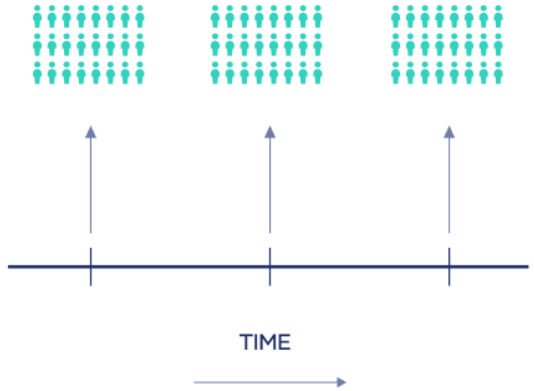
Developing a generative adversarial network-based method for longitudinal microbiome data imputation

CS6824 project proposal

Presenter: Joung Min Choi

Introduction

- **Microbiome studies**
  - The collection of microorganisms living in a certain environment
  - Key role in complex disease such as obesity, diabetes, cancer, and allergy outcomes
  - Potential biomarkers for disease diagnosis or as therapeutic targets for treatment



- **Longitudinal studies of microbiome**
  - Alteration over time by infections or medical interventions
  - Changes of microbiome composition over time and the association with disease outcomes

  - **Major challenge**
    : Limited sample size due to the uneven number of timepoints along the longitudinal timeline of different subjects

	1st month	2nd month	3rd month
Patient 1	O	X	O
Patient 2	O	O	X
Patient 3	O	O	O

Introduction

- **Generative adversarial network (GAN)**
  - Widely adopted in various fields to address the lack of data issue
  - Data augmentation framework to improve classification tasks by reducing overfitting
  - Recently, being utilized for imputation of missing values for a multivariate time-series with RNN

- **Applications of GAN in microbiome study**
  - MB-GAN (2020): a simulation framework for microbiome data based on GAN
  - DeepBioGen (2021): a data augmentation procedure that characterizes visual patterns of sequencing profiles

  ✓ **But still, the presented methods only simulate single time point microbiome data**

  ✓ **Data imputation for longitudinal microbiome data has not been addressed, yet.**

Methods

- **DeepMicroGen**
  - A generative adversarial network-based method for longitudinal microbiome data imputation

