CS230

Predicting Gene-Coexpression from a Convolution Neural Network using Single-Cell Sequencing Data: Final Report

Category: Healthcare

Mahdi Moqri, PhD
Department of Biomedical Informatics
Stanford University
moqri@stanford.edu

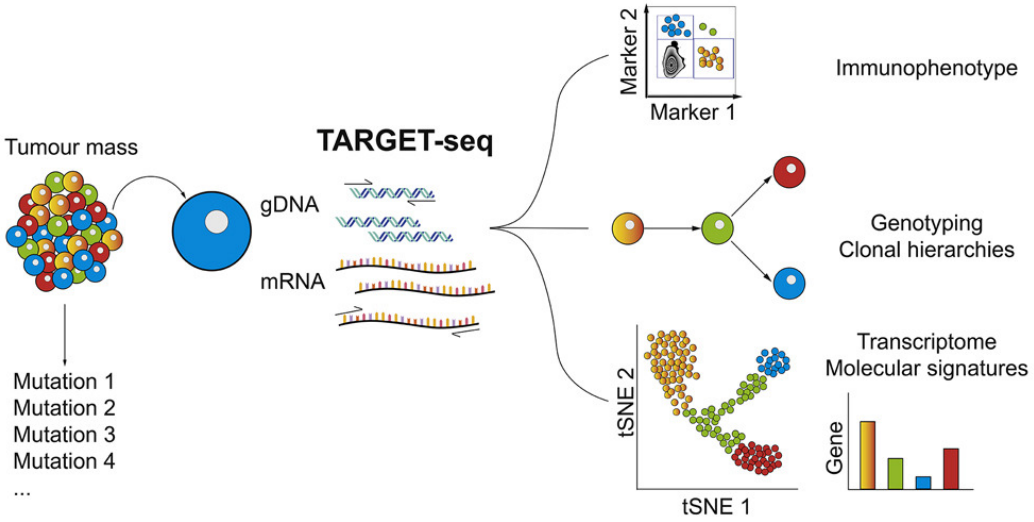
Sierra Lear
Department of Bioengineering
Stanford University
slear@stanford.edu

Mustafa Hajij
KLA Corporation
mhajij@stanford.edu

1 Introduction

Single-cell sequencing, heralded as "Breakthrough of the Year" in 2018 [1], is revolutionizing medicine by enabling researchers to explore biological samples at the level of individual cells [2]. The increased resolution of generated genomics data has allowed us to unravel the heterogeneity of cells in samples, leading to many ground-breaking discoveries, such as finding metastasis-initiating cells in breast cancer [3].

Figure 1: Single Cell Analysis - Rodriguez-Meira et al. 2019



CS230: Deep Learning, Winter 2019, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

However, one of the most exciting advances of increased single-cell resolution is the possibility to better infer gene-gene relationships. By utilizing gene expression data from individual cells, rather than a bulk average from a whole cell population, researchers can better predict causal relationships between different genes or even reconstruct gene-based pathways. TALK ABOUT SIMPSON'S PARADOX, HOW THE SIZE AND INCREASING AMOUNT OF SC DATA MAKES IT IDEAL FOR DEEP LEARNING OVER MACHINE LEARNING AS DATA SETS CONTINUE TO GROW AND GROW AND GROW.

Predicting gene co-expression has broad clinical applications. By building a model that is able to accurately predict gene co-expression, we could predict new transcription factors and genes in the future, opening up new therapeutic targets to several diseases.

2 Relevant Work

Deep learning applied to single-cell genomics data is still a relatively unexplored field. In part, this is because deep learning models fail to outperform classical machine learning on specific tasks using single-cell RNA-sequencing (scRNA-seq) data, such as predicting cell type [CITE 5 from proposal, CITE 6 from proposal]. In part, the difficulties are thought to be caused in part by high level of noise in the input data, the expression of different genes in a cell.

