**Online methods**

**Framework of GAN-GMHI**

The GAN-GMHI framework consists of three stages. First, a dataset containing phenotype and batch information for all samples is constructed. Second, GAN is applied to guide the batch effect correction of raw data. Third, the corrected dataset is output as the training dataset for GMHI prediction (**Figure S1**). The batch effect removal method of iMAP [1], a GAN method previously applied on single-cell RNA-Seq data, was adapted for batch effect removal in this study. It is worth noting that the datasets to be batch-corrected by GAN must be classified based on the phenotype first, and the sub datasets of each phenotype are regrouped according to the batch. Such processing ensures that the unwanted technical variations among different datasets are eliminated, and the biological differences between different phenotypes are retained. We should emphasize that GAN-GMHI applied the same core functions as iMAP. However, we optimized the structure of the model to better fit the microbiome abundance data (see the script at https://github.com/HUST-NingKang-Lab/GAN-GMHI/blob/main/demo/GAN4BER.py).

**The architecture of GAN model**

There are two stages of the GAN model. The first stage is disentangled representations of biological content and technical variations. The second stage is batch effect removal of the microbiome profile by GAN.

In the first stage, we designed a novel autoencoder structure to build representations of biological content, which are expected to be disentangled from technical variations. Three forward neural networks are deployed in this stage, including one content encoder , two generators (decoders) and . The inputs to the model include the microbiome profile vector of one individual’s gut microbial community denoted as , and its batch indicator vector . One-hot encoding strategy is used to indicate the batch of the gut. For instance, in the case of three batches, gut microbial communities from the first batch have their batch indicator vector . The output of the encoder is denoted as , which is expected to exclusively represent the biological differences of communities, and be ignorant of the technical measuring process. The neural network is deployed to generate the representation of measurement noise , since the measurement noise cannot be fully captured by a simple one-hot vector. Another generator is further used to finish the reconstruction of the original profile vector. The inputs to the generator include both and , because intuitively, it is possible for the generator to reconstruct the original measured microbiome profile vector only if both the biological content and technical noise are simultaneously provided. The final reconstructed microbiome profile vector is , where is the ReLU function , and is used to match the range of reconstructed vector with the original microbiome profile vector. In the second stage, we further use a GAN-based model to almost perfectly match the data distributions of the shared phenotypes across different batches and then generate the corrected microbiome profiles. This GAN-based model is composed of two neural networks, one generator , mapping microbiome profile vector to a pseudo-microbiome profile vector , and one discriminator , discriminating the pseudo-microbiome from the true profile vector .

We deploy a total of five neural networks. By default, the encoder from the first stage is a *d*→1024→ 512→*l* three-layer (not including the input layer) network (*d* is the input dimension of microbiome profile vectors, and *l* is the dimension of biological content representations). The decoder is a *n*→512→1024→*d* three-layer network (*n* is the number of batches), and the decoder is ()→512→1024→*d*. For all networks, the first two layers have a Mish activation, while the last layer have a ReLU activation. For the second stage, the generator is the ResNet, which means (*f* is a ReLU function), and *F* itself is an autoencoder structure, *d*→1024→512→*l*→512→1024→*d* (all layers are activated by Mish except the middle one). Be default, *l* is set to 256. The discriminator is again a three-layer network *d*→512→512→1. We adopt the Adam optimizer to train the networks, with the learning rate 5e-4 for first stage and 2e-4 for the second.

**Datasets**

In this study, a total of metagenomic data from 2,636 healthy and 1,711 non-healthy individuals were collected. Among them, the non-healthy individuals included 12 disease phenotypes. All the data were compiled into meta-datasets as raw data (RAW); The raw data was integrated based on the sample phenotype, and the batch correction data (GAN) was obtained by GAN within the group. Additionally, we also have used 679 samples (118 healthy and 561 non-healthy) as testing set to verify the stability of GAN-GMHI.

**Data integration**

Transferring the calculation process of a generative adversarial network (GAN) to the data integration of this research, the batch of data with the largest standard deviation needs to be fixed, and other data are regarded as generated data, the difference among fixed data and other batches of data can be minimized at last.

In data integration, the data is first classified by phenotype, and the sub datasets of each phenotype are regrouped according to batches. Batches with less than 15 samples are combined into a reorganization batch, whose sample capacity does not exceed 100, and a recombination batch is formed for excess samples until all samples are merged. It should be noted that the reorganization should be carried out as far as possible in accordance with the principle of "similar batches are merged first", that is, to ensure that the samples in the reorganized batches have larger repetitions or similar properties. Batch differences are mostly based on large sample sizes, and small sample sizes can be ignored, which means that the original data should be kept as much as possible to ensure that the biological significance remains unchanged. Therefore, for some phenotypes, the number of samples in its sub dataset is too small (less than 100 samples) but contains multiple batches, batch correction will not be performed.

For the integration of different batches within the same phenotype, the std function of the python package "NumPy" is used to calculate the standard deviation for descending order. The data with the largest standard deviation is regarded as fixed data, other batches need to be batch corrected by GAN. Choosing the fixed data with the largest standard deviation is beneficial to cover more sample points and types, as far as possible to make the samples of other batches match the fixed sample well, to effectively reduce the batch difference.