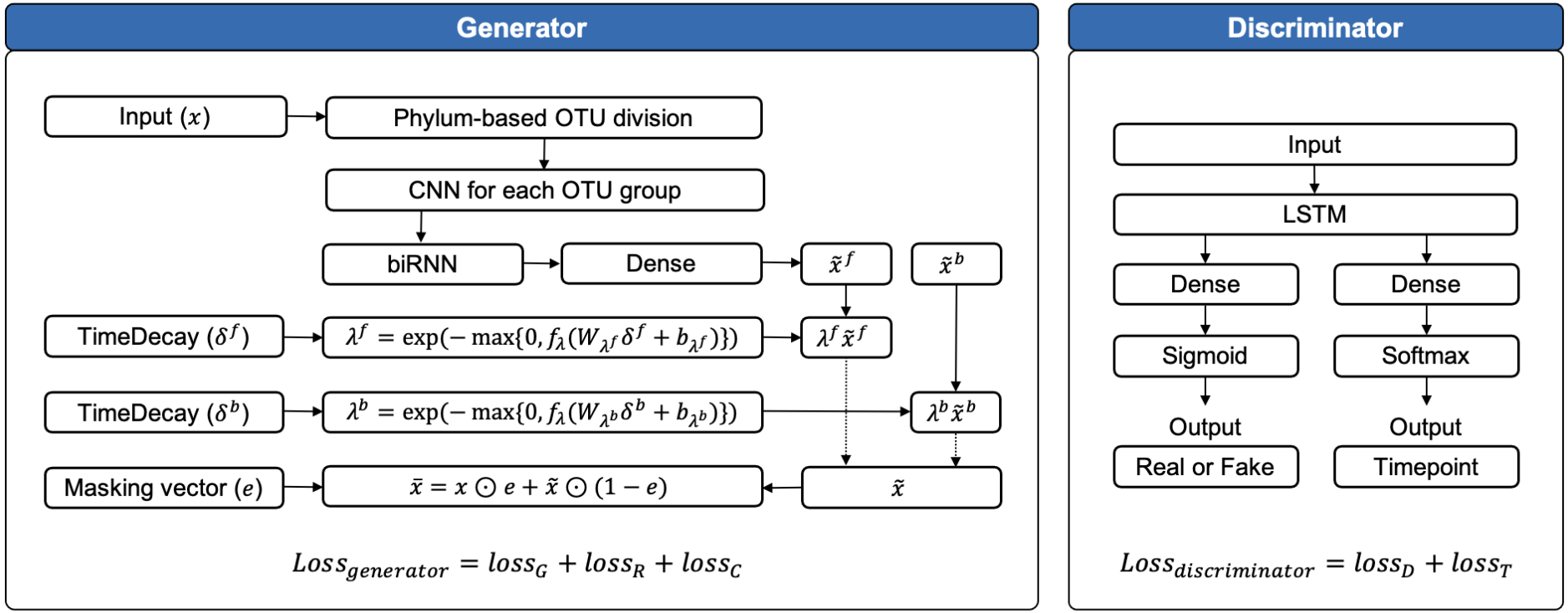


Fig 1. Illustration of DeepMicroGen.

Methods

	2 month	3 month	4 month	5 month
Patient 1	O	X	X	O
Patient 2	O	O	X	O

1. Generator

- **Input**
  - **Multivariate longitudinal data** X with n timepoints $T = (t_0, t_1, \ldots, t_n)$
  - **Mask vector** e : Indicating which components of X are missing

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

  - **Time gap** δ_i^f, δ_i^b

    : Between the last observed value and the current time step in the forward and backward direction

$$\delta_i^f = \begin{cases} t_i - t_{i-1}, & if\ \boldsymbol{e}_i = 1, i > 1 \\ \delta_{i-1}^f + t_i - t_{i-1}, & if\ \boldsymbol{e}_i = 0, i > 1 \\ 0, & if\ i = 1 \end{cases}$$

- **Output**
  - $\tilde{x}$: The generated values for the x
  - The generated values in $\tilde{x}$ are replaced by the actual values (if existed in the input) to obtain the final imputed output $\bar{x}$:

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

Methods

1) Phylum-based OTU division based on the method from phyLoSTM[1]

① Divide OTUs of input data into each group based on each phylum

② Reducing each row of the Spearman rank coefficient matrix to a cumulative correlation coefficient, using the formula :

$$\rho_{OTU_{row_j}} = \sqrt[p]{|\rho_{OTU_{j1}}| \cdot |\rho_{OTU_{j2}}| \cdot \ldots \cdot |\rho_{OTU_{jp}}|}$$

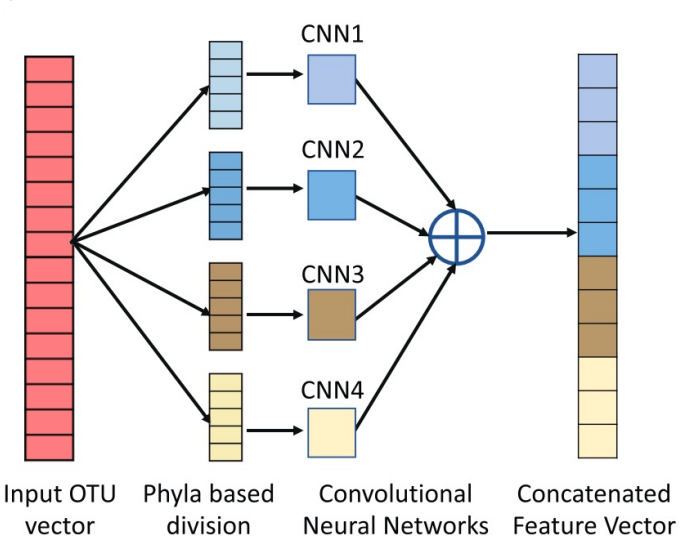
($p \times p$ matrix for p OTUs in a cluster)

③ Arranging each of these cumulative coefficients in a decreasing order

④ Re-ordering OTUs based on the decreasing order of the cumulative correlation coefficients

2) Apply CNN individually to each group to extract feature

3) Concatenate the features extracted from each CNN



[1] Sharma, D., & Xu, W. (2021). phyLoSTM: a novel deep learning model on disease prediction from longitudinal microbiome data. Bioinformatics, 37(21), 3707-3714.

