An overview of deep learning methods for genomics

Matthew Ploenzke

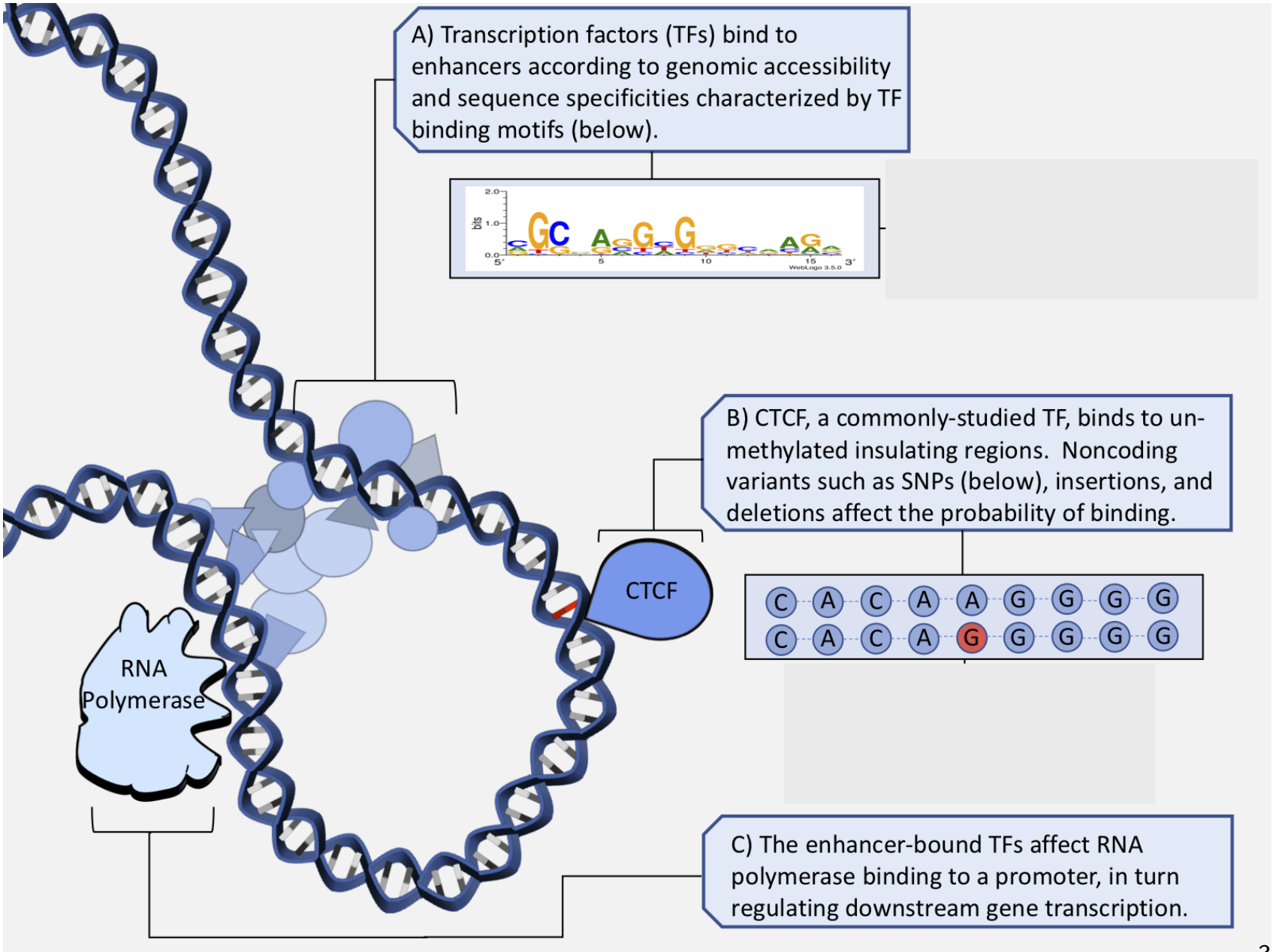
BST261
Harvard University

May 7, 2018

Snapshot

1. Case study #1: Discriminative learning
   - *Overview of ChIP-seq*
   - *Model formulation*
   - *Architecture variants*
   - *Model interpretations*
   - *Example*
2. Case study #2: Generative learning
   - *Review of GANs and VAEs*
   - *Overview of scRNA-seq*
   - *Model interpretations*

