Declan Borcich
12/6/23
Research Paper F23

Headings:

Introduction and Summary:

In this paper, I intend to summarize what I have learned about the existing corpus of research on Gene Regulatory Networks (GRNs) and the commonly used tools for working with this kind of data, as well as the challenges faced and discoveries I have made over the course of this semester.

The shorthand read of the results: DAZZLE GRNs *do* appear to pick up gene expression structural differences between the P100/control and LPC/experimental data sets. However, there needs to be work done on the best method of application/hyperparameter selection and tuning for how to capture the correct structural information for anomaly detection. Additionally, a preliminary cross-validation stability experiment *did not* support structural stability of DAZZLE GRNs, but it was quick and dirty, and may also indicate problems with my method of GRN-KNN-Clustering because the results of the experiment mirror those of the P100/LPC tests (both experiments show a pattern of the smaller data set of the two having smaller clusters and a more even distribution of genes in those clusters). Given that, I believe that is further discussion to be had about whether my methods for structure identification/anomaly detection were the right balance of sensitive and noise-immune (in terms of bias and variance). Finally, I will conclude with a discussion of potential applications of this project and directions for further investigation.

1. Background - GRNs, DAZZLE, embeddings

Gene expression data is a foundational but complex space with broad applications to many areas of biology and medicine. GRNs are networks used to describe the co-expression or regulatory relationships between different genes in a biological system (i.e. a cell or organism), which can potentially be quite informative about how a cell is behaving in a given state. Given the noisiness and relative inaccuracy of gene expression data, GRNs are inferred or constructed from expression data, and inference methods must thus account for this noise. Single cell expression data is particularly susceptible to dropout and sparsity (a high degree of zero-valued data, coming from both true non-expression of a gene and frequent machine error in sequencing systems such as Illumina). The point of this paper is to examine the application of GRNs to anomaly detection. GRNs via DAZZLE should capture structural information about the regulatory relations of a gene system, which should be brought out by cluster methods.

DAZZLE is a novel method for GRN inference on single cell RNA data, using zero-injection to try to combat this dropout problem (DAZZLE paper). Like DeepSEM, DAZZLE's architecture is based on the variational autoencoder model. At a high level, a VAE consists of two neural net models (the encoder and decoder) (Wikipedia - VAE). The encoder maps from the input space to a latent space (the parameters for an assumed noise distribution). The decoder then maps from the latent space back to the input space, which is how the VAE is generative. Loss on the output is backpropagated through both model so as to optimize the parameters learned against the output of the decoder. DAZZLE, like other models, learns these parameters (optimally), but also inject dropout noise in the form of a masking matrix M. DAZZLE then trains and uses a perceptron classifier to determine whether zero data is dropped out by noise (artificially) or truly by the data, which is then used to regulate the sparsity of the output data in a flexible (more varied, less biased) way.

Additionally, a final and crucial component to this problem is that of embeddings and functional prediction. Another guiding reference on this project has been the GLIDER paper (Devkota et al.) This paper describes an improved method of protein function prediction (in Protein-Protein networks) that uses a combined local-global modifier to random-walk based embeddings to capture a more complete notion of distance in these networks. Seeing as that gene expression and protein function are deeply related, there is theoretical basis to suppose that using such embeddings or leveraging additional information source such as Gene Ontology labeling for function prediction could be highly valuable in identifying changes or anomalies in gene expression data. Unfortunately, results were poor using cDSD (from Cao et. al) embedding as a pre-clustering method, even with additional methods applied after, for reasons that were not immediately obvious, so GLIDER was not used.

2. Tools and Data - Cloud HPC, GPUs, Hammond Data Set

The autoencoder and decoder of the VAE model, whose training consists of a highly parallelizable set of computations (like any neural network model), lends itself to much more efficient training via GPU as opposed to CPU. DAZZLE uses Pytorch CUDA wrappers for GPU usage. GPU access was attained through the Tufts HPC cluster on NVIDIA A100. This turned out to be more complicated than expected and difficult to configure, but access was ultimately solved for.

The data used was from Hammond microglial mouse cells with P100 referring to the mouse age (control, unperturbed group) and LPC being the perturbed group.

Gene ontology labels come from Mouse Genome Informatics via Gene Ontology. Biological Process labels specifically were used (all that have "Biological Process" as a GO ancestor).

3. Methods & Process - GRN Analysis and Clustering, Information Measurement

I. KNN (25) + Agg Clustering (100)
II. Direct clustering via Community Walktrap
III. cDSD embedding + KNN

(I) General Methods and Method I - KNN + Agg Clustering:

As mentioned, the goal of this project was to determine the potential value of using GRN DAZZLE in anomaly prediction in gene expression data. Like in any ML problem or experiment,

it is essential that we have a baseline/control data set against which we can measure the data in question for potential anomalous behavior/information. (This is different than the distinction between train and test sets on traditional supervised learning problems. In this case we need experimental (LPC) and control (P100) data sets to determine how likely it is that our experimental data set is to be anomalous, regardless of what learning algorithm or accuracy metric we choose).

