Cell-Conditional Generative Adversarial Network

by

Xi Zhang

BSc. McMaster University, 2019

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Bioinformatics)

The University of British Columbia
(Vancouver)

August 2021

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Cell-Conditional Generative Adversarial Network

submitted by **Xi Zhang** in partial fulfillment of the requirements for the degree of **Master of Science** in **Bioinformatics**.

Examining Committee:

Wyeth Wasserman, Medical Genetics, UBC
Co-Supervisor

Sara Mostafavi, Computer Science, University of Washington
Co-Supervisor

Raymond Ng, Computer Science, UBC
Supervisory Committee Member

Steven Jones, Bioinformatics, UBC
Supervisory Committee Member

Paul Pavlidis, Bioinformatics, UBC
Examination Bioinformatics head

Abstract

With single cell sequencing advances, research has increasingly focused on understanding cell-specific gene regulation mechanisms. However, single cell sequencing data are often noisy and the amount of sequence obtained from rare cell types small. Simulation can be a powerful approach to aid understanding when data is limited, both because the process used to generate such data can provide mechanistic insights into cell-specific regulation and the data produced can augment analysis methods development. We constructed and optimized a stand-alone cell-conditional GAN (CCGAN) to simulate cell-specific ATAC-seq data. We trained our model on published single cell ATAC-seq (SCATAC-SEQ) data that had been produced with different protocols on embryonic mice forebrain and adult mice brain. The CCGAN generated sequence was correlated in both Transcription Factor (TF) binding motif composition and positional distribution with the experimental SCATAC-SEQ.The CCGAN simulator was able to learn important cell-specific signals amidst noise. The CCGAN architecture holds broad potential for single cell regulatory data simulation beyond ATAC-seq, such as for ChIP-seq or epigenetic properties.

Lay Summary

Single cell technologies allow data to be obtained from individual cells, which reveal differences between cell types and new types of cells we did not know previously. However, the data obtained from single cell technologies is usually noisy and rare cell signals remain elusive. Understanding patterns in this complex data can help us understand disease mechanisms better and can lead to improved therapies. In order to understand the data from a single cell technology, we developed a machine learning model based on General Adversarial Networks (GANs) that use computer artificial intelligence to learn the properties of the data. The DNA sequence patterns generated by the GAN were found to be similar to DNA sequences identified in published experimental studies. In the future, the ability to identify cell-specific sequences using artificial intelligence could allow new therapies that would be designed to only treat the types of cells involved in a patient's disease.

Preface

The thesis includes two complementary introductory sections. The first is intended to provide the reader with a more complete understanding of the background, while the second is the expected introduction for a peer-reviewed manuscript for the thesis research. There is some redundancy between these sections, with the intention that the first introductory section allows readers with less domain-specific experience to understand the thesis document.

Table of Contents

List of Tables

List of Figures

Glossary

CCGAN cell-conditional GAN

CNN Convolution Neural Network

EMD Earth Mover's Distance

EMP Erythroid Myeloid Progenitors

GAN Generative Adversarial Network

PCA Principal Component Analysis

SCATAC-SEQ single cell ATAC-seq

TF Transcription Factor

VAE Variational Autoencoder

Acknowledgments

I would like to thank my MSc. supervisors Dr. Wyeth W. Wasserman and Dr. Sara Mostafavi for their continuous support and guidance in this project.

As well, I would like to thank everyone in the Wasserman and Mostafavi lab for their great advise given during lab meetings, and their friendship.

I would also like to thank the members of my committee: Dr. Raymond Ng, Dr. Steven Jones for their insight in making this project better.

In the end I would like to thank my family and my partner who has always been there for me, especially when it gets difficult.

This project was funded by NSERC CREATE and NSERC CGS-M awards.

Chapter 1

Thesis Background

1.1 Generative Adversarial Network

Generative Adversarial Network (GAN) is a type of generative model that involves two components, a generator and a discriminator. The generator generates data using a random vector z, and the discriminator is responsible for discerning the real and fake data generated by the generator. The intuition behind GAN lies in game theory in which two players help the model reach optimality through competition. Generative models such as GAN flourish with little data in contrast to other discriminative models, making it appropriate for modelling single cell data. Compared to other generative models such as Variational Autoencoder (VAE), the samples generated by GAN appeared to be higher quality [8, 9]. As well, GAN architecture is less constrained than VAE and can have objective functions in a variety of forms for different training goals. Whereas VAE focuses solely on minimizing the variational lower bound, GAN can modify its objectives to better model distributions that have irregularities and may be discontinuous on certain regions. Similar to VAE, GAN contains a "latent space" which is the space where the random vector z is generated.

1.2 Wasserstein GAN Gradient Penalty

Wasserstein GAN (wGAN) is a special structure employed in the GAN to increase its stability. The traditional GAN is prone to diminishing gradients from the generator since it is easier for the discriminator to differentiate the samples when the generator has only been trained for a short amount of time. wGAN utilizes Earth Mover's Distance (EMD), expressed as:

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} E_{(x,y)\sim\gamma}[||x-y||]$$

EMD measures the similarity between distributions by how one distribution can become the other by moving "dirt", which refers to the physical part of the distribution. The distance is calculated via the amount of "dirt" moved multiplied by the distance it is moved. It differs from other distance metrics such as Kullback-Leibner (KL) divergence, Jensen-Shannon (JS) divergence and Total Variance (TV). There are two major benefits using EMD, which are its continuous differentiable nature and resistance to mode collapse, a common form of GAN failure. EMD has an important property that makes it an attractive objective function, which would decrease the difficulty in training GANs. Arjovsky et al., (2017) proved that under low dimensional manifolds, EMD is continuous, and provides a gradient that can be used whereas other distances such as JS divergence cannot [2]. Its continuous nature means finding the optimum would be possible and less tricky. As such, wGAN is resistant to mode collapse, a common GAN training problem due to discriminator saturation, in which discriminator has no more to learn. This is caused by diminishing gradients in the GAN training loop, in which the discriminator gives too little feedback to the generator so the generator cannot improve. Mode-collapsed GAN generator produces the same or highly similar samples that fail to reflect the data space. Regular GANs have diminishing gradients whereas wGAN optimal discriminator or critic converge to linear gradient.

However, wGAN has some shortcomings that prevent it from generating the best sample possible. Lipschitz constraint, expressed as

$$||f(x_1) - f(x_2)|| \leq k||x_1 - x_2||$$

was implemented to guarantee the norm of the gradient is bounded. It alleviates the vanishing gradient problem and stabilizes GAN training. To satisfy the Lipschitz constraint, wGAN originally had gradient clipping implemented. Gradient clipping constrains the weight of the model to a fixed interval, and has negative impacts for GAN training. With weight clipping, the discriminator may take longer to reach optimality. Some measures have been implemented including increasing the iteration the discriminator is trained compared to the generator. The state of the art implementation for wGAN currently is wGAN- Gradient Penalty (GP) [11]. In the GP version, weight clipping was replaced by gradient penalty to allow GAN to learn more complicated distributions and generate high quality samples. GP satisfies the 1-Lipschitz constraint by limiting the norm of the gradient to 1. The magnitude of GP can be adjusted by a λ parameter.

$$L = E_{\tilde{x}\sim P_g}[D(\tilde{x})] - E_{x\sim P_r}[D(x)] + \lambda E_{\hat{x}\sim P_{\hat{x}}}[(||\nabla_{\hat{x}} D(\hat{x})||_2 - 1)^2]$$

1.3 Gumbel-softmax

Gumbel-softmax distribution was used in the generator for one-hot encoding of the output. The variable generated from the distribution is differentiable and is an essential component to generate one-hot encoded sequences directly from the generator. To generate discrete output that is one-hot encoded, a discrete or stochastic node is needed as opposed to deterministic or continuous nodes. Contrasted with deterministic and continuous nodes, stochastic or discrete nodes cannot be back-propagated through. Back-propagation is needed to update the neural network and make it "learn", however. Therefore, a reparameterization trick was used in gumbel-softmax to turn the otherwise discrete node deterministic, and make back-propagation possible. This reparameterization trick is a technique used to express an otherwise non-differentiable variable in a form where it can be differentiated. The gumbel-softmax distribution itself is not categorical, but can be smoothly formed to a categorical distribution. The second attribute to the gumbel-softmax involves a temperature (T) parameter that controls the underlying distribution, which is a mix between categorical argmax and uniform. When T=0, it is a discrete distribution, as T increases to infinity, the distribution becomes an uniform

distribution. During the training process, T approaches 0 but never reaches it to prevent exploding gradients, in a process called annealing. During testing, however, the gumbel-softmax only produces discrete signals. Thus it may introduce a small amount of error from the difference between discrete and continuous output, making error during evaluation larger than expected.

1.4 Data background

The type of data utilized is single-cell ATAC-seq. ATAC-seq categorizes the open region of the chromatin, which may be available to be bound by transcription factors for initiating the transcription process. It utilizes the hyperactive Tn5 transposase to preferentially cut open regions, forming fragments with a maximum length of 1000bp and mode length of 100bp to 200bp long. From a functional aspect, ATAC-seq categorizes promoter regions, enhancer regions and many intronic and intergenic regions. "Footprint" patterns can be found from alignments which signifies the region where the transcription factor is bound. As well, the enrichment for motifs are often found in the centers of detected peaks. Most data use 500bp as the arrangement for peak range for capturing the maximum amount of information. ATAC-seq has been used to detect Transcription Factor (TF) motifs and interactions [35].