Methods

4) Generation

- Obtain outputs from bidirectional recurrent neural network layer: $\tilde{x}^f$, $\tilde{x}^b$
- Combine the forward and backward imputation vectors to calculated the final generated values:

$$\tilde{x} = \lambda^f \tilde{x}^f + \lambda^b \tilde{x}^b$$

✓ **Combination factors** (λ^f, λ^b)

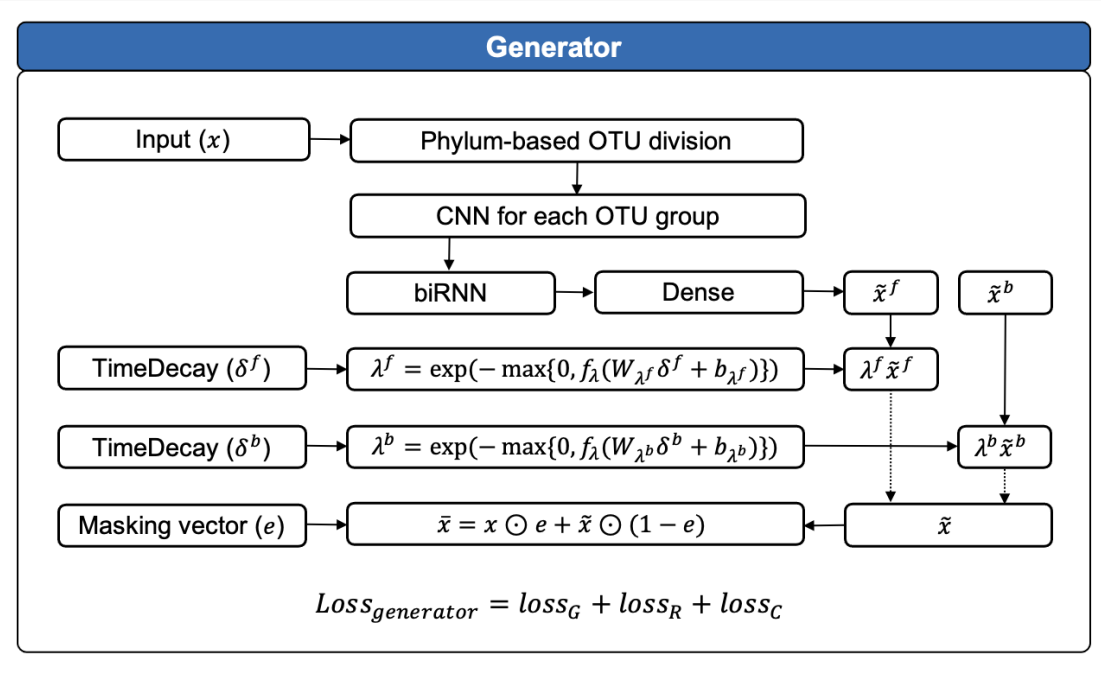
: Controlling the influence of the forward and backward imputed values based on how far the last observed value in both directions was

$$\lambda^f = \exp(-\max\{0, f_\lambda(W_{\lambda_f}\delta^f + b_{\lambda_f})\})$$

$$\lambda^b = \exp(-\max\{0, f_\lambda(W_{\lambda_b}\delta^b + b_{\lambda_b})\})$$

- **Final imputed value**

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



Methods

- **Generator loss**

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

  - **Generator loss**

    : Training the generator to maximize the probability that discriminator D classifies the fake instances as actual values

$$loss_G = -\log(D(\bar{x} \odot (1 - e)))$$

  - **Reconstruction loss**

    : The absolute error between the actual values and the corresponding generated values

$$loss_R = |x \odot e - \tilde{x} \odot e|$$

  - **Consistency loss**

    : The difference between the imputed output from the forward and backward directions