The process first required gathering data (Hammond) and then concatenating the matrices of this data such that all gene expression data appeared in the same column across all cells in the concatenated result, as well as finding and selecting the intersection of all genes in these data sets. We then filtered out genes that had no expression across all cells (columns) as the DAZZLE code would produce erroneous results in such cases, and also removed *cells* with fewer than 400 or more than 3000 unique genes. We then trained a DAZZLE GRN on that filtered data. The GRN could then be clustered and analyzed for structure.

Before clustering, I decided to use KNN to filter/reduce the GRNs in question. The motivation behind running KNN first was two-fold: first, to translate the GRN data into a lower dimensional space (because the GRN matrix ranges over the real numbers and does not directly represent connectivity in exactly the same sense of an adjacency matrix, which I imagined might be accurate for clustering), and second to reduce the level of noise overall (these graphs have many edges and many are probably not meaningful).

All clustering was done with the selected number of 100 clusters. The first clustering method I used was Agglomerative Clustering. Agglomerative Clustering is a bottom up distance-based hierarchical clustering algorithm. I used sci-kit learn's implementation with the connectivity linkage metric (as my connectivity was already defined in the KNN matrix), and for the KNN algorithm, I used *sci-kit learn*'s implementation with the ball-tree distance metric.

This method gives us two clusterings of gene expression links (1 for p100/control and 1 for LPC/perturbed) which then must be labeled. (The labeling process is the same regardless of clustering technique). Firstly, genes were mapped to GO labels which were then mapped to Biological Process (BP) labels, in two dictionaries. This way, I could iterate over all clusters in each clustering and collect the BP labels of every gene that was present in the cluster. This list (the list of BP labels plus appearance frequency in cluster) was used to generate a single cluster label by majority vote.

(Reference link: https://current.geneontology.org/products/pages/downloads.html)

(II) KNN + Community Walktrap:

The second method of using GRNs for anomaly detection used the igraph software library (for Python) to more directly perform clustering (community detection). This method is related to Diffusion State Distance method (DSD, by Cao et al.) referenced in the GLIDER paper.

(Reference link: https://igraph.org/c/doc/igraph-Community.html)

(III) cDSD Embedding - With and Without KNN Clustering:

I was interested in trying DSD embedding directly (which is closely related to GLIDER) due to the promising results from GLIDER on PPI Networks. This did not turn out as well as I expected--the model generated one giant cluster in both graphs that contained almost all of the genes. I tried DSD embeddings with and without KNNs, and used both Community Walktrap (CWT) and Agglomerative Clustering methods and received similar results each time.

4. Results

(GRN DAZZLE - graph structure; KNN + Agg, CWT, cDSD + KNN)

Method I. (KNN + Agglomerative Clustering):

Preliminary results for KNN + Agglomerative Clustering showed changes across p100 and LPC groups (see Images: Figures 3 + 4). At this point, it became clear that measuring changes or differences between our graph clusters would be far from easy. In Figures 1 and 2, I had not yet removed the largest cluster labels; Figures 3 and 4 have these removed. As you can see, the shortened BP labels are quite different, and the cluster sizes are generally much smaller with more evenly distributed numbers of genes in the LPC group, indicating structural differences between these GRN models.

Method II. (KNN + Community Walktrap Clustering) (*see Figure 5*):

I did a little deeper analysis on the Community Walktrap method. With this method we also see marked differences between p100 and LPC groups in terms of clustering and connectivity. In Figures 5 and 6 (and 7 + 8, filtered for top 4), we have a similar but less exaggerated margin of distribution between the P100 and LPC KNN graphs, where the LPC graph is more evenly distributed with smaller cluster sizes (see X-axis) and different majority vote labels.

Figure 9 shows the cluster locations (labels) of genes from P100 clusters as they appear in the LPC clustering for this algorithm group. If this distribution were totally random, we would see these genes from the P100 cluster (6, 9 and 12) randomly distributed amongst LPC clusters; however, they end up mapping to a surprisingly small number of clusters such that many of the genes stay in the same group. This suggests that there is some structure that is maintained between the P100 and LPC groups, and some that is altered. We might benefit from more statistically rigorous analysis on this (i.e. assume that the graphs are organized randomly and determine if the similarity/difference between them is more likely due to genuine structure and information capture than chance).