Single cell sequencing aims to reconstruct a higher resolution sequence space for learning cell-type specific signals. In contrast, bulk sequencing may miss signals from rare cell types as noise and only extract the most prominent pattern from the cell with the largest population. Single cell sequencing can be combined with RNA-seq, ATAC-seq and other types of sequencing. There are several approaches in single cell sequencing, including single nucleus sequencing and droplet-based. The downside of single cell sequencing is a high amount of noise in the data, and rare cell type signals may be affected by the noise [15].

Chapter 2

Introduction

Understanding of cell-specific properties is expanding rapidly as a result of sequencing technologies that allow characteristics of individual cells to be profiled. Diverse genomics methods have been applied to single cells, ranging from RNA expression to chromatin accessibility to TF binding. Concurrent with the sequencing-based methods arrival, a dramatic expansion of machine learning capacity, deep learning, has allowed the identification of subtle patterns in vast data. The intersection of data generation and data analysis advances has created opportunities for insights into the formation and properties of human tissues composed of heterogeneous mixtures of cells.

Much single cell research focuses on understanding the regulatory mechanisms governing gene transcription, in part under the expectation that understanding the processes by which cells transition between states will allow improved engineering for the production of cells and tissues for biomedical applications. A critical bioinformatics step in such analysis is the detection of motifs in the active regulatory regions in a cell. Such motifs represent the target cis-regulatory sequences of sequence-specific DNA binding transcription factors. The signals can be difficult to detect and this type of analysis is sensitive to noise.

Despite great promise, there are shortcomings of single cell sequencing to overcome. These challenges include low depth of sequences generated per cell and a high level of noise that makes signal detection and analysis challenging. Deep learning methods have proven to be powerful tools for overcoming noise to focus

on informative signals. Within the field, generative models can capture these informative signals to produce output that resembles the data under study - to simulate the data. There are two major types of generative deep learning models: Generative Adversarial Network (GAN) and Variational Autoencoder (VAE) [9, 20]. Based on game theory principles, GAN has two components - a generator and a discriminator - that compete with each other in learning the properties of the data. VAE functions similar to a data compressor, projecting data onto a low dimensional latent space and subsequently extracting it. Results from computer vision research suggested that generated samples from VAE were less detailed than GAN [9, 24]. On the other hand, GAN was much more difficult to train due to it being prone to diverge or reach mode collapse.

It is expected that by establishing the capacity to generate artificial sequences that are indistinguishable from input sequences, a generator will have identified aspects of the input sequence relevant to their function and will have removed aspects relevant to the noise. GAN allows class-specific data generation. In contrast, VAE does not readily allow generation of conditional data from its latent space due to shared latent space between different classes [29]. The generation of synthetic biological sequences is becoming increasingly useful. Most described synthetic sequence generators produce short sequences (less than or equal to 200 units in length). A leading model for synthetic sequence generation is FBGAN, which contains a feedback loop structure combining GAN training with a convolutional neural network for generating antimicrobial peptides [12].

Previous applications of sequence generating GANs were mainly for generating novel functional sequences for experimental characterization.For example, Killoran et al., (2017) and Linder et al., (2020) used GAN to generate synthesized splice sites [19, 26]. Linder et al., (2020) used the structure to generate alternative forms of Green Fluorescent Proteins (GFPs). When combined with high throughput assays, GANs present an attractive way to design and find proteins or DNA sequences with desired function and properties. GANs can also expand augment limited datasets. Sanfort et al., (2019) used data generated using their GANs for training Convolutional Neural Networks (CNNs) and found CNNs trained with GAN-augmented data were better generalized [32].

In this report the focus is placed upon single cell ATAC-seq (scATAC-seq) data,

which reveals locations within the genome that are accessible within the nucleus of a cell and therefore more likely to function as regulatory sequences. Like other methods, recovered DNA is sequenced, mapped onto reference genomic coordinates and a count matrix is generated containing the number of reads recovered for each position. As noted, single cell techniques are noisy and scATAC-seq potentially more so. Therefore substantial effort has been made to develop bioinformatics methods to denoise such data, such as SCALE and AtacWorks [22, 34]. SCALE operates on a count matrix and does not take into account the primary sequence of the DNA for interpretation. Similarly, AtacWorks operates on the count level. Both methods are applied to experimentally generated data. GAN-based scATAC-seq data processing models exist, such as scDEC, which are also count-based and therefore appropriate for specified regions along a chromosome [27]. Thus, an approach to scATAC-seq data synthesis that incorporates primary sequence properties into the model may complement existing approaches.

In this report we set out to develop a cell class-specific generative model for scATAC-seq data. We combined conditional GAN (cGAN) architecture optimized for generating one-hot encoded sequences and stability, creating the cell-conditional GAN (CCGAN). As a proof of concept, we trained two ccGANs on the data from Preissl et al., (2017) and Lareau et al., (2018), respectively, to demonstrate its ability in producing biological signals and adaptability in sequencing methods [23, 31]. Here we present a cell-specific sequence generative model CCGAN that is able to denoise single cell ATAC-seq while preserving biological characteristics of sequences such as motif presence and locations.

Chapter 3

Method

3.1 Data curation

The processed embryonic mice forebrain snATAC-seq data from Preissl et al., (2018) was used with cell type information as indicated [31]. Summarizing the original report, tissue was acquired from frozen mouse forebrain samples. There were 12,733 cells in total, with eEX2 having the most number of cells and EMP having the least number of cells in the category. To reduce spurious peaks, only regions that are open in more than 1% of the cells are counted as positive peaks. A binary matrix is formed after filtering. The cell type is fed into both generator and discriminator, separately from the DNA input. For the droplet based scATAC-seq data from Lareau et al., (2019), derived from adult mouse brain, positive peaks are filtered to include only those that are overlapped by a peak in at least 50% of the cells of the indicated subclass [23]. This threshold is implemented to accentuate unique DNA sequences of each particular cell type. It is confirmed by having the same cell types clustering together via hierarchical clustering. The data are then one-hot encoded and five percent of data are used for validation and testing, for both sets of single cell ATAC-seq. The two sets of data were used to train separate GANs since they profiled different stages of mouse development and may therefore contain different motifs.

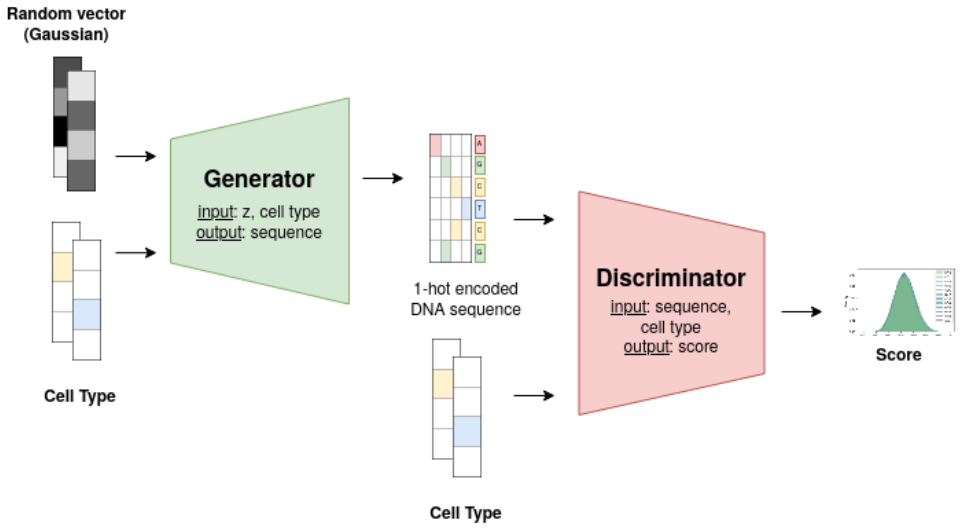

Figure 3.1: Cell-conditional GAN architecture involves a wGAN-GP related base model with cell type conditional to generate sequences specific to a particular cell type.

3.2 GAN structure

The GAN is a neural network consisting of a generator and a discriminator [9]. The cost function within the implemented GAN is as follows:

$$L = E_{\tilde{x}\sim P_g}[D(\tilde{x})] - E_{x\sim P_r}[D(x)]$$

Specifically, we used wasserstein GAN gradient penalty (wGAN-GP) [2, 11]. The loss specific to wGAN-GP is:

$$L = E_{\tilde{x}\sim P_g}[D(\tilde{x})] - E_{x\sim P_r}[D(x)] + \lambda E_{\hat{x}\sim P_{\hat{x}}}[(||\nabla_{\hat{x}} D(\hat{x})||_2 - 1)^2]$$

The gradient penalty λ is set to 10. The generator accepts an input of z, a random vector of size 128, and c, the class or cell type of the sequence. The output of the generator is a one-hot encoded DNA sequence. The structure of the generator consists of two fully connected linear layers for processing the random vector z and the class information, respectively. Residual block structures are then used for intermediate layers to transfer information between layers with minimal loss

of information. The final activation function used is gumbel-softmax for generating one-hot encoded sequences [14]. The model was constructed and optimized based on FBGAN, using the same residual block and Gumbel-softmax structure and hyperparameter [12]. The discriminator accepts one-hot encoded sequences and its class or cell type. From there, it employs a similar structure as the generator. However, the activation function is a linear function, as a wGAN [2]. This is intended to prevent gradient diminishing in GAN, causing the generator to stop learning because the discriminator is able to distinguish real and generated samples early on in the training. wGAN also has several characteristics such as training discriminator more than generator for one iteration. As the GAN training may be unstable, the one-sided label smoothing techniques are implemented for increasing stability. Label smoothing reverses a small percentage of data labels from real to fake, which would bring the real and the generated distribution closer together. This may also make it more difficult for the discriminator to learn the distribution to prevent diminishing gradients. Multiple values of learning rate were tested ranging from 1e-3 to 1e-6. An overview for the model can be seen in figure A.1. The model was implemented using Pytorch v 1.7.1 [30]. A more detailed structure figure is included in the Appendix (figure A.1). The code is available at https://github.com/wassermanlab/ccGAN.

