, they randomly discarded **10%–80%** of the samples considering them as missing and performed data imputation using DeepMicroGen.
  - ➢ The experiment was repeated five times, and the average MAE was calculated.
- To investigate whether DeepMicroGen could outperform the other methods for all cases, the performance of the baseline methods was also measured.

Results: Effect of different data missing rates and mechanisms

- Imputation performance results with different missing rates for DeepMicroGen and other baseline methods



Results: Effect of different data missing rates and mechanisms

- Imputation performance results based on the **average MAE** for different missing data mechanisms using **DIABIMMUNE** and **BONUS-CF** dataset.

DIABIMMUNE dataset

	MAR					MNAR				
Missing rate	30%	40%	50%	60%	70%	30%	40%	50%	60%	70%
DeepMicroGen	**1.600**	**1.663**	**1.755**	**1.784**	**1.741**	**1.654**	**1.623**	**1.676**	**1.752**	**1.765**
MICE	1.832	1.857	1.820	1.812	1.810	1.823	1.821	1.810	1.809	1.829
Cubic	1.841	1.868	1.835	1.825	1.818	1.837	1.829	1.818	1.823	1.842
Linear	1.876	1.900	1.870	1.869	1.866	1.870	1.863	1.869	1.873	1.865
Mean	1.852	1.937	1.891	1.976	1.902	1.925	1.915	1.907	1.941	1.912
Median	1.834	1.964	1.873	1.997	1.902	1.924	1.932	1.908	1.944	1.925
Moving window	1.788	1.861	1.834	1.908	1.798	1.855	1.857	1.869	1.893	1.818
LOCF	1.969	2.058	2.089	2.148	1.993	2.077	2.087	2.072	2.077	2.012

BONUS-CF dataset

	MAR					MNAR				
Missing rate	30%	40%	50%	60%	70%	30%	40%	50%	60%	70%
DeepMicroGen	**0.488**	**0.491**	**0.478**	**0.478**	**0.493**	**0.461**	0.463	**0.491**	**0.502**	**0.509**
MICE	0.524	0.538	0.555	0.536	0.535	0.523	0.525	0.542	0.557	0.543
Cubic	0.642	0.651	0.659	0.643	0.651	0.623	0.629	0.656	0.668	0.652
Linear	0.687	0.693	0.701	0.688	0.695	0.666	0.674	0.700	0.710	0.696
Mean	0.690	0.696	0.704	0.693	0.700	0.669	0.676	0.704	0.713	0.699
Median	0.499	0.508	0.526	0.503	0.502	0.499	0.490	0.512	0.530	0.527
Moving window	0.491	0.495	0.531	0.508	0.510	0.480	0.471	0.513	0.516	0.525
LOCF	0.512	0.503	0.532	0.525	0.514	0.480	**0.461**	0.516	0.519	0.523

Results: Effect of different data missing rates and mechanisms

- **Average MAE** results from bi-directional and unidirectional RNN-based DeepMicroGen with different missing data mechanism.

DIAMIMMUNE dataset

	MAR					MNAR					MCAR				
Missing rate	30%	40%	50%	60%	70%	30%	40%	50%	60%	70%	30%	40%	50%	60%	70%
DeepMicroGen (bi-directional)	1.600	1.663	1.755	1.784	1.741	1.654	1.623	1.676	1.752	1.765	1.573	1.589	1.616	1.642	1.741
DeepMicroGen (unidirectional)	1.635	1.676	1.777	1.821	1.764	1.662	1.697	1.702	1.768	1.773	1.635	1.676	1.677	1.721	1.764

BONUS-CF dataset

	MAR					MNAR					MCAR				
Missing rate	30%	40%	50%	60%	70%	30%	40%	50%	60%	70%	30%	40%	50%	60%	70%
DeepMicroGen (bi-directional)	0.488	0.491	0.478	0.478	0.493	0.461	0.463	0.491	0.502	0.509	0.443	0.475	0.476	0.470	0.505
DeepMicroGen (unidirectional)	0.490	0.492	0.496	0.486	0.527	0.488	0.484	0.493	0.519	0.539	0.461	0.474	0.485	0.485	0.518

Results: Preserving characteristics of missing samples

- **Objective:** To address whether the imputed RA profiles preserve similar characteristics to the original samples, they **randomly selected 10%** of the data considering them as missing.