$$loss_c = \left|\tilde{x}^f - \tilde{x}^b\right|$$

Methods

2. Discriminator

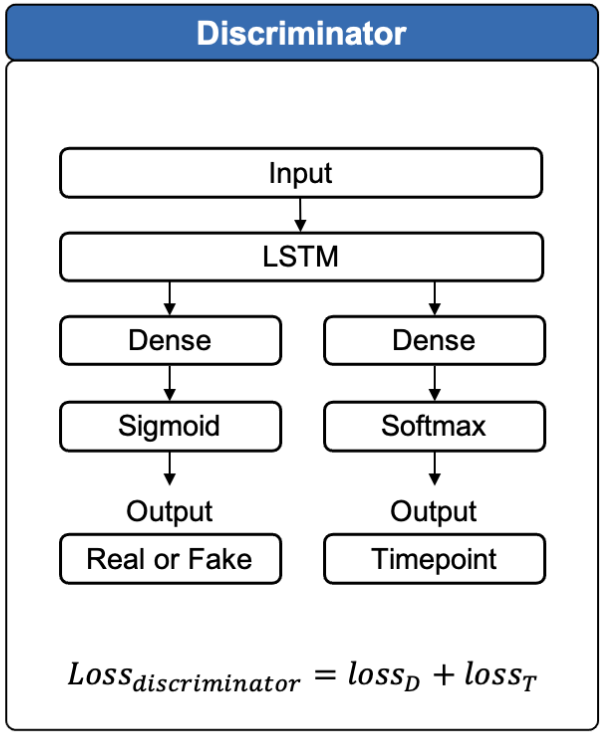
- One LSTM recurrent neural network and fully connected layers
  1) Training to classify the actual (real) and the generated values (fake)
  2) Training to classify the timepoint of each sample

- **Discriminator loss**

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

- $loss_D$: binary cross entropy loss to maximize the probability of correctly classifying the actual values as real and the generated values as fake

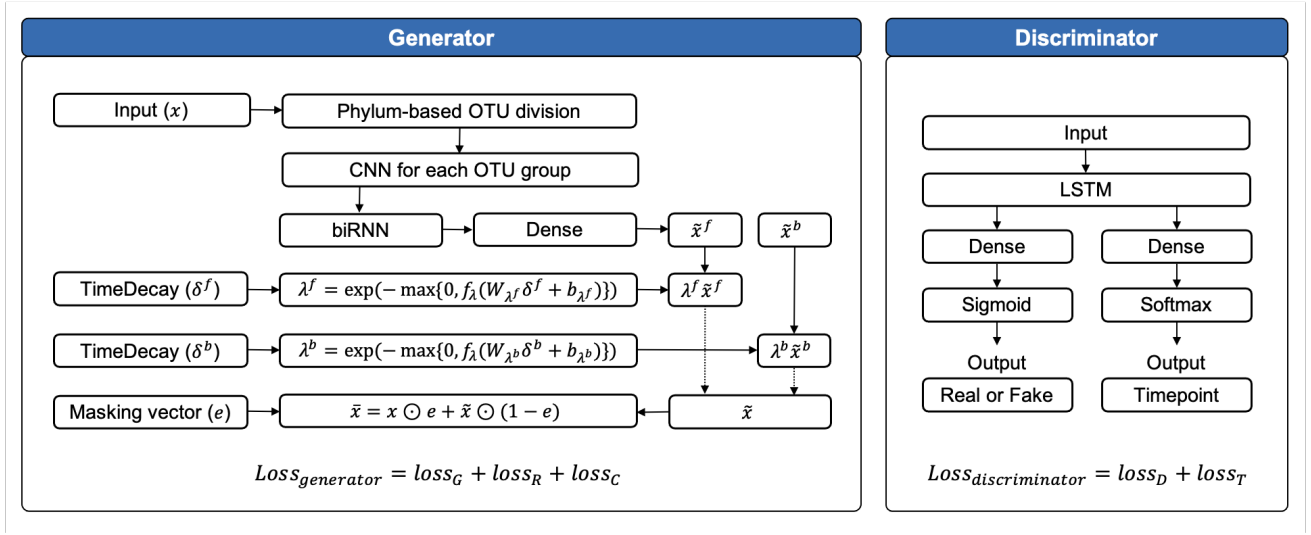
- $loss_T$: cross entropy loss to maximize the probability of correctly predicting the time point for each sample