3.3 Adjunct Convolution Neural Network (CNN) training, structure and feedback

The adjunct CNN is included as an external neural net for evaluating the performance of the network. The input to the CNN is the DNA sequence and the output is the predicted class or cell type that sequence belongs to. The CNN consists of convolutional units that can detect patterns in a space-invariant manner [21]. The architecture of the adjunct CNN is similar to other CNNs that detect regulatory regions such as Basset [16]. It has three convolutional layers each followed by a max pooling layer and two linear layers at the end. The network is given both positive datasets and negative datasets for training. The negative dataset used in this case is generated by BiasAway, a tool that generates k-mer matched background sequences as a negative control [18] using the setting of k-mer shuffle with sliding

window. The CNN is trained with early-stopping to prevent overfitting. A feedback loop is incorporated between the CNN and the GAN after training both separately. For each epoch, top quality sequences generated by the generator and evaluated by the CNN are added to the training data for the GAN.

3.4 GAN result verification

The sequences generated by the generative adversarial network are verified in three aspects. The first aspect is duplication of generated sequences. As GAN training can be unstable and result in mode collapse, in which the generated data are very similar to each other, the diversity of sequences must be ensured. BLASTN 2.5.0+ was used to compare generated sequences to another group of generated sequences [1, 5]. The parameters of BLASTN include an e-value threshold of 1e-1, and dust=no, task=blastn-short to account for short alignments. As a negative control, validation sequences were compared.

The second aspect is memorization. Overfit neural networks can memorize the data instead of learning underlying distributions and properties. BLASTN is used to test for network memorization via blasting generated sequences to training sequences. The same parameter and e-value threshold was used for this instance as above.

The third aspect is the verification that the sequences contain biologically significant properties. For mimicking ATAC-seq results, TF binding motif presence and composition could be similar to the experimental data, as could the positioning along each sequence. Such properties for the top-ranked TF binding motif for the experimental data was examined. For motif enrichment analysis, Analysis of Motif Enrichment (AME) 5.3.3 from the MEME SUITE was employed to compare the frequency of motifs between the foreground sequences and a control set of sequences provided by AME itself [28]. The JASPAR 2018 Vertebrates database of TF binding profiles was used as the motif collection within the AME web service. AME parameters include average odds score for scoring method and Fisher's exact test for testing significance. The negative control was generated by AME using random shuffle preserving 2-mer frequency. The p-value limit was set to be 0.01. The motif composition was compared between the generated sequence control and the

top-ranked sequences for each particular cell type. Motif enrichment pattern was evaluated using CentriMo 5.3.3 from the MEME SUITE [4]. CentriMo detects motif enrichment locally for specified motifs (again the JASPAR 2018 Vertebrates database). The parameters for CentriMo were set to default parameters and scores. CentriMO is set to test for enrichment of motifs in the center of the sequence and assess its significance via a binomial test. For motif presence, a targeted search for specific motifs of interest is used via MEME SUITE FIMO [10]. This is conducted when a certain motif was mentioned to be enriched in particular cell subtypes in the original paper that published the dataset. FIMO calculates the number of times a motif appears in a set of sequences, with a p-value threshold of 1e-3. We extracted the motif of SOX10, Nr4a2, and JunB from the JASPAR 2020 database and Bcl11b from CIS-BP for FIMO to compare composition between generated data and real data [7].

The ability for GAN to reconstruct motifs is also tested using MEME SUITE STREME, which detects ungapped motifs that are enriched or relatively enriched compared to control sequences [3]. The control sequences in this case are generated by shuffling the original sequence preserving k-mer with default parameters. The motif discovered is then compared to all JASPAR 2020 motifs using MEME SUITE TOMTOM [13].

3.5 Latent Space Exploration

The latent space of the CCGAN can be explored by manipulating the latent vector z. All z vectors are generated from a standard Gaussian distribution unless specified. By varying z it is possible to map out the approximate sequence space learned by the GAN. By using latent space exploration it is possible to ascertain the approximate locations of TF binding motifs and to highlight cell-specific motif presence. We first tried to show the learned cell-specific signal by comparing the generated sequence to other generated sequences using the same z vector but different cell type specification. In this case, if CCGAN has learned cell-specific motif characteristics, then cell-specific generated sequences would contain unique patterns of motifs enrichment even if z is shared. However, if no unique motifs were found by varying cell conditional, it would indicate no cell-specific motif has been learned.

To ensure we capture differences in cell conditional specification, z vectors that are close together are used to decrease stochasticity in generating sequence. Thirty-two pairs of randomly generated z vectors were drawn from a normal distribution. Between each pair, 100 z vectors were generated equally spaced based on spherical linear (slerp) distance (6400 sequences in total). The motif composition was tested using MEME SUITE AME with JASPAR 2018 Vertebrate database for motif enrichment patterns [17]. A heatmap was generated using the motif composition value, as well as a Principal Component Analysis (PCA) analysis against the top-ranked sequences. The effect of variance in generating random sequences has also been tested in the latent space. A range of values for standard deviation was used to generate sequences and motif enrichment was tested via MEME SUITE AME [28].

Chapter 4

Result

4.1 Data processing reveals high noise level in single cell ATAC-seq

Published scATAC-seq data from Preissl et al. (2017) and Lareau et al. (2018) was prepared for the study. Both brain-related data sets contain a diverse range of cell types, which we took as assigned in the source papers, including the subclasses [23, 31]. All of the DNA signals were used for training regardless of quality. The top-ranked sequences (i.e. peak with highest presence in respective cell types) for each cell type were extracted as positive controls. Spurious signals were removed, excluding those that appear in less than 1% of counts in a type of cell. For the embryonic mice forebrain data from Preissl et al., (2017), filtering of the spurious peaks yielded on average 4573 unique peaks per cell category with wide variance (figure A.3a) [31]. The top-ranked regions based on counts extracted were observed to be present in a minimum of 20 cells for the respective cell types. Hierarchical clustering was performed for a random sample of peaks (figure A.2a), resulting in grouping of similar cell types. Embryonic inhibitory neuron subclasses 1, 2 and 4 clustered together, while subclass 3 was placed closer to retinal glial class 3. Excitatory cells are clustered together and Erythroid Myeloid Progenitors (EMP) are the furthest from other cell types.For the adult mice brain scATAC-seq data from Lareau et al., (2018), most cells were classified as excitatory neurons and were grouped together (figure A.2b), with the exception of EN17 which clustered

with inhibitory neurons. Endothelial (E1) was the furthest from the rest of the cell types. EN13 was clustered with other non-neuronal cells such as astrocyte (A1) and microglial cells (M1).

4.2 GAN training stabilized by wGAN-GP and label-smoothing

Identification of suitable stabilization techniques and parameters is an important step in model development. The training of GAN was convergent when label-smoothing was applied and the learning rate was slower (learning rate of ideal range between $1e^{-5}$ or $1e^{-6}$ compared to $1e^{-3}$). The magnitude of the loss of the GAN is the difference between the loss of generator and the discriminator, often viewed separately, which informs about the stability between the two player generator-discriminator balances. A stable relation allows the generator to learn to generate better data based on the feedback from the discriminator. Eventually the discriminator should be unable to distinguish between generated and real data Mode collapse, which is a GAN failure model in which replicates of data were generated, was observed early in the model. For successful models, the loss stabilizes after the first epoch and does not reach zero. Training of the CNN approach took less epochs, completing in around 3 epochs compared to at least 15 for the GAN approach. The AUC value for embryonic mice forebrain data is 0.64 (vs random value of 0.5) with EMP cell type, the class with the least data, the most difficult to learn (figure A.4). The average AUC score of all cell type classification was 0.73. For the adult mice brain data from Lareau et al., (2018), the average AUC scores were 0.70 (figure A.5).

4.3 Generated sequence tested negative for memorization of training data and diversity

Since GAN training is prone to failure such as mode collapse, in which highly similar samples are generated, sample diversity must be confirmed for the synthetic sequences produced. As well, such testing can reveal if the neural net has memorized the training sample, causing overfitting and inflated performance. The similarity between generated sequences and the similarity between generated se-

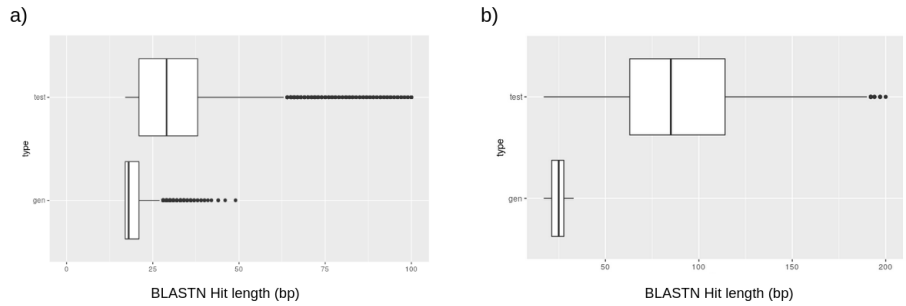

Figure 4.1: a) **BLASTN test for GAN mode collapse**: BLASTN hit result against partitioned test set and generated sequence to itself. Quality check for potential mode collapse which result in GAN producing the same sequences. E-value cutoff was set to be 1e-1 to include short hits of sequences; b) **BLASTN test for training set memorization**: BLASTN hit query length of test and generated sequences against training sequences. E-value cutoff was set to 1e-1 for including short hits.

quences and training sequences were assessed using BLASTN comparisons. The average query lengths (and the variance of the lengths observed) in BLASTN detected segments of similarity between generated sequences was smaller than the range of BLASTN query lengths between validation sequences (figure 4.1a). Since the alignment e-value threshold was set to be permissive, low quality long alignments were also included in the validation sequence BLASTN results. The average alignment length for generated sequences against themselves is 18 bp, whereas validation sequences against themselves were on average 32 bp. The maximum length of alignment is 49 bp for generated sequence comparisons compared to 500 bp for validation sequences against itself. When sequences were tested for memorization, the BLASTN alignment length on average was 24 bp for generated sequences against training sequences and 90 bp for validation sequences against training sequences (figure 4.1b).