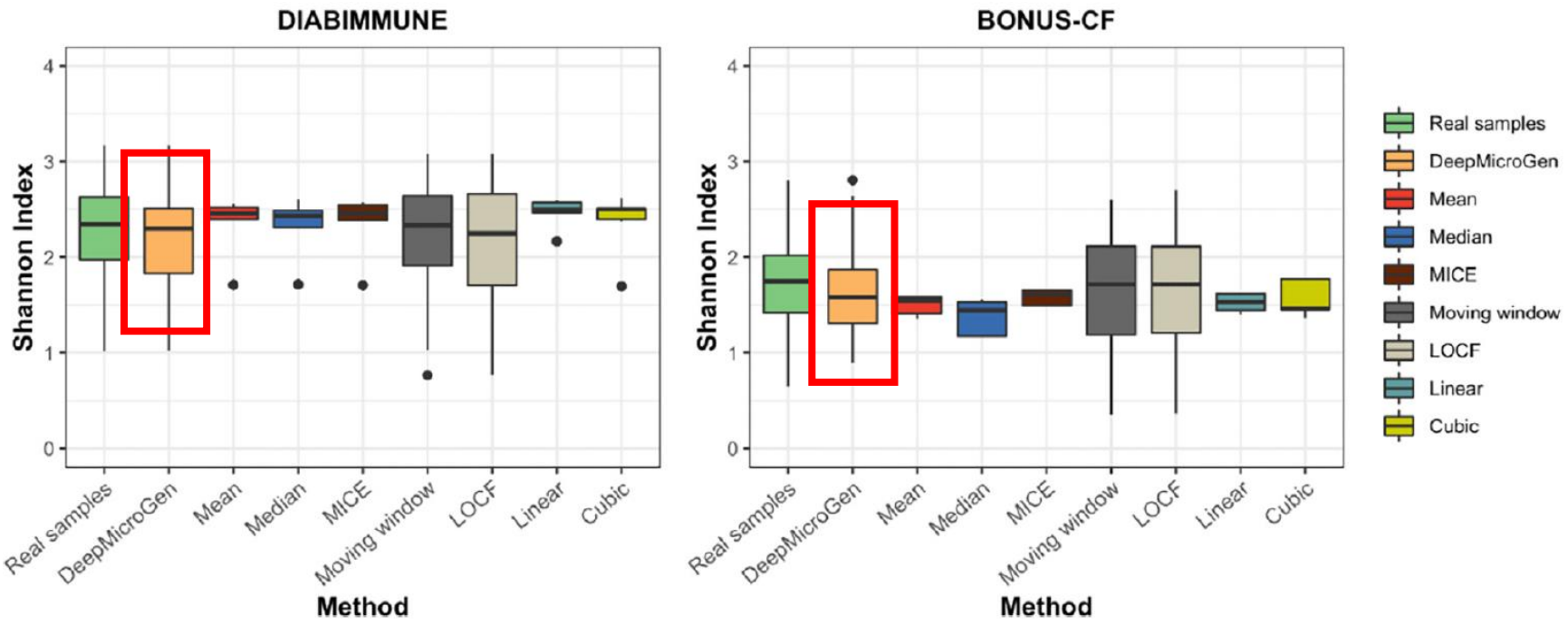
- **Dataset**: DIABIMMUNE & BONUS-CF datasets

- **Methodology:** Performed imputation with DeepMicroGen and baseline methods.

- **Performance Evaluation:**
  1. Comparing **alpha-diversity** (Shannon index) of real vs. imputed data.
  2. Measuring **beta-diversity** using Bray-Curtis distance.
     → Visualizing differences with NMDS.