Methods

- **Training**
  - Adam optimizer with the learning rate of 1e-3
  - The generator was trained after five iterations of the discriminator
  - The model training was stopped when the reconstruction loss did not increase within 100 epochs

- **Hyperparameter tuning**
  - Choose the number of nodes and the number of layers showing the best reconstruction loss



Experimental Design

1. Datasets

: Species-level relative abundance profiles

- **DIABIMMUNE dataset**
  - Infants recruited from three countries having substantial differences of incidence of type 1 diabetes and allergies
  - 16S rRNA dataset with 115 OTUs at the species level
  - 1064 samples of 133 subjects for 8 time points (4, 7, 10, 13, 16, 19, 22, 28 month)

- **BONUS-CF dataset**
  - Fecal samples of the 231 infants during regularly scheduled clinical visits in the first year of life (3, 4, 5, 6, 8, 10, and 12 month)
  - Shotgun sequencing dataset with 1374 OTUs at the species level
  - Total of 452 samples of 113 subjects for 4 time points (5, 6, 8, 10 month)

Experimental Design

2. Competing methods

1) Simple imputation method

- Mean
- Median

2) Traditional time-series imputation method

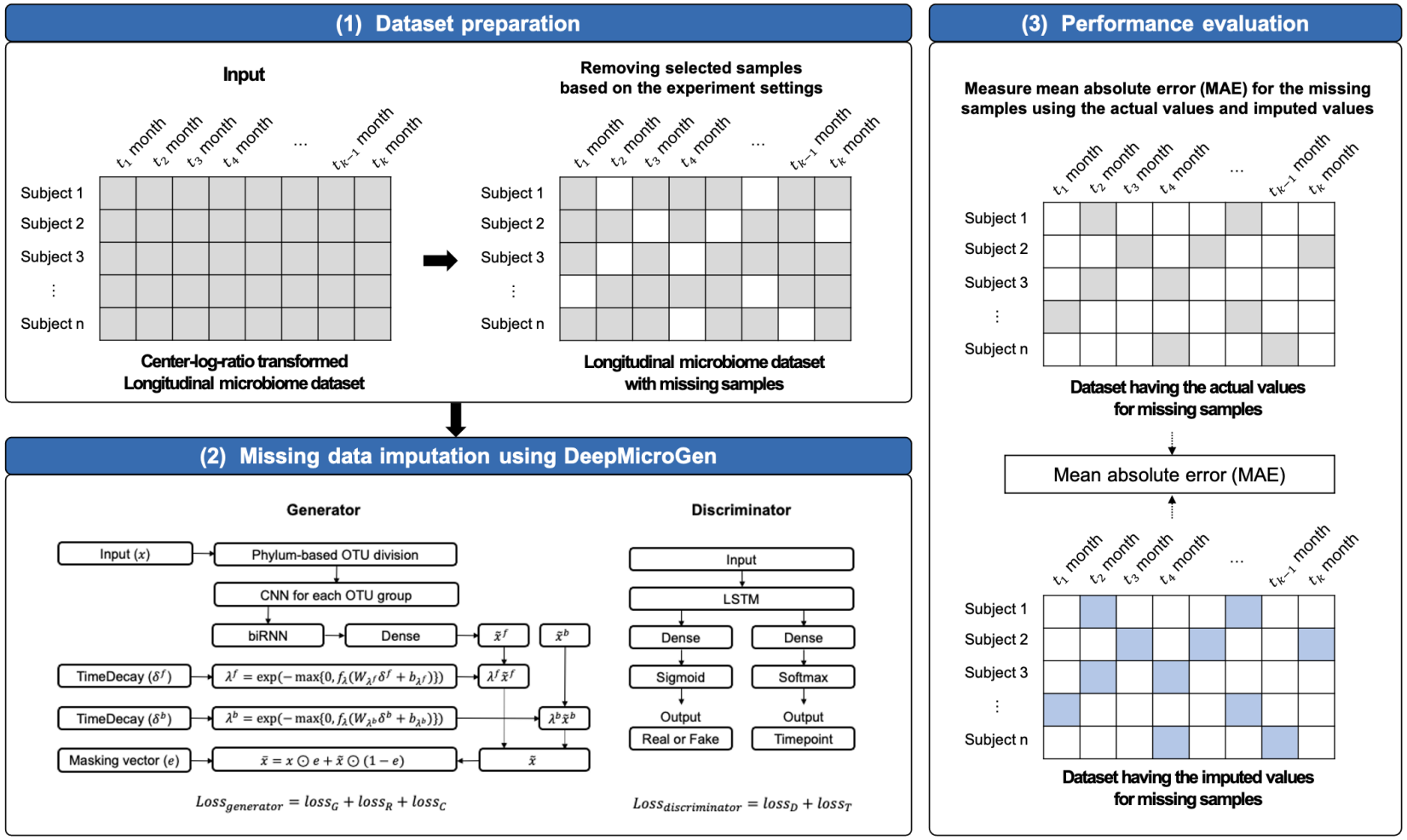
- Linear curve fitting
- Cubic curve fitting
- Moving-window-based imputation (win size = 3)