Differences can also be seen in highest-degree node connectivity of the KNN GRN graphs. This was just a quick check to see if the most highly connected notes in the GRN-KNN graphs were similarly connected between the experimental and control groups. In this measure, I took the Top 10 genes in the KNN graph taken from the P100 GRN and checked those genes for connectivity in the LPC KNN graph; this was also done for the Top 20 genes (see Figures 10 and 11; 12 and 13). This data also seems to support the idea that meaningful structural alteration is being captured between these groups.

Finally, Figures 14 and 15 are the complete p100 and LPC KNN graphs in a circular layout. Higher quality images can be provided with additional zoom. Unfortunately this layout did not order the circles consistently with respect to Gene ID. number, so these images are not particularly useful.

Method III. cDSD Embedding With and Without KNN:

My expectations about this method were not reflected in the results. Running this method resulted in one very large cluster containing almost all genes (99%) (Figure 16). The top 20 genes in the P100 graph are highly connected (Figure 17, 18), while those genes are interestingly all pointing to a sink node of gene number 3546 ('Hexb', related to a specific enzyme's production in microglia).

Additionally, I tried replicating results similar to those in Methods I and II by not performing KNN and using both Agglomerative and CWT clustering, and got similar "clumpy" clustering. I would guess that the DSD embedding is somehow connecting all the gene far too closely in the embedding. If that is the case, any method of doing additional clustering on top of that is going to say that most genes are highly related to most other genes.

5. Further Considerations - Model Stability (Cross-Validation & Bootstrapping)

One point of interest in examining the quality of the DAZZLE model is the model's stability or consistency within a dataset. This can be considered in two different senses: 1) stability of the model's output construction within a data set, and 2) stability of the model's predictions with respect to the anomaly prediction application.

In the first sense, the model should produce GRN matrices that are similar in structure when trained on the same data. This could be performed by bootstrapping the model against one data set (Slonim Slides - CS169), which is a reliable method. One could generate DAZZLE GRNs on a number of randomly sampled subsets of the data, i.e. the Hammond data and compare the graphs structures directly by an algorithm such as EDIT-DISTANCE. The unfortunate limitation for this point is that DAZZLE takes approximately 6-8 hours on a dataset the size of the Hammond P100 dataset, which would mean that a substantial bootstrapping would 100s of training hours.

The stability test methodology I did here (as a preliminary litmus) was to simply partition the P100 data set in to "train" (set A, 67% of the cells) and "test" (set B, 33%) subsets with sci-kit learn's API and train GRNs on them and filter/reduce with KNN. Interestingly, the networks produced had marked differences in structure and cluster size (Figures 19-22), very similarly to the difference between the original P100 and LPC data sets. The commonality between those was also that the control/train group was substantially larger than the experiment/test group (P100 was 3x the size of LPC, "train" was 2x test). This is evidence that the DAZZLE + KNN method is overly sensitive to random changes, or that is highly dependent on having sufficient data for consistent structural capture.

It is important to note that this is not a sufficient measurement of the stability of the DAZZLE-GRN model; bootstrapping would be a more reliable method. Additionally, it would be an important next step to choose optimal methods and hyperparameter options to apply these

GRNs to anomaly detection so as to perform an additional controlled experiment for validation (so as to avoid "data leakage" or tuning the model to closely to get our desired result). I have found a few data sets in *scPerturb* that may be good options for this (Frangieh 2021, Benevolenskaya 2021).

6. Conclusions

My work this semester has demonstrated to me how complicated and difficult clustering with gene expression data can be. While biological data already tends to be noisy, the lack of structure inherent to unsupervised learning methods like clustering adds to the challenge. Having to work on a remote shared HPC system is also surprisingly challenging (though it is more than worth the training time gains from GPUs). I also very much enjoyed learning about the application of VAE models in DAZZLE and the numerous embedding, distance metrics, and clustering methods in the field at present.

When a GRN is generated and clustered, it is not immediately clear what the best metric is to compare the clustered graphs because they are going to be different along so many dimensions, namely i) the identities of the genes in each graphs' clusters, ii) the sizes of the clusters, and iii) the functional label for the clusters. EDIT-DISTANCE can capture all of these in one metric in some sense, but all structural information about the clusters is destroyed in the process, which fails to take advantage of the additional information that functional labelling of the clusters can supply.

Another consideration that needs to be dealt with is what the actual best method of interpreting or utilizing the GRNs should be. As a novice in this field, I assumed that a KNN filter applied to the GRN would produce more stable results, but I have not done a thorough analysis of this in relation to other potential methods (clustering directly on the GRN, etc.).

In conclusion, while it certainly appears that there is structure or signal being identified and captured by DAZZLE which can be used to positively identify anomalous gene expression in biological systems, more work needs to be done to determine under what conditions DAZZLE is most stable and reliable, and to configure and hyper-parameterize the model for this application for best results.