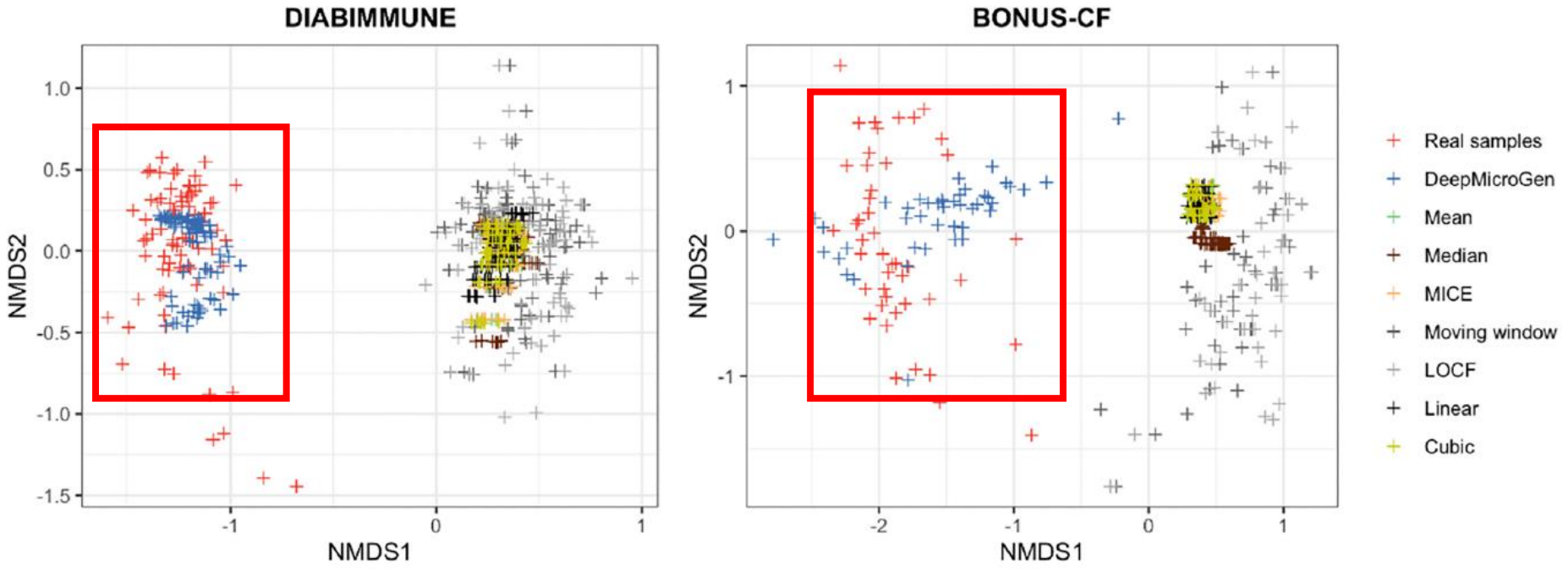
Results: Preserving characteristics of missing samples

- Comparison of **alpha-diversity** based on Shannon index measured from real samples and imputation output from DeepMicroGen and other baseline methods.



Results: Preserving characteristics of missing samples

- **Beta-diversity** visualization using NMDS for the real samples and imputation output from DeepMicroGen and other baseline methods.



Results: Preserving characteristics of missing samples

- Pearson correlation coefficients between the **alpha-diversity** and **beta-diversity** of real samples and the imputation output from each method.

Alpha-diversity

Dataset	DeepMicroGen	Mean	Median	MICE	Linear	Cubic	LOCF	MW
DIABIMMUNE	0.641	0.344	0.410	0.329	0.399	0.326	0.133	0.217
BONUS-CF	0.698	–0.271	0.263	–0.277	–0.273	–0.287	0.136	0.204

Beta-diversity

Dataset	DeepMicroGen	Mean	Median	MICE	Linear	Cubic	LOCF	MW
DIABIMMUNE	0.364	0.336	0.334	0.338	0.321	0.320	0.262	0.309
BONUS-CF	0.210	0.067	–0.009	0.026	0.263	0.147	0.099	–0.048

Results: Disease prediction improvement

- **Objective:** To examine if DeepMicroGen-based imputation enhances disease prediction accuracy.
- **Dataset**: DIABIMMUNE dataset with 141 infants' clinical data on egg, milk, and peanut allergies.
- **Methodology:** Built a one-layer LSTM neural network for each allergy prediction.
- **Performance Evaluation:** Measured Area Under Curve (AUC) for allergy outcome predictions.

Results: Disease prediction improvement

- Average AUC results for the allergy outcome predictions of the classifier trained with the addition of the 25 imputed subjects using different methods, repeating 5-fold cross-validations five times.

Allergy	w/o Imp	With imputation							
		DeepMicroGen	Mean	Median	MICE	Linear	Cubic	LOCF	MW
Milk	0.566	**0.605**	0.589	0.589	0.556	0.550	0.585	0.584	0.563
Egg	0.556	**0.638**	0.542	0.550	0.553	0.590	0.552	0.535	0.514
Peanut	0.512	**0.612**	0.559	0.511	0.515	0.487	0.483	0.565	0.508

Results: Comparison with other Deep Learning models:

- Average MAE results for longitudinal microbiome data imputation with DeepMicroGen and Gao et al. performing 10-fold cross-validation.

Dataset	DeepMicroGen	Gao et al.
DIABIMMUNE	**1.609**	3.267
BONUS-CF	**0.474**	1.011

- Imputation performance results based on the average MAE for different missing data mechanism.