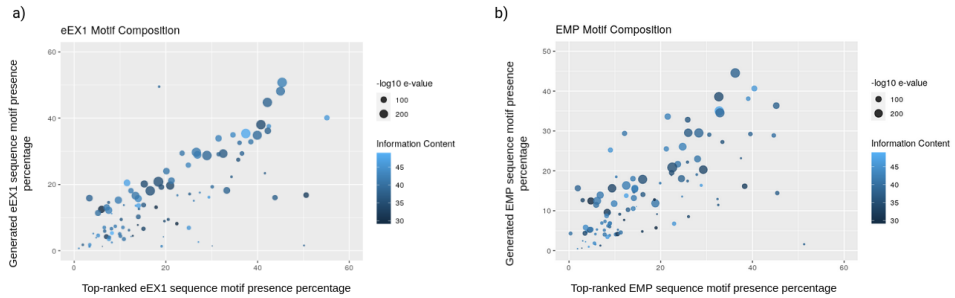

Figure 4.2: a) **Embryonic mouse forebrain excitatory type 1 motif composition using MEME SUITE AME**: Motif composition tested against the JASPAR 2018 vertebrate database of top-ranked sequences in excitatory type 1 and generated excitatory type 1 sequences via MEME SUITE AME. ; b) **EMP motif composition analysis**: Motif composition of EMP compared between top-ranked EMP sequences and generated sequences tested using MEME SUITE AME using JASPAR 2018 Vertebrate database.

4.4 Generated sequence motif composition and enrichment pattern match real top-ranked sequences

Amongst the most important aspects of generated sequences is the retention of biologically relevant characteristics. Cell-specific regulatory sequences are expected to contain functional features such as TF binding motifs. Enriched TF binding motifs were highly correlated between generated and validation sequences across cell types (ure 4.2). It is noted that due to the highly noisy property of single cell data, randomly selected sequences did not yield any positive result in motif enrichment when tested using MEME SUITE AME. Due to this reason, the positive control was instead composed of top-ranked sequences that appeared among the highest frequency in respective cell type. As the GAN generated sequences depend on a random vector z, the sequence motif composition in generated sequences may not represent the entire potential space of simulations. For instance, the TF binding motif for NEUROD1 was detected in 35.78 % of the generated sequences compared to 24.75% of the validation sequences for embryonic mice forebrain ATAC-seq eEX1 cell type. The performance of CCGAN on motif composition for generated sequences compared with the top ranked sequences had a high correlation r^2 of

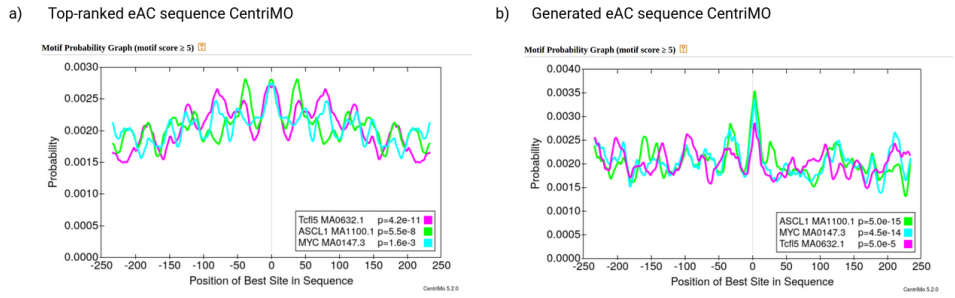

Figure 4.3: a) **CentriMO analysis of top-ranked (validation) eAC sequences**: MEME SUITE CentriMO enrichment pattern found for top ranked sequences in embryonic astrocyte for Tcl5, ASCL1 and MYC. Both forward and reverse complement sequences were included.; b) **CentriMO analysis of generated eAC sequences**:MEME SUITE CentriMO analysis for TF enrichment pattern of generated embryonic astrocyte sequences.

0.34 (figure 4.2b). As well, EMP-specific motifs such as MEF2C were found to be enriched in a similar proportion (29.28% and 20.30% in generated and top ranked sequences, respectively). CentriMo analysis, which assesses the positional distribution of TF motifs, displayed a symmetric wave pattern with its peak in the center of the 500 bp validation sequences (figure 4.3a) and generated sequences (figure 4.3b). The symmetric pattern for the real sequence (figure 4.3a) results from the inclusion of both forward and reverse sequences in the dataset. This pattern is observed in the generated sequence (figure 4.3b), with lower symmetricity. The enrichment of TF binding motifs emerged relatively early in the model development process (figure 4.4a). The motif enrichment often emerged before the 24th epoch in the model training process, but there was less noise in the exact composition as the model continued to train. After 53 epochs, the GAN model was observed to have less noise in its motif composition, and a lower e-value score for its motif enrichment, as shown in (figure 4.4a).

As the dscATAC-seq adult mice brain publication highlighted certain TFs enriched in particular groups, FIMO was used to compare the specific TF motif compositions in a sample of sequences between groups. For highlighted TF enrichment from Lareau et al., (2018), the enrichment level was compared between top-ranked

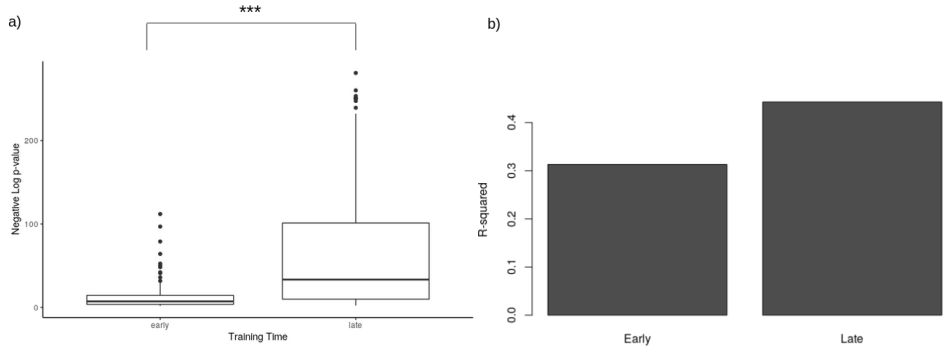

Figure 4.4: a) **Negative Log p-value for motif enrichment of early vs late CCGAN model conditioned on eEX1**: MEME SUITE AME was used to test for motif enrichment, the negative log p-value for late model was significantly higher ($p < 0.0001$) tested by unpaired Wilcox rank sum test.; b) r^2 **value of generated sequence motif composition correlation to top-ranked sequence motif composition of early and late model model conditioned on eEX1**:MEME SUITE AME analysis for TF enrichment percentage in sequence was calculating correlation.

sequence and generated sequence in table 4.1. FIMO tests for presence of a specified motif in a sequence, with a significance threshold of p$\leq$1e-4. The percentage of sequences that contained the specified motif is compared between generated and top-ranked sequences, which serve as a positive control. If the denoising process was successful, those highlighted motifs previously reported to be enriched in the cell type should have a similar or higher presence percentage in generated sequences. The presence percentage MG1 or microglial cells were observed to be highly enriched for the Bcl11b motif, with generated and top sequences showing 6.45% and 6.72% of presence percentage respectively. SOX10 was highly enriched in the OG1, oligodendrocytes had 6.72% and 6.45% of presence percentage in generated and real sequences, respectively. For the Nr4a2 motif which was observed to be highly enriched in EN13 and EN15, the presence percentage for the motif in generated sequences and real sequences are 11.7% and 9.50% for EN13, 11.8% and 8.78% for EN15, respectively. JunB motif was significantly present in 14.1% generated sequence and 8.89% in IN01. For all TF enriched in certain cell types reported by Lareau et al., (2018) generated sequences have similar or higher motif

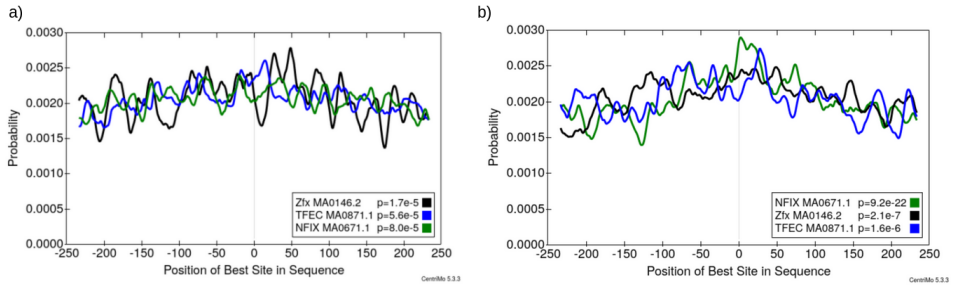

Figure 4.5: a) **CentriMO analysis of generated oligodendrocytes sequences**:MEME SUITE CentriMO analysis for TF enrichment pattern of generated adult oligodendrocytes sequences. ; b) **CentriMO analysis of top-ranked (validation) oligodendrocytes sequences**: MEME SUITE CentriMO enrichment pattern found for top ranked sequences in adult oligodendrocytes for Zfx, TFEC and NFIX. Only forward sequences included in positive control.

presence than top-ranked sequences [23].

