

Our proposed approach consists of multiple pieces:

- Preprocessing steps to train the SVM classifier, creating decision boundaries for each cell state based on genes with non-zero variance across all unperturbed cells
- Gene embeddings generated from knowledge graphs created by [Genewalk](#) (Pathway Commons) and embedded using [TransE](#)
- Conditional Generative Adversarial Network (GAN), where a Generator and Discriminator battle to determine if a generated transcript is for a specific knockout
- SVM Classifier with RBF kernel

GANs have been shown to be able to learn complex distributions of data and generate samples that are difficult to distinguish from real samples (Goodfellow et. al, 2014). Additionally, their rich structure enables a variety of outputs to be queried, a useful property for this challenge. We therefore thought they could be a useful tool in generating transcripts, based on the relationships between genes learned from gene embeddings and the training data. A high percentage of the total 15,077 genes were found in the knowledge graph produced by Genewalk, which subsequently were embedded using TransE. This gave us confidence that our knowledge graph and embeddings would provide useful information to our GAN and a good starting point for its initialization.

To train the GAN, we first selected a random unperturbed cell as an input to the generator with the condition of the gene embedding of an experimentally tested gene knockout (random selection with replacement of the 71 experimentally knocked out genes, if they were in the gene embeddings). The Generator takes these inputs and produces expression values for 15,077 genes with the perturbation “applied” and passes it to the Discriminator, who checks it against the real transcript which was experimentally determined. As with all GAN architectures, the models are optimized in order to fool/detect the other. Over time (~300 epochs), the Generator is able to fool the Discriminator with its “fake” transcript. We allowed both the Generator and the Discriminator access to the gene embedding for the gene of interest. The trained Generator was then isolated and connected to the SVM classifier. Here, for each gene of interest in the 15,077 genes in the challenge dataset, we produced 100 generated transcripts predicted perturbed cells. These 100 transcripts were then fed into the SVM classifier and a distribution of T-cell states with that knockout applied were produced.

Although we are excited to have a working end to end pipeline, SVM classification results showed that all generated transcripts were classified as “effector” cells. This is likely due to the well-known phenomenon of “mode collapse” in GAN training. The Generator begins mapping all inputs to a single output that fools the discriminator, and as the discriminator updates to catch that one special case, the generator shifts the one case just enough to keep ahead of the discriminator. There are a variety of recent techniques for improving the stability and quality of GAN training that could be attempted, including adoption of the Wasserstein-GAN loss function, incorporation of output variance into the generator’s loss function, training the generator against an ensemble of discriminators, and using cycling “phases” to train the generator and discriminator alternately to stability. If the GAN training were stabilized, further validation would be necessary to determine how well the model generalizes to new perturbation conditions by leveraging the gene embedding information.