Problem Formulation: The Epigenome



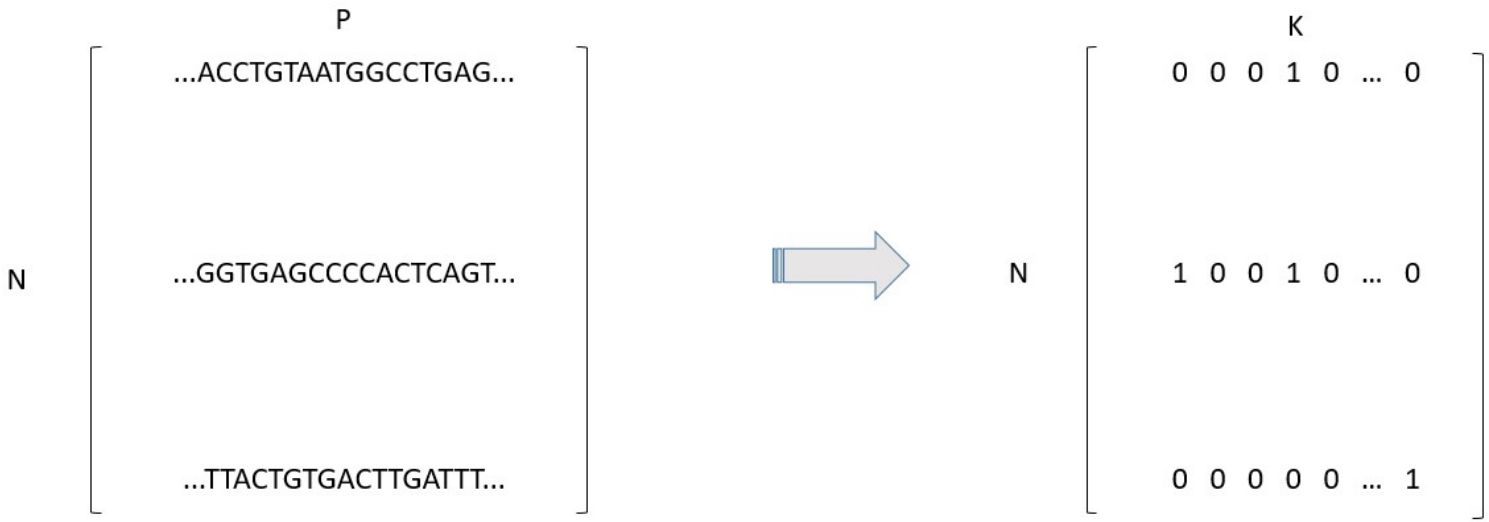
Problem Formulation: ChIP-seq

- Chromatin Immunoprecipitation Sequencing (ChIP-seq) technology used to crosslink and sequence segments of the genome bound by protein/transcription factor (TF)
- Output from an experiment is a long list of nucleotide sequences
- Computational tools (e.g. MACS[1]) may then be used to call peaks (i.e. predicted binding sites defined by high count locations)

How do these regions vary between cell type? Condition?

[1] Zhang, Yong, et al. "Model-based analysis of ChIP-Seq (MACS)." Genome biology 9.9 (2008): R137.

