Applying Conditional Generative Adversarial Neural Networks (cGANs) to Generate Realistic Microarray Gene Expression Data

[1]Lawrence Bai, [2]Madeleine Scott
lawrence.bai@stanford.edu, scottmk@stanford.edu
[1]Immunology Program, Stanford University School of Medicine, Stanford, California, USA; [2]Biophysics Program, Stanford University School of Medicine, Stanford, California, USA

Introduction

- The NCBI Gene Expression Omnibus (GEO) is a repository of microarray datasets derived from tissue biopsies or blood.
- In biomedical research, low number of observations available is due to a lack of available biosamples, prohibitive costs, or ethical reasons
- The development and usage of GANs and VAEs for omics data augmentation is scarce
- Augmenting observations with generated *in silico* samples could lead to more robust analysis results and a higher reproducibility rate
- A recently published method using conditional single-cell Generative Adversarial Neural Networks (cscGANs) used single-cell RNA-seq data to generate realistic cells of defined types
- We adopt this cGAN and apply it to sample-level gene expression data derived from GEO to improve biological analyses for rare diseases

Data and Methods

- We mined the GEO database for biopsy microarray gene expression samples from cancer patients and healthy tissue controls
- In total, we collected over 7000 samples from across more than 50 datasets in 4 different cancers (colorectal, breast, pancreatic, and lung). We use colorectal cancer (27 datasets, 2495 samplaes) as proof-of-concept.
- We use a well-established batch-normalization technique, ComBat, to remove batch effects in microarray datasets inherent to heterogeneous conditions such as different platform technologies, geographic location, etc.

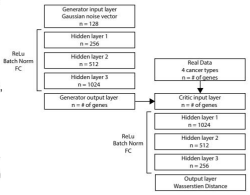

Figure 1. cGAN workflow adopted for microarrays. We use two fully-connected neural networks for our generator and discriminator. In the generator, there are three hidden layers of increasing size and vice-versa for the discriminator.

Experiments

Hyperparameter tuning

- We tuned learning rate values between 0.1 to 0.00001 logarithmically. Our final model used learning rate of 0.0001, decaying to 0.00001.
- We tuned batch size, from 2^1 to 2^7, and our final model was 2^7.

Distance metrics

- Wasserstein distance general equation:

$$W(P_r, P_s) = inf_{\gamma \in \Pi(P_r, P_s)} \mathbb{E}_{(x,y)\sim\gamma} |x - y|$$

- Wasserstein distance (Kantorovich-Rubinstein duality):

$$W(P_r, P_s) = sup_{||f||_L \leq 1} \mathbb{E}_{x \sim P_r} f(x) - \mathbb{E}_{x \sim P_s} f(x)$$

Results

Model evaluation

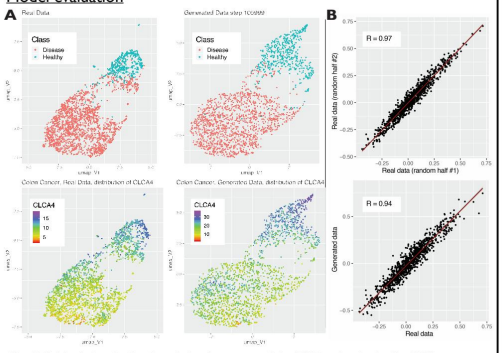

Figure 2. Model evaluation using clustering methods and gene-gene correlation. (A) Using a clustering algorithm, UMAP, we compare the nonlinear combination of features that define disease and healthy groups and compare the clustering pattern of real data with generated data. (B) We look at the correlation values of gene expression between real data and generated data, where one point is a single specific gene.

- (A) Generated data had a similar clustering pattern as real data.
- (A) Clustering was biologically relevant; here we show that CLCA4, which is highly expressed in the GI tract, is a defining feature for the formation of clusters.
- (B) When real data is compared to generated data, gene expression values is highly correlated, suggesting the GAN is retaining gene-gene network structures.

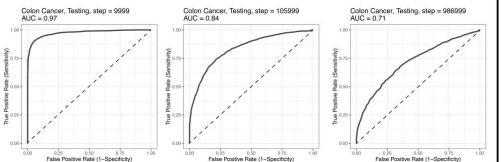

Figure 3. Model evaluation using random forest classifier to predict accuracy of discriminator during training. In the beginning of the model, the discriminator easily distinguishes generated data from real data. Over the course of training the cGAN, it becomes increasingly harder for the discriminator to correctly distinguish generated and real data, as measured by ROC.

- Used random forest classifier with 5-fold CV and ROC to measure accuracy of discriminator.
- Training cGAN over time resulted in worse accuracy of discriminator.

Results (cont.)

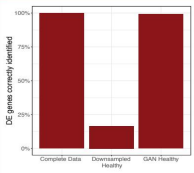

Figure 4. Model evaluation using differentially expressed genes between healthy and disease samples. Using all real data, we first calculate differentially expressed (DE) genes between disease and healthy, at a p-value < 0.00001. Downsampling the healthy samples, we lose significance in most of these genes, whereas replenishing missing healthy samples with generated healthy samples recapitulated most DE genes.

- Downsampling healthy samples reduces # of DE genes, potentially hiding biologically relevant genes.
- Replenishing with GAN-generated healthy samples restores DE genes equivalent to that from the real data.

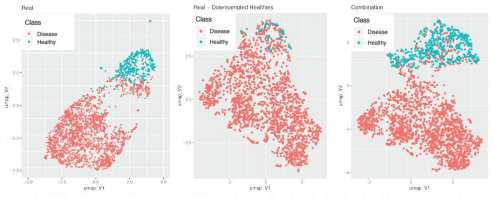

Figure 5. Model evaluation using downsampling analysis. Using clustering algorithms like UMAP, we show that downsampling of healthy samples results in the disappearance of a distinct cluster between healthy and disease groups. The replenishment of healthy samples using GAN-generated data trained using downsampled data restores clusters.

- Downsampling eliminates distinct healthy and disease samples, but generated data using downsampled data can restore the clusters
- In severely imbalanced data, cGAN can learn enough gene-gene interactions from the disease samples to infer healthy state

Conclusions/Future Directions

- Overall, our model evaluations suggest that our cGAN was able to generate real-like data that:
  1) retained gene-gene interactions and the gene expression network
  2) was difficult to distinguish from real data
  3) and could learn even with downsampling using disease samples as a reference.
- Can this be applicable across diseases? Focused on CRC, what about other cancers? Other diseases?
- Is this usable for rare diseases? Push the model with less data
- Validation needed in downstream biological analyses--perhaps some experiments focused on genes that would not have been picked up given current amount of data