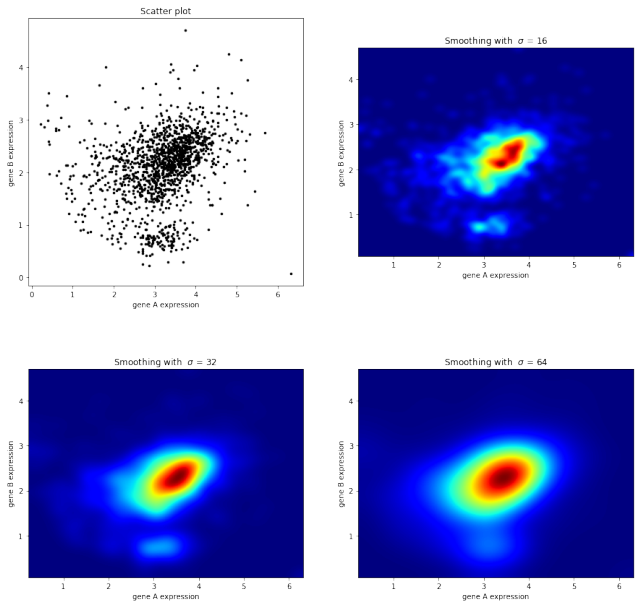
However, one paper reports that by pre-processing scRNA-seq data into an image, they were able to utilize the strength of a convolutional neural network (CNN) for image-like data to mine gene-gene relationships [4]. FIGURE OUT WAY TO SPIN SOMETHING NOVEL FROM THIS.

3 Approach

3.1 Data Preprocessing

CNNs are traditionally thought to work best with image-like data. In contrast, raw scRNA-seq data has an organized, tabular structure where numerical values are used to document the expression of different genes within a cell . To take advantage of the strengths of CNNs, we first preprocess our data into a "heatmap" which compares the expression of two different genes to each other. These visual heat maps are the inputs into our CNN.

Figure 2: Scatter plot and heat maps comparing expression of two genes within a pancreatic cancer scRNA-sequencing dataset. Heatmaps are shown with different σ, which determines how smooth the heatmap appears.



We have already used this data preprocessing pipeline to create heatmaps for different genes for highly expressed genes within a pancreatic cancer dataset (see Figure 2.)

Additionally, we recently also processed a dendritic mouse cell line dataset where relationships between different genes are already known [4]. This dataset will serve as our labelled dataset, as the additional information allows us to label the genes corresponding to each heatmap as either co-expressed (y = 1) or random (y = 0). The preprocessing pipeline with both the pancreatic (rnaseq2image.ipynb) and labelled dataset (naseq2image_cnnc.ipynb) can be found at the cited URL [5].

3.2 CNN Models and Design

For our experimentation we choose to train two convolutional neural networks.

The first neural network that we built is very similar in architecture to the typical classification CNN networks such as AlexNet [6]. More specifically, the input for the neural network is input layer of $32 \times 32 \times 1$, and this is the size of the heatmap generated by scRNAseq expression data. This input then goes through a sequence of convolutional and pooling layers followed by a flatten layer and a softmax layer. The number of convolutional layers are 10. In the final two layers, the convolutional layer is flatten and two dense layers are utilized. The final layer is an softmax layer.

For our experimentation we also tried working with other neural network architectures. This was our attempt to explore various hyperparameters in order a better results. More specifically, in our implementation we explore with a neural network with a deeper convolutional architecture while keeping in mind adding more dropout layers to prevent the neural network from overfitting.

Given we are working with a binary classification task where 0 equals a pair of random genes and 1 equals co-expressed genes, the loss function we are attempting to minimize is cross-entropy loss:

$$L(\hat{y}_i, y_i) = -\sum_{i=1}^{m} y_i log\hat{y}_i + (1-y)_i log(1-\hat{y}_i), \tag{1}$$

where y is the ground-truth label and $\hat{y}$ is the predicted classification.