3) Widely-used imputation method for longitudinal dataset

- Multiple imputation by chained equations (MICE)
- Last Observation Carried Forward (LOCF)

Experimental Design

Fig 2. Illustration of the experimental design for the performance evaluation

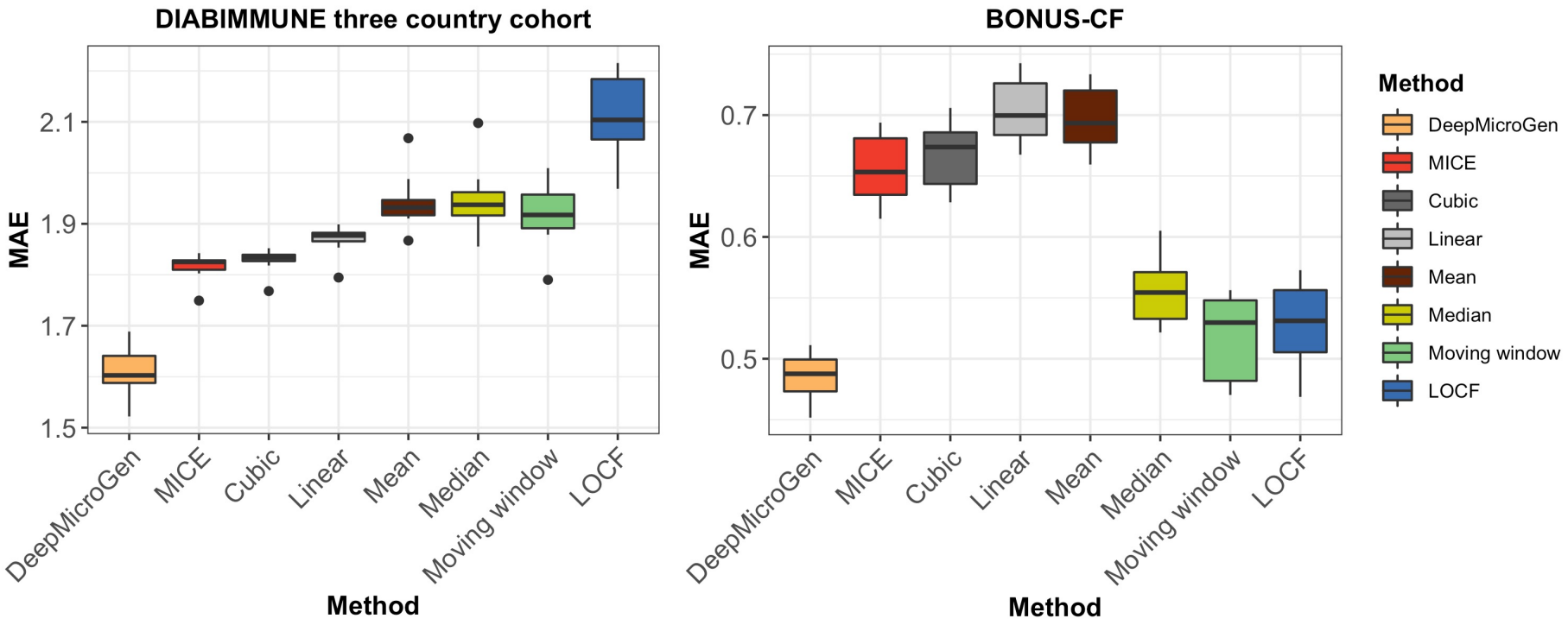


Results

1. Performance evaluation of DeepMicroGen

- Split the dataset based on the 10 cross-validation and consider the test set as missing datasets
- Perform data imputation and measure the mean absolute error