**Evaluation and benchmark**

In order to verify the accuracy and stability of GAN-GMHI, vertical and horizontal comparisons were made in different ways. Firstly, using the RAW and GAN data as the training data sets respectively. Then, for the two data sets, GMHI and Random Forest methods are used to establish a health prediction model respectively, and make predictions on independent population cohorts (**Figure S1**). On the other hand, we respectively calculated the prediction accuracy of GMHI based on the RAW, GAN, and Combat data sets.

We compared the GAN method with three leading batch effect removal methods: ComBat [2], Seurat3 [3], and Harmony [4]. To assess the performance of GAN method and other methods, we used accuracy, precision, recall, F1, and AUROC measures with the following formulas:

where is the number of true positive predictions, is the number of true negative predictions, is the number of false positive predictions, and is the number of false negative predictions. AUROC is the area under the receiver operating curve.

**Table S1. The performance of different methods before and after batch correction.**

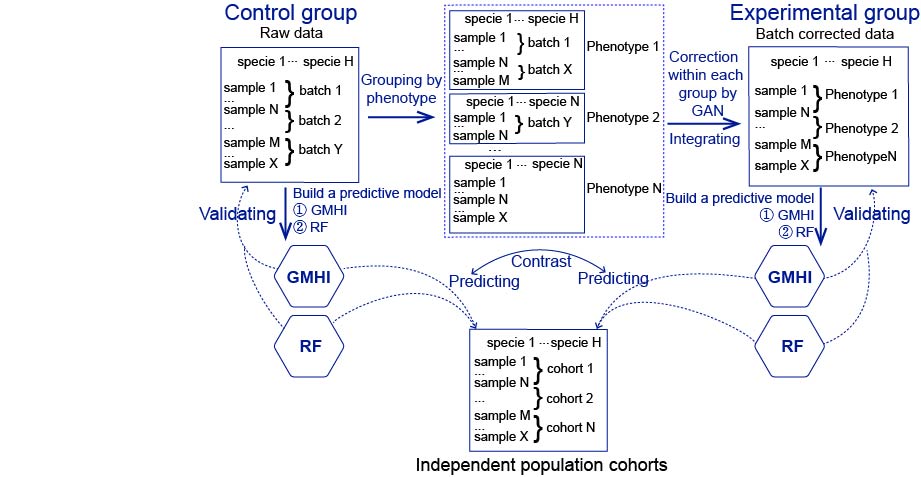
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Method** | **Training cohort (4347 samples)** | | | **Testing cohort (679 samples)** | | |
|  | **Accuracy** | **AUROC** | **F1** | **Accuracy** | **AUROC** | **F1** |
| GMHI | 70.95% | 0.7772 | 0.7635 | 72.61% | 0.8097 | 0.4727 |
| GAN-GMHI | 88.70% | 0.9476 | 0.9040 | 73.05% | 0.6774 | 0.2802 |
| ComBat-GMHI | 72.00% | 0.7843 | 0.7671 | 62.89% | 0.5481 | 0.2941 |
| Seurat3-GMHI | 70.36% | 0.7674 | 0.7605 | 63.62% | 0.8016 | 0.4513 |
| Harmony-GMHI | 44.65% | 0.5689 | 0.2885 | 49.78% | 0.2817 | 0.0671 |

*Note*: ComBat-GMHI, GMHI with ComBat enhancement; Seurat3-GMHI, GMHI with Seurat3 enhancement; Harmony-GMHI, GMHI with Harmony enhancement.

**Table S2. The performance of the GMHI and random forest for classification**

|  |  |  |
| --- | --- | --- |
| **Method** | **Training cohort (4347 samples)** | **Testing cohort (679 samples)** |
|  | **Accuracy (%)** | **Accuracy (%)** |
| GMHI | 70.95 | 72.61 |
| RF | 99.71 | 73.17 |
| GAN-GMHI | 88.70 | 73.05 |
| GAN-RF | 99.98 | 75.27 |

*Note*: RF, random forest; GAN-GMHI, GMHI with GAN enhancement; GAN-RF, random forest with GAN enhancement.



**Figure S1. Schematic diagram of the GAN-GMHI framework.** The GAN-GMHI framework consists of three stages. First, a dataset containing phenotype and batch information for all samples is constructed. Second, GAN is applied to guide the batch effect correction of raw data. Third, the corrected dataset is output as the training dataset for GMHI prediction.

**References**

1. Wang D, Hou S, Zhang L, Wang X, Liu B, Zhang Z. iMAP: integration of multiple single-cell datasets by adversarial paired transfer networks. Genome Biology 2021;**22**:63.

2. Stein CK, Qu P, Epstein J, Buros A, Rosenthal A, Crowley J*, et al.* Removing batch effects from purified plasma cell gene expression microarrays with modified ComBat. BMC Bioinformatics 2015;**16**:63.

3. Stuart T, Butler A, Hoffman P, Hafemeister C, Papalexi E, Mauck WM*, et al.* Comprehensive Integration of Single-Cell Data. Cell 2019;**177**:1888-902.e21.

4. Zhang Q, He Y, Luo N, Patel SJ, Han Y, Gao R*, et al.* Landscape and Dynamics of Single Immune Cells in Hepatocellular Carcinoma. Cell 2019;**179**:829-45.e20.