Both results obtained using embryonic mice forebrain snATAC-seq and adult mice brain dscATAC-seq data showed a high correlation between generated sequences and top ranked sequences among the motif composition tested by AME, shown in figure 4.2 and figure A.6a. This means generated sequences contain the motifs enrichment as well as similar motif presence in top-ranked sequences. Similar peak patterns for the same TFs tested via MEME SUITE CentriMO were also observed, shown in figure 4.5. CentriMO is used to test local enrichment patterns along the sequence. It examines how closely generated sequences' local motif enrichment pattern matches real top-ranked sequences. In this case, the CCGAN has only been trained for 33 epochs.

Another important aspect of good generated sequences is its ability to reconstruct important motifs *de novo*. After extracting motifs using MEME STREME, certain motifs emerged that were highly relevant in neurons. As seen in figure 4.6, oligodendrocyte relevant motifs KLF9, ETV4, TCF7, MAFK have been discovered in oligodendrocyte specific sequences. TCF7 was described by Weng et al., (2017) to promote oligodendrocyte differentiation and remyelination in mice [33]. ETV4 was also known to regulate TFs important for gliogenesis, which is the

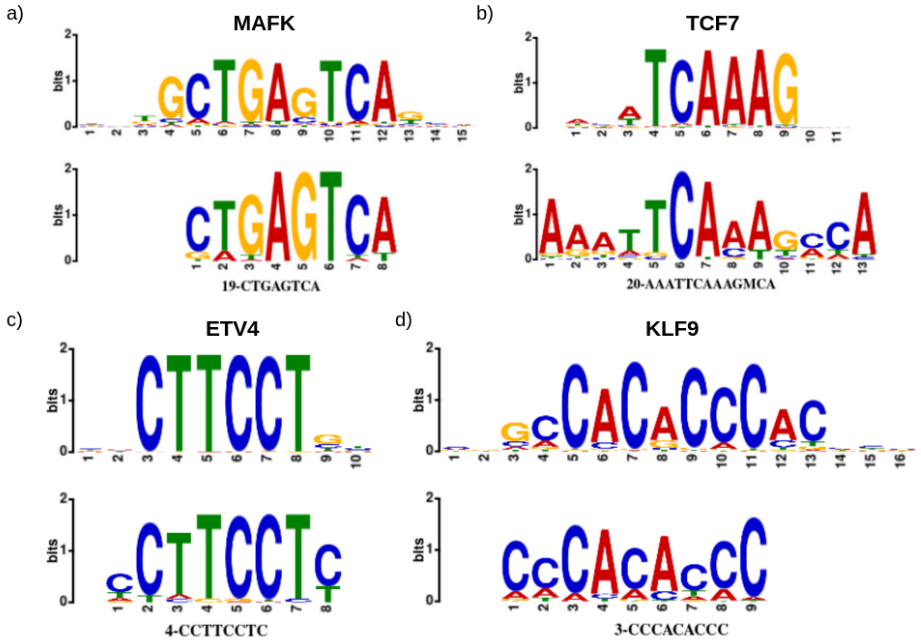

Figure 4.6: Discovered OG1 specific motifs from generated data ($p<0.05$, $q<0.05$) using MEME SUITE STREME and TOMTOM compared to consensus motifs: Specifically MAFK, TCF7, ETV4 and KLF9, all of which were described to be involved in oligodendrocytes.The generated motif is displayed at the bottom and the consensus motif is at the top.

developmental process for generating astrocytes and oligodendrocytes [25]. Therefore, generated sequences could be used to reconstruct motifs that has cell-specific significance.

4.5 Feedback network training was unnecessary for improving generated sequence quality

In previously described GAN models [12, 19, 26], an adjunct network has been employed for improving GAN performance. The adjunct network served as an external corrector to perfect sequences generated by the GAN. For ccGAN the adjunct network was not necessary. After two epochs of training, no noise reduction

Table 4.1: MEME SUITE FIMO analysis of highlighted motifs composition in generated vs top-ranked sequences: The percentage presence of the motif in 6400 sampled sequences is compared to top-ranked sequence motif presence.

	Top-ranked sequence	Generated sequence
Bcl11b (microglia)	6.72%	6.45%
SOX10 (oligodendrocyte)	6.45%	6.72%
Nr4a2 (excitatory neuron type 15)	9.50%	11.7%
Nr4a2 (excitatory neuron type 17)	8.78%	11.8%
JunB (inhibitory neuron type 1)	8.89%	14.1%

or more motifs emerged, which can be observed by comparing motif enrichment in figure A.6. The loss of CNN was expected to be positive for sequences deemed fake and negative for sequences deemed real. The inclusion of the feedback loop did decrease late stage CCGAN loss for CNN evaluation. After the feedback loop training, the CNN loss for generated sequences changed from unimodal to bimodal, shifting towards the real sequence distribution which is a normal distribution with mode around -1 (as shown in figure 4.7a). However, for the earlier version of the model which has been trained less, the loss was more optimal, as seen in figure 4.7c. For the early model, the CNN evaluation loss was more negative than real validation sequences, meaning the CNN think those sequences were better classified. Previous results suggested that more training of CCGAN resulted in a higher concentration of motifs and better motif composition correlation to top-ranked sequences. Therefore, loss evaluation given by the trained CNN may not be the most important aspect evaluating sequence quality.

4.6 Latent space exploration revealed simulated sequence properties similar to real top-ranked sequences

By comparing simulated sequences generated with the same z vector for different cell types, one can determine cell-specific properties that have been learned. Sequences generated using fixed z as illustrated by the heatmap in figure 4.8b, had more signals unique to cell types than sequences generated from different z in fig-

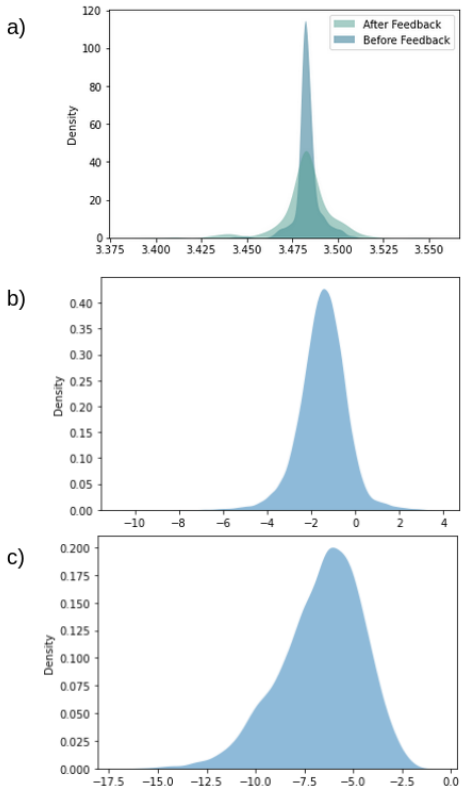

Figure 4.7: a) Generated sequence loss before and after feedback loop evaluated using the pre-trained adjunct CNN; b) Real snATAC-seq test set sequence CNN loss: Density plot of test snATAC-seq sequence loss evaluated using pre-trained adjunct CNN; c) Mid-stage training generated sequence CNN loss: Density plot of GAN generated sequence loss evaluated using pre-trained adjunct CNN.

ure 4.8a. TF binding motifs for developmental-relevant TFs such as FOXD3 in RG2 and PBX1 in EMP were found to be cell type specific. (Some inconsistencies were observed, such as the EMP-related CDX2 which was enriched in excitatory type 1 cell sequences.) Motif enrichment percentage tested using MEME SUITE AME and analyzed by PCA reveal similarity among cell types. Since the PCA axis tries to maximize the variance, the magnitude of distance between cell types could imply their similarity. Therefore, PCA is used to evaluate how well a randomly selected area in the latent space can differentiate between the cell types. For se-

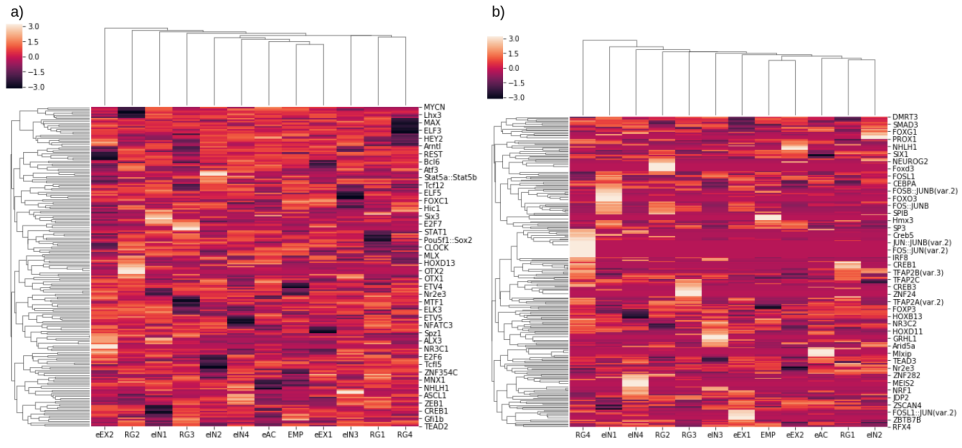

Figure 4.8: a)**Latent space cell-specific motif discovery using distinct z**: Heatmap of TF presence percentage compared across cell types for generated cell-specific mice embryonic forebrain AME motif enrichment analysis. The vector z is generated randomly from a gaussian distribution for each sequence and not shared between different cell conditionals; b)**Latent space cell-specific motif discovery using the same z with conditioned on different cell type.**: Heatmap of TF presence percentage compared across cell types for generated cell-specific mice embryonic forebrain AME motif enrichment analysis. The random vector z fixed and sequences are generated with different cell conditionals.

quences generated using fixed latent space vector z, retinal glial cells and inhibitory neurons were close to its generated counterpart on the PCA axis, as seen in figure 4.9. On the other hand, EMP and astrocytes were far apart from each other. This highlights the shortcomings of using fixed latent z which may amplify signals from space with lower quality sequences.