Fig 3. Performance comparison of DeepMicroGen with the baseline methods based on 10-fold cross validation



Results

2. Effectiveness of each component in DeepMicroGen

- Design four variants of DeepMicroGen by removing or changing the components in the model
- Evaluate the performance based on 10-fold cross-validation

- **Four variants**
  - **RNN** : Eliminate both the discriminator and the CNN-based feature extraction
  - **biGAN** : Remove CNN-based feature extraction
  - **MDeep+biGAN**: Features encoding the phylogenetic correlation based on the method presented in MDeep
  - **AE + biGAN**: Replace CNN by the autoencoder (AE)

Table 1. Average MAE results under different neural network architectures for imputation performing 10-fold cross-validation

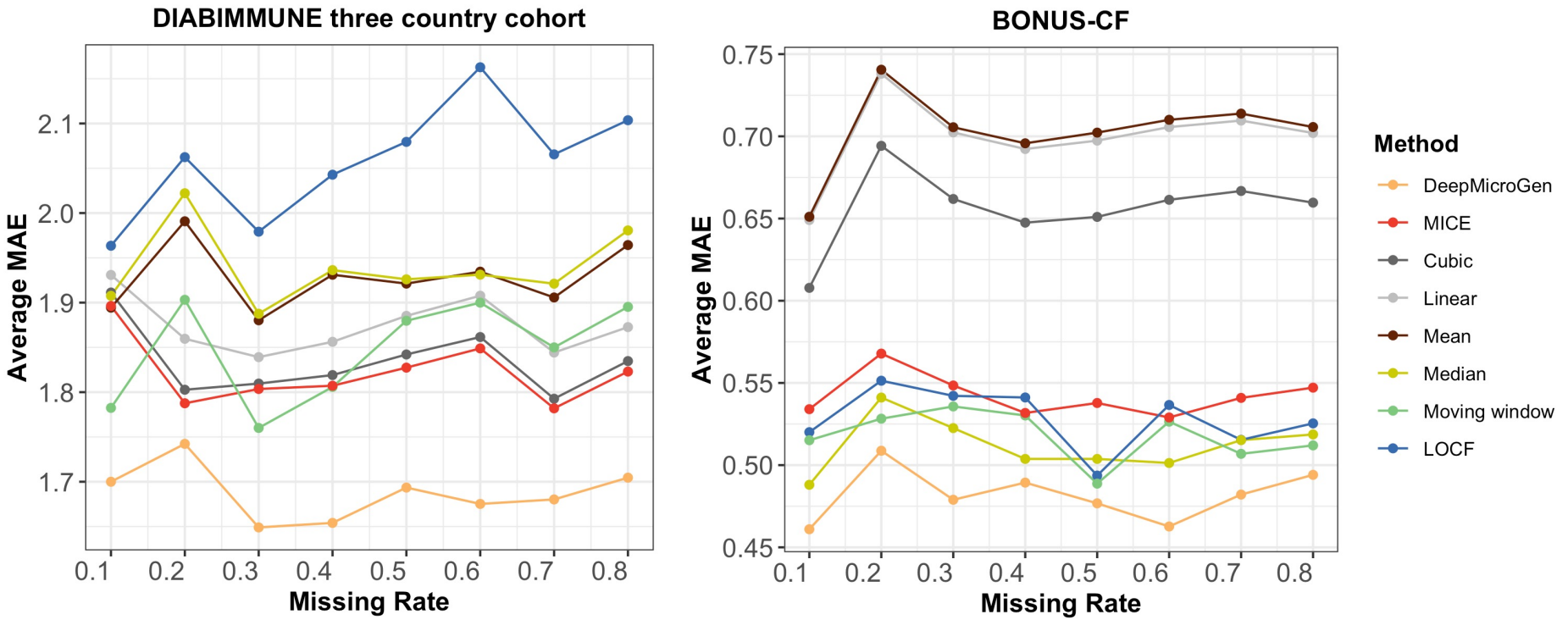
Dataset	RNN	biGAN	MDeep+biGAN	AE+biGAN	DeepMicroGen
DIABIMMUNE	1.672	1.662	1.884	2.759	**1.609**
BONUS-CF	0.535	0.521	0.559	0.732	**0.474**