		MAR					MNAR				
Dataset	Missing rate	30%	40%	50%	60%	70%	30%	40%	50%	60%	70%
DIABIMMUNE	DeepMicroGen	1.600	1.663	1.755	1.784	1.741	1.654	1.623	1.676	1.752	1.765
	Gao et al.	3.320	3.230	3.277	3.215	3.313	3.335	3.241	3.165	3.319	3.249
BONUS-CF	DeepMicroGen	0.488	0.491	0.478	0.478	0.493	0.461	0.463	0.491	0.502	0.509
	Gao et al.	1.036	0.977	1.103	1.014	0.997	0.982	0.984	1.016	1.051	0.999

Discussion

Discussion: Overview

- **DeepMicroGen Overview:**
  - A GAN-based model for imputing longitudinal microbiome data.
  - Utilizes CNN for feature extraction and biRNN for generating imputed datasets.
  - Discriminator differentiates between actual and imputed values and predicts the timepoint.

- **Performance Evaluation:**
  - Tested against baseline methods using both simulated and real-study datasets.
  - Showed lowest average MAE, outperforming other methods.
  - Demonstrated robust imputation performance.

Discussion: Limitation and extensions

- **Limitation of DeepMicroGen:**
  - Assumes uniform time intervals between samples, may not perform well with irregular intervals.
  - Requires a sufficient number of samples per time point for effective training.

- **Potential Extensions:**
  - Methodology could be adapted for other omics datasets, like RNA-seq and DNA methylation data.
  - Requires specific training and optimization for each omics data type.

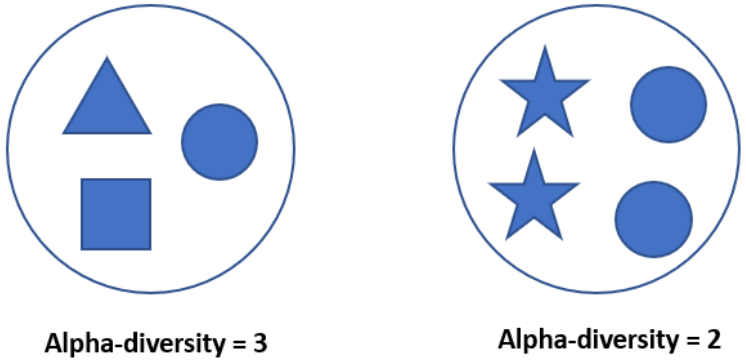
Thank you

Supplementary

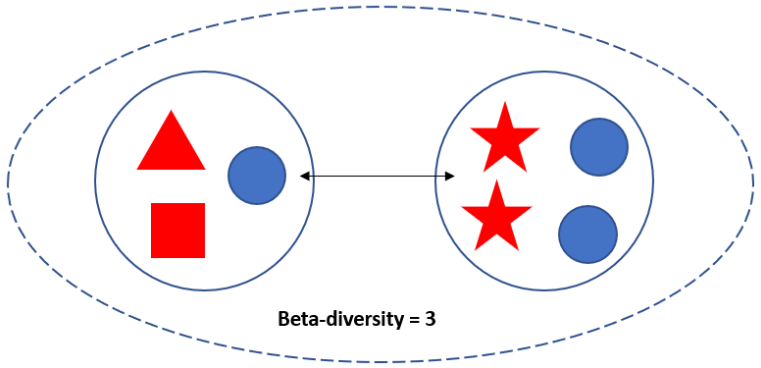
- **Missingness:**
  - **MCAR**: Probability of missingness is independent of the data.
  - **MAR**: Probability of missingness is independent of the missing values given the observed data. (but it is related to some of the observed data.)
  - **MNAR**: MNAR occurs when the missingness is directly related to the values of the missing data itself.

Supplementary

- **Alpha diversity:** Diversity within ecological units or habitats



- **Beta diversity:** Differences in diversity between habitats



Supplementary

- **Shannon Index:** Measure of diversity in a community. It considers both the abundance and evenness of the species present.

$$\mathrm{H} = -\sum_{i=0}^{s} p_i \ln p_i \qquad p_i = \frac{n_i}{N}$$

→ Where p_i is the proportion of samples belonging to i^{th} species in the dataset.

- **Bray-Curtis:** Used to represent how different in terms of abundance

$$BC = \frac{\sum |n_{i1} - n_{i2}|}{\sum (n_{i1} + n_{i2})}$$

→ Where n_{i1} and n_{i2} are the counts of species i in the first and second community