Our repository is available here "https://github.com/moqri/deep_cell.git".

3.3 Initial inspection of the data

One of the interesting features of the data is that it is highly unbalanced. The vast majority of the data consists of examples that are labeled 0, 98% of the data, and only 2% are labeled negative. Given the unbalance nature of the training data set illustrated above our choice of the metric is "balanced-accuracy" which is equivalent to recall.

Another route that we are planning to implement to encounter this difficulty is to implement This a different loss function different from the classical cross entropy loss given in 1. More specifically. The *weighted cross entropy* given in the following formula:

$$L(\hat{y}_i, y_i) = -\sum_{i=1}^{m} w_i^+ y_i log\hat{y}_i + w_i^- (1-y)_i log(1-\hat{y}_i), \tag{2}$$

will most likely yield better results for our classification task. Here w_i^+ and w_i^- are the weight associated with the example i.

4 Experiments/Results/Discussion

4.1 Choosing hyperparameters and models

Our goal was to test each of the three possible neural net architectures using a number of hyperparameters associated with the pre-processing or development of the final heatmap. These included σ, the smoothing factor to determine how smooth the heatmap appeared, and bin size (DESCRIPTION OF BIN SIZE). DISCUSS WHY WE THOUGHT THESE WOULD BE IMPORTANT–DUE TO RESOLUTION. Furthermore, we also planned to test α, the learning rate for the model.

Regardless of hyperparameter and neural net architecture, we were unable to teach any of our models to distinguish between different heatmaps. Instead, regardless of the number of epochs trained, accuracy and loss never changed from the first epoch. Furthermore, the accuracy always reflected the ratio of majority and minority class present in the training and test data. This result suggested that our model learned to always output the majority class label (or a random label in the case of perfectly balanced training data). Indeed, the evaluation metric XXX of 0 of every trained model confirmed that our model learned to always output the exact same label, regardless of input image.

PICTURES OF UNCHANGING ACCURACY AND SCORES REFLECTING PICKING MAJORITY CLASS ONLY.

To address this issue, we:

1. Down-sampling to create a perfectly balanced dataset –go into more detail, creating narrative for each of these parts, include visualizing heatmaps as part of it.
2. Changing how we selected gene pairs from our original dataset
3. Choice of pre-processing hyperparameters, namely σ and bin size, to increase resolution and information in heatmaps

4.2 Visualizing Data Preprocessing

When it was clear that our models could not distinguish any salient features to help categorize different input images, we next decided to visually inspect heatmaps belonging to both classes as well, to see if we as humans could detect any differences.

5 Conclusion/Future Work

A different data preprocessing approach, such as XXXX or XXXX, may result in better performance due to XXXXX.

Future work includes: new preprocessing model, try hyperparameter search including for the architecture itself (learning rate).

After working model, can use it to predict new co-expressed genes in other datasets, mention how we performed data preprocessing on a pancreatic cancer dataset that we planned to use assuming that we could get a working model.

References

[1] National Foundation for Cancer Research. "Breakthrough: Single-Cell Sequencing." https://www.nfcr.org/blog/breakthrough-single-cell-sequencing/

[2] Science in the News, Harvard Graduate School of the Arts and Sciences. "The Single Cell Revolution: Zooming into human health & disease." http://sitn.hms.harvard.edu/flash/2017/single-cell-revolution-zooming-human-health-disease/

[3] Lawson, Devon A., et al. "Single-cell analysis reveals a stem-cell program in human metastatic breast cancer cells." Nature 526.7571 (2015): 131. https://www.nature.com/articles/nature15260

[4] Yuan, Ye & Bar-Joseph, Ziv. "Deep learning for inferring gene relationships from single-cell expression data." PNAS 116 (52) 27151-27158 (2019). https://www.pnas.org/content/116/52/27151.short

[5] CS230, Deep Cell: https://github.com/moqri/deep_cell/tree/master/CNNC

[6] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.