Results

3. The effect of different data missing rates

- Randomly sampling the 10% to 80% of the samples and consider them as missing samples
- Perform data imputation and measure MAE, and repeat the experiment five times

Fig 4. Imputation performance results with different missing rates for DeepMicroGen and other baseline methods



Results

4. Testing performance for the different missing data mechanism (Working on)

- Previous experiments settings are said to be missing completely at random (MCAR), often unrealistic for the data at hand

- Performance evaluation for MAR and MNAR settings with different missing rates

  - **Missing at random (MAR)**

    : The cause of the missingness depends on some observed variable or variables that have been collected

    ex) If male participants are less likely to complete a survey about depression severity than female participants,
    that is, if probability of completion of the survey is related to their sex but not the severity of their depression

  - **Missing not at random (MNAR)**

    : The cause of the missingness cannot be controlled for

    ex) The fact that the data are missing is systematically related to the unobserved data,
    that is, the missingness is related to events or factors which are not measured by the researcher

Results

5. Improvement of the disease prediction through data imputation

- DIABIMMUNE dataset
  - Number of infants having all samples for 8 time points with clinical information: 117 subjects
  - Number of infants having missing samples with clinical information: 25 subjects

- Constructing a clinical outcome classification model and perform 5-fold cross-validation

  : Simple one-layer LSTM-based neural network

  1) Model w/o Imp : Training based on 117 subjects
  2) Model with Imp : Training based on 117 + 25 subjects (Imp)

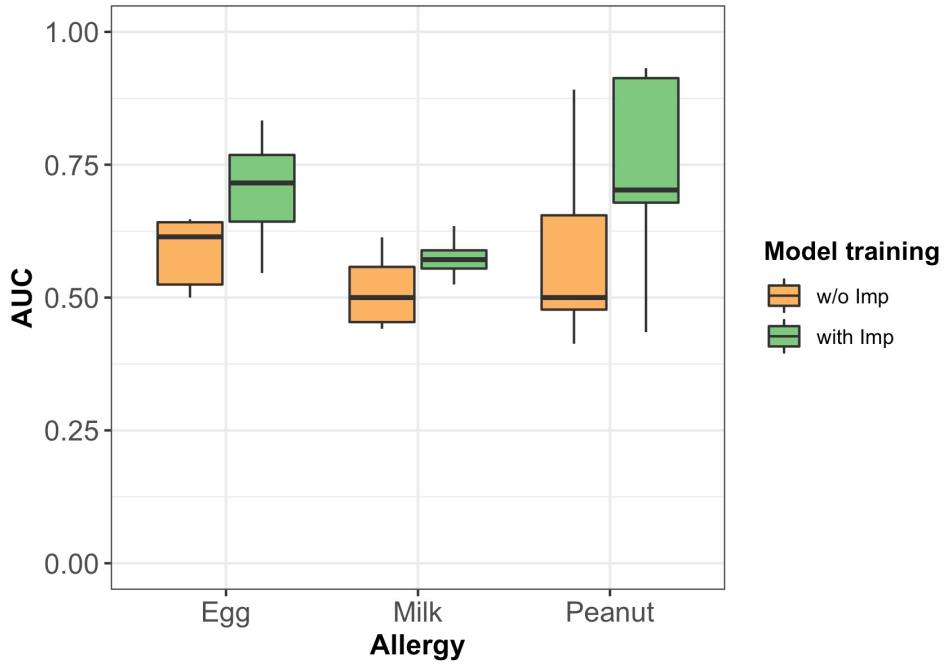


Fig 5. Performance improvement results for the allergy outcome prediction when training the model with the addition of imputed 25 subjects having missing samples