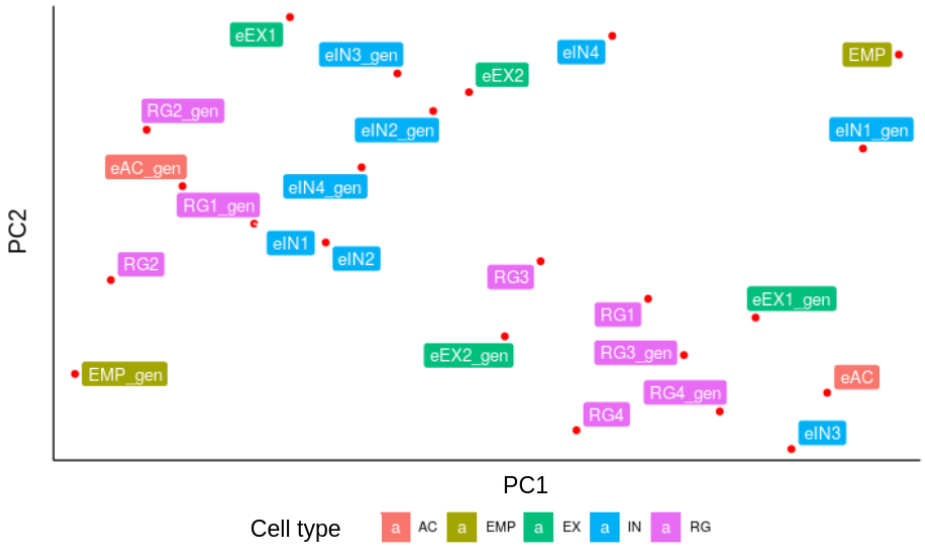

Figure 4.9: Fixed latent vector z with varying cell conditional motif enrichment PCA: Top-ranked cell-specific sequence motif composition is plotted with generated sequences conditioned on cell types. For those generated sequences, the z is fixed in between cell types. Analyzed using MEME SUITE AME.

Chapter 5

Discussion

Here we present a deep learning based model, CCGAN, that generates cell type-specific sequences with biologically relevant characteristics and minimal noise demonstrated across two single cell ATAC-seq (SCATAC-SEQ) data sets. Since SCATAC-SEQ reveals chromatin accessible regions, TF binding motif analysis of these regions can give insights into cell type-specific regulatory programs. Based on TF binding motif composition, enrichment percentage and pattern, the generated sequences appear similar to validation sequences from the experimental data sets. As opposed to previously published GAN-based simulators, the CCGAN model achieved its performance without inclusion of an independent CNN, demonstrating its capacity to learn the sequence distribution of different cell types independent of additional neural network components.

GAN-based simulators are drawing increasing focus due to their capacity to generate high quality, realistic data. However, the approach has drawbacks, including difficulty in selecting parameters and the duration of training. We attempted to use a structure that would be more forgiving of parameter choices by using wGAN-GP to decrease the possibility of mode collapse or diminishing gradient. The approach has proven robust, learning the properties of sequences (such as motif distribution) without memorization of training sequences nor generation of highly similar sequences. Using a slower learning rate helped to balance the competition between the generator and the discriminator and resulted in mode collapse. Analysis of the properties of models over the course of training revealed

that within relatively cycles of training (around 24 epochs), sequence characteristics have been learned. Longer training decreases the noise further, as seen for the GAN trained with embryonic mice forebrain data. In this case, the TF binding motif enrichment in the generated sequence became more pronounced with additional training, and the difference in motif composition decreased. Users can adjust the amount of training time, as some tasks can be accomplished more quickly (e.g. de-noising the simulated sequences).

The quality of prediction from generated sequences rests on the strength of the signals (both information content,frequency and uniqueness). The training result in terms of data required for prediction contrasted between a GAN implemented with an adjunct CNN and CCGAN without. In the embryonic mice brain dataset, EMP was (one of) the rarest cell types. With the adjunct CNN, the EMP AUC score was the lowest among the embryonic forebrain cell types (figure A.4). Despite having few signals, the regions identified as accessible chromatin in EMP cells differed from other cell types, and consequently the EMP sequences contained more unique peaks, as shown in unique peak counts in figure A.3. This uniqueness could also be inferred by the hierarchical clustering result of the binarized accessibility matrix (figure A.2). For discriminative models such as CNN, only a few amounts of data can be detrimental to the model's performance. Nevertheless, CCGAN, a generative model, was not affected by the lack of samples and EMP had one of the best motif composition correlations between top-ranked and generated sequences of 0.34. In addition to identifying unique TF motifs involved in mesoderm development. The CCGAN incorporated relevant signals despite only being given a non-stringent binary designation of peak importance. A possible contributor to the success may be the ability of ccGAN to learn the underlying distribution of motifs and how they differ between cell types. More data for a cell type is correlated with improved performance. For instance, eEX1 has the largest amount of peaks in total and one of the highest r^2 values (0.42) despite not having a high amount of unique peaks.

In previous sequence-generating GAN or other models, de-noising property has rarely been mentioned. Here we demonstrate the robustness of CCGAN, which can learn important data properties despite the presence of large amounts of noise. As well, previous sequence generating GANs specialize on one type of sequence such

as antimicrobial peptide, splice site without further class specification [12, 19]. With cell-specificity becoming increasingly important to decipher disease mechanisms, the ability to generate highly specified signals from rare cells can help us understand them better. For most previously described sequence-generating GANs, an adjunct model was required to evaluate the sequence [12, 19, 26]. Here we show CCGAN could generate sequence stand alone and adjunct models could have great limitations when data is sparse.

The ability of CCGAN to generate diverse cell type-specific sequences is unusual in the field. Cell-specific targeting of therapies is gaining increasing attention. New sequencing techniques, such as MIRACL-seq, focus on the regulatory signals of rare cell types [6]. Such sequences may be used in gene therapy to deliver expression in a selective manner. We foresee the utility of CCGAN to simulate such sequences as offering a new direction for such work. With an endless production of sequences, we can now explore the whole sequence space and test the candidate sequences to select optimal sequences for gene therapy delivery.

Bibliography

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, Oct. 1990. doi:10.1016/s0022-2836(05)80360-2. URL https://doi.org/10.1016/s0022-2836(05)80360-2. → page 11

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan, 2017. → pages 2, 9, 10

[3] T. L. Bailey. STREME: accurate and versatile sequence motif discovery. *Bioinformatics*, Mar. 2021. doi:10.1093/bioinformatics/btab203. URL https://doi.org/10.1093/bioinformatics/btab203. → page 12

[4] T. L. Bailey and P. Machanick. Inferring direct DNA binding from ChIP-seq. *Nucleic Acids Research*, 40(17):e128–e128, May 2012. doi:10.1093/nar/gks433. URL https://doi.org/10.1093/nar/gks433. → page 12

[5] C. Camacho, G. Coulouris, V. Avagyan, N. Ma, J. Papadopoulos, K. Bealer, and T. L. Madden. BLAST: architecture and applications. *BMC Bioinformatics*, 10(1):421, 2009. doi:10.1186/1471-2105-10-421. URL https://doi.org/10.1186/1471-2105-10-421. → page 11

[6] E. Drokhlyansky, C. S. Smillie, N. V. Wittenberghe, M. Ericsson, G. K. Griffin, G. Eraslan, D. Dionne, M. S. Cuoco, M. N. Goder-Reiser, T. Sharova, O. Kuksenko, A. J. Aguirre, G. M. Boland, D. Graham, O. Rozenblatt-Rosen, R. J. Xavier, and A. Regev. The human and mouse enteric nervous system at single-cell resolution. *Cell*, 182(6): 1606–1622.e23, Sept. 2020. doi:10.1016/j.cell.2020.08.003. URL https://doi.org/10.1016/j.cell.2020.08.003. → page 28

[7] O. Fornes, J. A. Castro-Mondragon, A. Khan, R. van der Lee, X. Zhang, P. A. Richmond, B. P. Modi, S. Correard, M. Gheorghe, D. Baranašić,

