**Method**:

**Datasets and preprocess**:

We use the dataset created and made available by Tsagkogeorga *et al*. Original dataset collected from Harmonizome website(<https://maayanlab.cloud/Harmonizome>). 15 one-hot-encoded datasets were selected to construct our dataset. The dataset was initially standardized to continuous-value ranged from 0 to 1, or -1 to 1 where 1 indicated strong positive gene-feature association, 0 indicates no gene-feature association observed and -1 indicates strong negative gene-feature association.

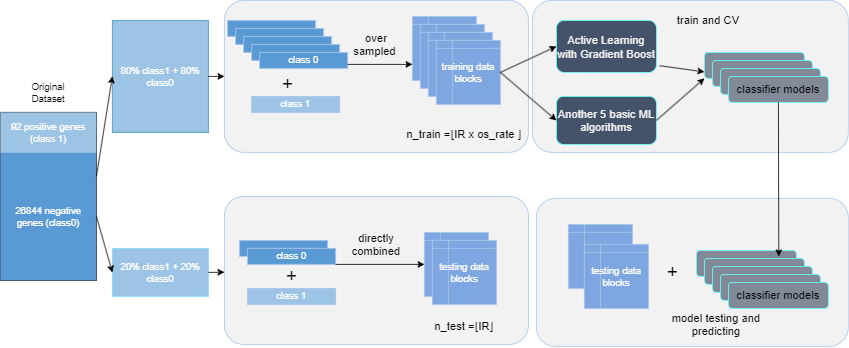
To gain high-informative features and reduce feature dimensions, following 2 kinds of features will be removed: (i) zero values more than 70%. (ii)variance less than 16% of whole.

**Data augmentation by SMOTE**:

To predict genes associates with RNA methylation pathways, we took 80% positive and negative samples for training and CV while other 20% samples remaining for testing. Next, calculated the imbalance ratio (IR), i.e., the ratio between negative and positive examples.

Realizing dataset’s IR is higher than most public datasets’, reaching more than 290, oversample algorithm was applied to alleviate imbalance problem: Assuming parameter *OSRATE*, a decimal number between 0 and 1 used to determine oversample rate. This parameter can be obtained through cross-validation. We next divided majority training samples (negative training samples) into *NTRAIN = IR\* OSRATE* (round down) equal pieces, combining each negative piece with the same positive sample to get *NTRAIN* copies of data blocks. Then, each data block was balanced by SMOTE. Values in generated samples was assigned to discrete or continuous value between -1 and 1.

For the test set, calculating the IR should yield the same value as the training set. Then, divide the negative samples in the test set into IR (rounded down) blocks evenly. Combine each block of negative samples with the same positive samples to obtain a test set with IR training blocks.. As shown in FIG 1.



【FIG1】

**Generating synthetic protein feature samples with WGAN-GP:**

We trained a generative adversarial network (GAN) to fit the distribution of positive samples in the training set. The generator of GAN was utilized for oversampling the minority set. To achieve the best fit of the distribution of positive samples, whole 92 positive samples were used to the GAN training. Realizing the unstable nature of GAN training, we set early-stopping parameter *STOPTHRESHOLD*. We employed SVM to perform *classifier two-sample test* (CTST) following each epoch ending. The accuracy of SVM’s cross-validation served as the CTST outcome. The models which have difference with 0.5, as measured by CTST result, lower than the *STOPTHRESHOLD* will be saved.

After generator training, we conducted additional CTST on all retained generators. Each generator was used to generate fake samples equal to 20% of the number of positive samples, and the CTST experiment will be conducted using SVM. CTST will repeats N times on each generator. Generators with average CTST result closest to 0.5 will be selected and then used for similar oversampling training approach as SMOTE.

**Training detail:**

We applied 5 machine-learning classifiers: SVM, GB, GNB, RF, LR. Trained on NTRAIN=147 balanced data blocks. Grid-search technique was used to select best (accuracy) hyperparameters among given parameter combinations in 3 machine-learning classifiers (SVM, GB, RF). 3-fold cross-validation was used in grid-search to select best hyperparameters. None-sample method (OSRATE = 1) was set as baseline.