W. Santana-Garcia, G. Tan, J. Chèneby, B. Ballester, F. Parcy, A. Sandelin, B. Lenhard, W. W. Wasserman, and A. Mathelier. JASPAR 2020: update of the open-access database of transcription factor binding profiles. *Nucleic Acids Research*, Nov. 2019. doi:10.1093/nar/gkz1001. URL https://doi.org/10.1093/nar/gkz1001. → page 12

[8] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. → page 1

[9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014. → pages 1, 6, 9

[10] C. E. Grant, T. L. Bailey, and W. S. Noble. FIMO: scanning for occurrences of a given motif. *Bioinformatics*, 27(7):1017–1018, Feb. 2011. doi:10.1093/bioinformatics/btr064. URL https://doi.org/10.1093/bioinformatics/btr064. → page 12

[11] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans, 2017. → pages 3, 9

[12] A. Gupta and J. Zou. Feedback GAN for DNA optimizes protein functions. *Nature Machine Intelligence*, 1(2):105–111, Feb. 2019. doi:10.1038/s42256-019-0017-4. URL https://doi.org/10.1038/s42256-019-0017-4. → pages 6, 10, 21, 28

[13] S. Gupta, J. A. Stamatoyannopoulos, T. L. Bailey, and W. Noble. Quantifying similarity between motifs. *Genome Biology*, 8(2):R24, 2007. doi:10.1186/gb-2007-8-2-r24. URL https://doi.org/10.1186/gb-2007-8-2-r24. → page 12

[14] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax, 2016. → page 10

[15] A. Jindal, P. Gupta, Jayadeva, and D. Sengupta. Discovery of rare cells from voluminous single cell expression data. *Nature Communications*, 9(1), Nov. 2018. doi:10.1038/s41467-018-07234-6. URL https://doi.org/10.1038/s41467-018-07234-6. → page 4

[16] D. R. Kelley, J. Snoek, and J. L. Rinn. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Research*, 26(7):990–999, May 2016. doi:10.1101/gr.200535.115. URL https://doi.org/10.1101/gr.200535.115. → page 10

[17] A. Khan, O. Fornes, A. Stigliani, M. Gheorghe, J. A. Castro-Mondragon, R. van der Lee, A. Bessy, J. Chèneby, S. R. Kulkarni, G. Tan, D. Baranasic, D. J. Arenillas, A. Sandelin, K. Vandepoele, B. Lenhard, B. Ballester, W. W. Wasserman, F. Parcy, and A. Mathelier. JASPAR 2018: update of the open-access database of transcription factor binding profiles and its web framework. *Nucleic Acids Research*, 46(D1):D260–D266, Nov. 2017. doi:10.1093/nar/gkx1126. URL https://doi.org/10.1093/nar/gkx1126. → page 13

[18] A. Khan, R. R. Puig, P. Boddie, and A. Mathelier. BiasAway: command-line and web server to generate nucleotide composition-matched DNA background sequences. *Bioinformatics*, Nov. 2020. doi:10.1093/bioinformatics/btaa928. URL https://doi.org/10.1093/bioinformatics/btaa928. → page 10

[19] N. Killoran, L. J. Lee, A. Delong, D. Duvenaud, and B. J. Frey. Generating and designing dna with deep generative models, 2017. → pages 6, 21, 28

[20] D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2013. → page 6

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf. → page 10

[22] A. Lal, Z. D. Chiang, N. Yakovenko, F. M. Duarte, J. Israeli, and J. D. Buenrostro. Deep learning-based enhancement of epigenomics data with atacworks. *Nature Communications*, 12(1), Mar. 2021. doi:10.1038/s41467-021-21765-5. URL https://doi.org/10.1038/s41467-021-21765-5. → page 7

[23] C. A. Lareau, F. M. Duarte, J. G. Chew, V. K. Kartha, Z. D. Burkett, A. S. Kohlway, D. Pokholok, M. J. Aryee, F. J. Steemers, R. Lebofsky, and J. D. Buenrostro. Droplet-based combinatorial indexing for massive-scale single-cell chromatin accessibility. *Nature Biotechnology*, 37(8):916–924, June 2019. doi:10.1038/s41587-019-0147-6. URL https://doi.org/10.1038/s41587-019-0147-6. → pages 7, 8, 14, 20

[24] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In M. F.

Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1558–1566, New York, New York, USA, 20–22 Jun 2016. PMLR. URL http://proceedings.mlr.press/v48/larsen16.html. → page 6

[25] X. Li, J. M. Newbern, Y. Wu, M. Morgan-Smith, J. Zhong, J. Charron, and W. D. Snider. MEK is a key regulator of gliogenesis in the developing brain. *Neuron*, 75(6):1035–1050, Sept. 2012. doi:10.1016/j.neuron.2012.08.031. URL https://doi.org/10.1016/j.neuron.2012.08.031. → page 21

[26] J. Linder, N. Bogard, A. B. Rosenberg, and G. Seelig. A generative neural network for maximizing fitness and diversity of synthetic DNA and protein sequences. *Cell Systems*, 11(1):49–62.e16, July 2020. doi:10.1016/j.cels.2020.05.007. URL https://doi.org/10.1016/j.cels.2020.05.007. → pages 6, 21, 28

[27] Q. Liu, S. Chen, R. Jiang, and W. H. Wong. Simultaneous deep generative modeling and clustering of single cell genomic data. Aug. 2020. doi:10.1101/2020.08.17.254730. URL https://doi.org/10.1101/2020.08.17.254730. → page 7

[28] R. C. McLeay and T. L. Bailey. Motif enrichment analysis: a unified framework and an evaluation on ChIP data. *BMC Bioinformatics*, 11(1), Apr. 2010. doi:10.1186/1471-2105-11-165. URL https://doi.org/10.1186/1471-2105-11-165. → pages 11, 13

[29] M. Mirza and S. Osindero. Conditional generative adversarial nets, 2014. → page 6

[30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf. → page 10

[31] S. Preissl, R. Fang, H. Huang, Y. Zhao, R. Raviram, D. U. Gorkin, Y. Zhang, B. C. Sos, V. Afzal, D. E. Dickel, S. Kuan, A. Visel, L. A. Pennacchio,

K. Zhang, and B. Ren. Single-nucleus analysis of accessible chromatin in developing mouse forebrain reveals cell-type-specific transcriptional regulation. *Nature Neuroscience*, 21(3):432–439, Feb. 2018. doi:10.1038/s41593-018-0079-3. URL https://doi.org/10.1038/s41593-018-0079-3. → pages 7, 8, 14

[32] V. Sandfort, K. Yan, P. J. Pickhardt, and R. M. Summers. Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks. *Scientific Reports*, 9(1), Nov. 2019. doi:10.1038/s41598-019-52737-x. URL https://doi.org/10.1038/s41598-019-52737-x. → page 6

[33] C. Weng, M. Ding, S. Fan, Q. Cao, and Z. Lu. Transcription factor 7 like 2 promotes oligodendrocyte differentiation and remyelination. *Molecular Medicine Reports*, 16(2):1864–1870, Feb. 2017. doi:10.3892/mmr.2017.6843. URL https://doi.org/10.3892/mmr.2017.6843. → page 20

[34] L. Xiong, K. Xu, K. Tian, Y. Shao, L. Tang, G. Gao, M. Zhang, T. Jiang, and Q. C. Zhang. SCALE method for single-cell ATAC-seq analysis via latent feature extraction. *Nature Communications*, 10(1), Oct. 2019. doi:10.1038/s41467-019-12630-7. URL https://doi.org/10.1038/s41467-019-12630-7. → page 7

[35] F. Yan, D. R. Powell, D. J. Curtis, and N. C. Wong. From reads to insight: a hitchhiker's guide to ATAC-seq data analysis. *Genome Biology*, 21(1), Feb. 2020. doi:10.1186/s13059-020-1929-3. URL https://doi.org/10.1186/s13059-020-1929-3. → page 4

Appendix A

Supporting Materials

A.1 Supplemental Figures

A.1.1 Detailed Architecture for GAN

The hyper-parameters for ccGAN are as follows:

Sequence length: 500

Matrix dimension (nucleotides): 4

Number of cell types: 12 for embryonic mice brain snATAC-seq, 27 for adult mice brain dscATAC-seq

Batch size: 16

Hidden units: 512

Learning rate: 1e-4

Discriminator steps per generator step: 5

Lambda: 10

Probability of label flip: 0.05

z dimension: 128

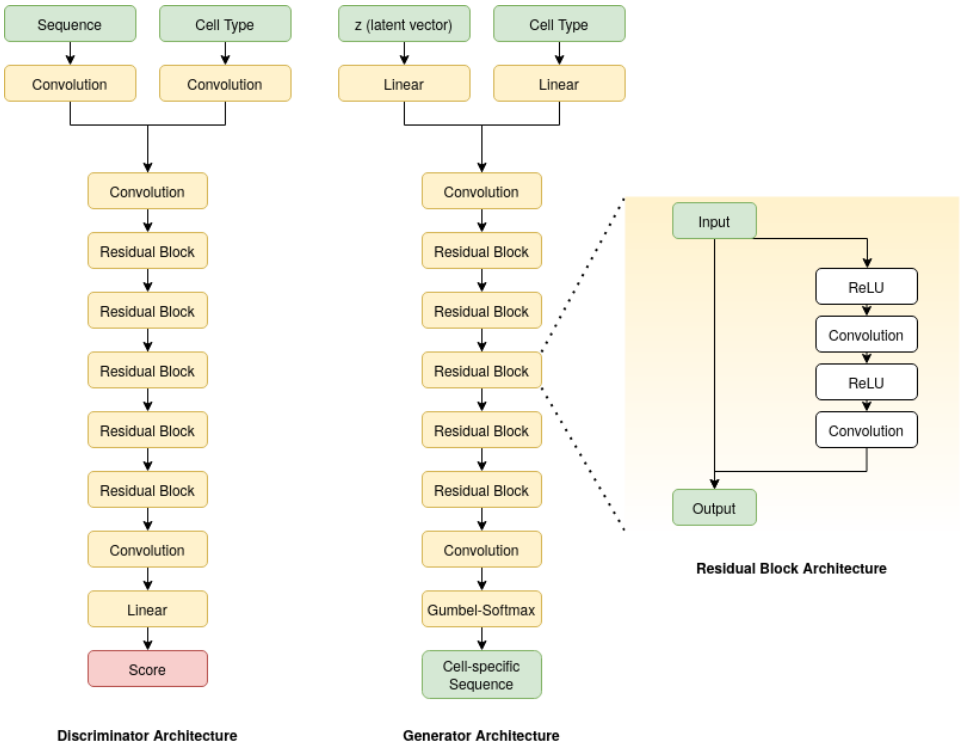

Figure A.1: ccGAN detailed architecture: Both the discriminator and the generator is consisted of residual blocks, which allows gradient to flow through without information loss. The cell conditional is incorporated into the input after the first layer. The architecture of residual block consisted of ReLU, convolution, ReLU and convolution, respectively.

A.1.2 snATAC-seq data processing

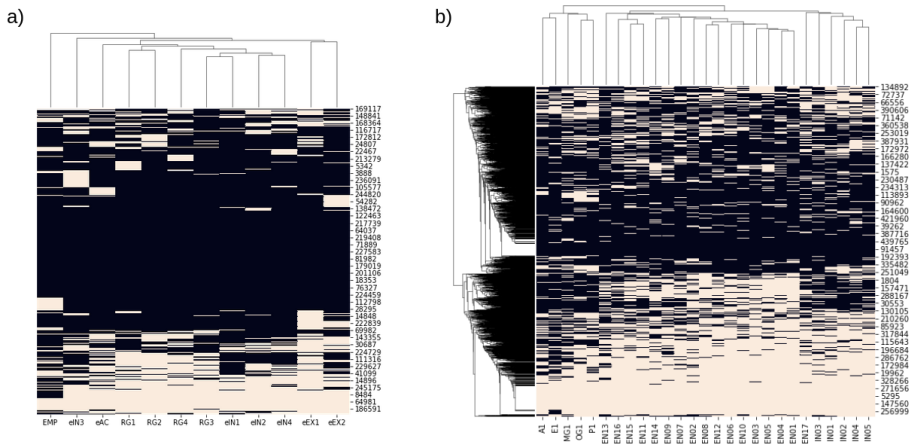

Figure A.2: a) **Embryonic mice forebrain data post-process clustering**: Presence-absence matrix of embryonic mouse forebrain snATAC-seq from Preissl et al., (2018). Peaks present in less than 1% of a cell type were marked as 0. 5000 peaks were sampled for assessing peak correlation between cell types using hierarchical clustering. Multiple subclasses of excitatory, inhibitory neurons and retinal glial cell classes are more closely grouped, while inhibitory type 3 (eIN3), and erythroid myeloid progenitors (EMP) or were more isolated; b) **Adult Mice brain dscATAC-seq data post-process clustering presence absence matrix**: Five thousand peaks were sampled randomly after binarization. Hierarchical clustering is used to show association between cell types. Excitatory and inhibitory neurons mainly clustered together, and glia cells clustered together.

A.1.3 Adjunct CNN Training

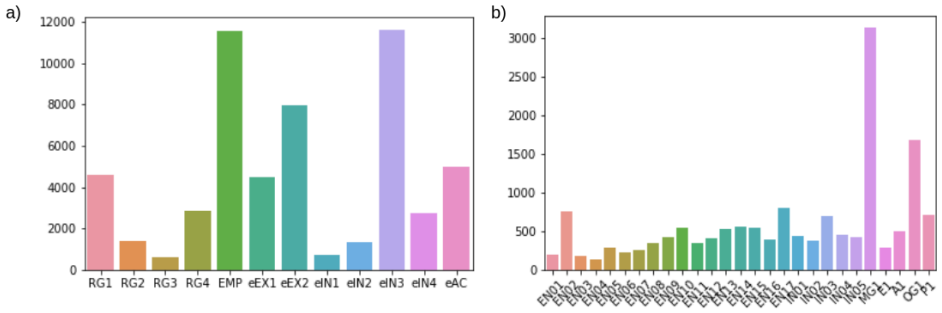

Figure A.3: a) **Mice Forebrain snATAC-seq unique peaks post-processing**: Unique peaks present for each cell type after initial filtering of peaks less than 1% of cells. EMP, eEX2 and eIN3 have high amounts of unique signals. RG3, eIN1 and eIN2 have comparatively few unique signals. b) **Mice brain dscATAC-seq unique peaks post-processing**: Unique peaks present for each cell type after filtering bottom 50% peaks ranked by frequency of cell with the region open. MG1, OG1 have high amounts of unique peaks while other cells have less.

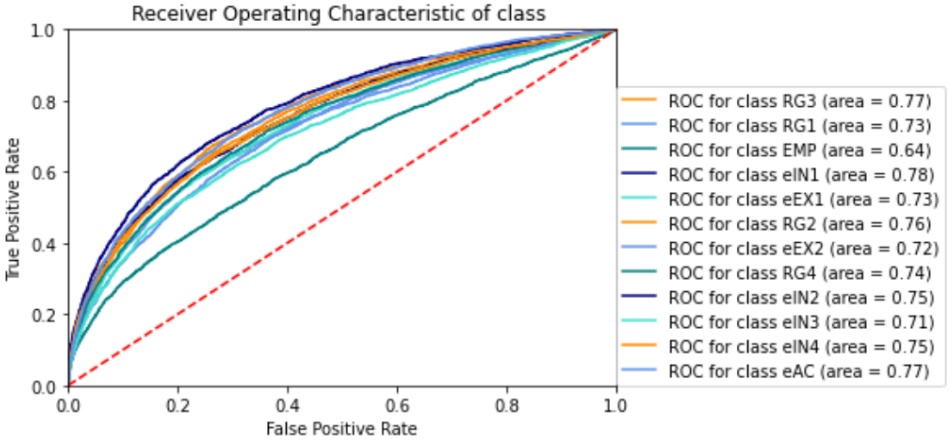

Figure A.4: Embryonic mice brain adjunct CNN ROC curve: Test set ROC curve with AUC value for each cell type from embryonic mice forebrain snATAC-seq. The accuracy of the CNN was the highest for inhibitory 1 and lowest for erythroid myeloid progenitor class.

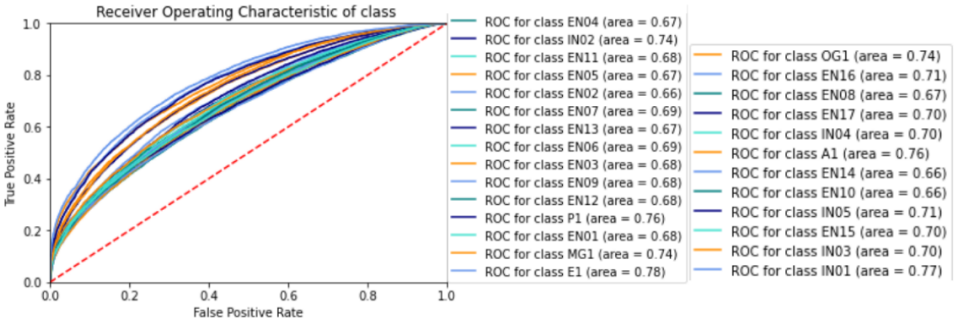

Figure A.5: Adult mice brain dscATAC-seq CNN ROC curve: Test set ROC curve for each cell type in the adult mice brain dscATAC-seq trained CNN.

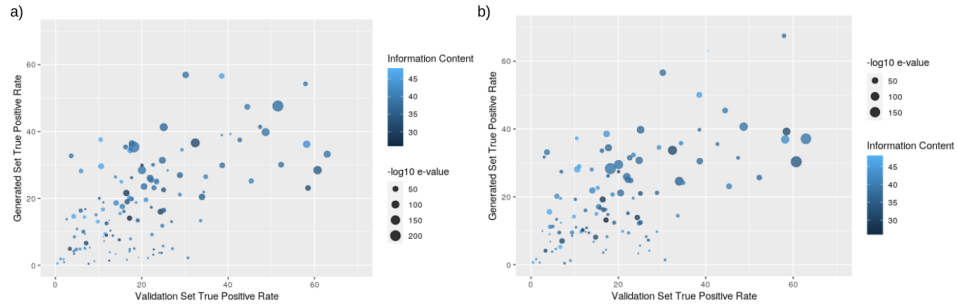

Figure A.6: a) **Adult mice brain dscATAC-seq astrocyte motif composition prior to the feedback loop**: Motif composition analysis of adult mice brain dscATAC-seq astrocyte compared between top-ranked astrocyte sequences and generated sequence prior to feedback loop. The analysis was done using MEME SUITE AME, using the JASPAR 2018 vertebrate database.; b)**Adult mice brain dscATAC-seq astrocyte motif composition after training GAN with the feedback loop**: Motif composition analysis of dscATAC-seq of astrocyte top ranked sequence compared to generated data set after training GAN using feedback loop. The analysis was conducted via MEME SUITE AME using the JASPAR 2018 vertebrate database.