We choose WGAN-GP as our GAN, CTST repeats 100 times to select the best generator. SVM’s parameter in CTST was default assigned. *STOPTHRESHOLD* was set to 0.02. To better apply to gene expression data, we improved the architecture of the GAN by replacing CNNs in both generator and discriminator with full-connected layers. Each FC layer followed by leaky-relu activate function (negative slope = 0.2). The discriminator consists of 4-layer fully connected layers with input dimension of 1517. The output size of each layer was half of its input, final output size of discriminator was set to 1. For generator, input size was set to 128, the output size of each layer was double of its input, final output size of generator was set to 1517(equal to the data dimension). The optimizer of GAN was ADAM, with parameters beta1 and beta2 being 0.95 and 0.9, respectively. The batch size was set to 92, and the learning rate for both discriminator and generator was 0.0001. The parameter *NCRITIC* was set to 7, which means that the generator was trained once after every 7 training iterations of the discriminator. We set the epoch num to 1500, however, the generator was able to fully fit the distribution of the training samples after approximately 600 to 800 epochs. Parameter *OSRATE* in both SMOTE and GAN methods were set to 0.5 which means half of the positive training sample were generated by SMOTE or GAN.

**Result:**

**WGAN-GP successfully generated high-quality positive samples.:**

In general, the samples synthesized by WGAN-GP blended well with the original positive samples. We added additional CTST after each training epoch, and at the start of generator training, the generated samples were far from the distribution of positive samples, resulting in a CTST accuracy of 1.0. This indicated that the classifier in CTST was able to distinguish the generated samples from the positive samples effectively. By the 200th epoch, the generator showed signs of capturing the feature distribution of the positive samples, and the generated samples started to overlap with the positive samples in the feature space, leading to a decrease in CTST accuracy to around 0.7. By the 500th epoch, the overlapping region between the generated and positive samples further expanded, and the CTST result further decreased to around 0.6. After more than 800 iterations of further training, WGAN-GP demonstrated that it had learned the genetic feature distribution of the real positive samples well, with the CTST result decreasing to 0.5 and fluctuating around that value. This indicated that the generator was able to fit the distribution of positive samples. The training loss curve also showed that the generator reached its local optimum when the CTST result dropped to 0.5. Additionally, conducting independent CTST on each saved generator model revealed that over half of the generators could maintain a classifier accuracy of around 0.5. This suggested that it was difficult to differentiate the generated samples from the positive samples using the classifier. The generator indeed could generate high-quality synthetic samples.

**Synthetic positive samples generated by WGAN-GP improved the performance of our task:**

总的来说，合成的样本成功的提高了原始训练样本组合的预测性能，我们分别使用SVM,GB,GNB,RF,LR五个常见机器学习分类器，通过method章节描述的方法进行实验。每个分类其各有147个模型参与实验，我们取每个分类器的147个模型对所有基因给出的置信度平均分数进行性能评估。

**Generative adversarial data augmentation performs better than SMOTE:**

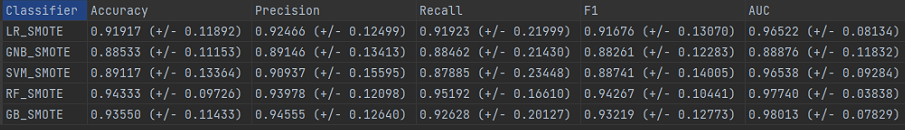
(…..)

**Influence of parameter OSRATE:**

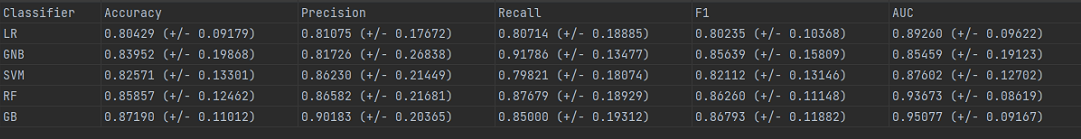
(…..)

**Influence of datablock:**

(…..)



【FIG2】



【FIG3】