Problem Formulation: The Model



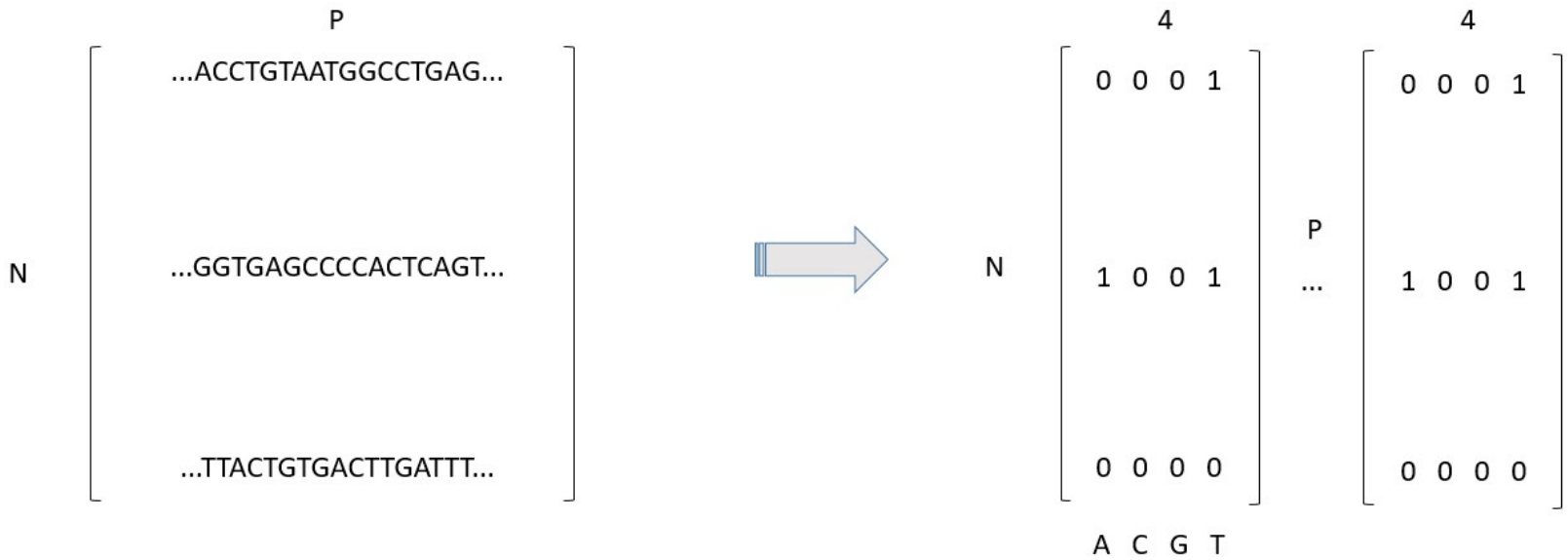
X

(nucleotide sequence peaks)

Y

(cell type/assay)

Problem Formulation: The Model



X

(nucleotide sequence peaks)

S

(one hot encoded)

Problem Formulation: Notation

Consider observations $X_i \in [A, C, G, T]^P$ for $i = 1, \ldots, N$ with corresponding labels $Y_i \in \{0, 1\}^K$. Taken together, X is a $N \times P$ matrix and Y is a $N \times K$ matrix.

We wish to learn a function $\mathcal{G}(\cdot)$ mapping $X \rightarrow Y$ through empirical risk minimization. For example, the logistic loss function under $K = 1$:

$$\min_{\omega}(R_{\mathcal{G}}) = \min_{\omega}\left(-Y\left[\log\left(\frac{1}{1+e^{-\mathcal{G}_\omega(X)}}\right)\right] - (1-Y)\left[\log\left(\frac{e^{-\mathcal{G}_\omega(X)}}{1+e^{-\mathcal{G}_\omega(X)}}\right)\right]\right)$$

As long as $R_{\mathcal{G}}$ is differentiable we may use the chain rule (*backpropagation*) to calculate the derivative and perform gradient descent to update parameter values ω and in turn minimize empirical risk. Penalties (e.g. L1) may simply be tacked on the loss.

Problem Formulation: Notation

We consider $\mathcal{G}(\cdot)$ to be composed of M compositional functions (*layers*) such that:

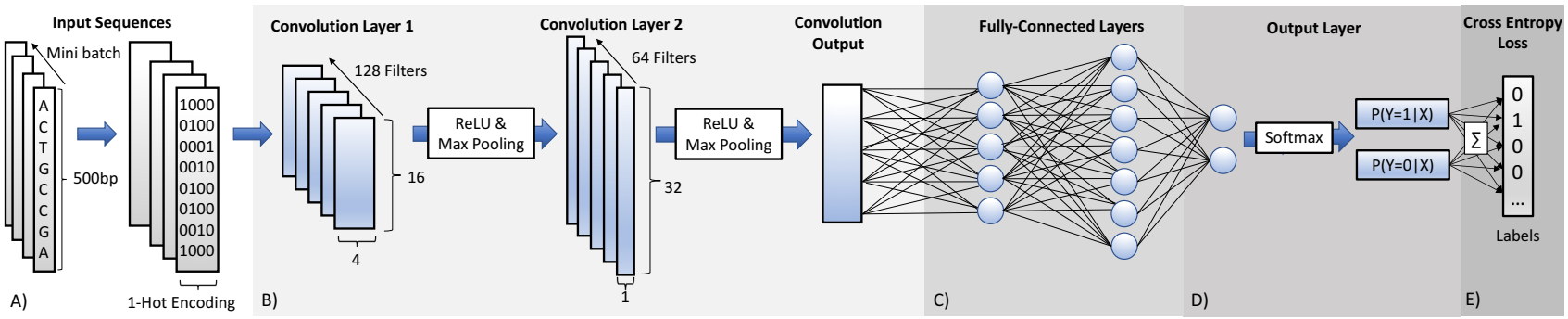
$$\mathcal{G}(S_i) = g_M(g_{M-1}(\ldots(g_2(g_1(S_i)))))$$

and require $g_1(S_i)$ to be a convolutional layer such that for convolutional filter f of length l_f at sequence position index J:

$$g_1^f(S_{i,J}) = \sum_{j=J}^{J+l_f} \sum_{n \in \{A,C,G,T\}} \omega_{1,j,n}^f \mathbb{1}_{S_{i,j}=n}$$

Problem Formulation: Visualization

For example, our CNN may look like this:



We then train our model for several epochs and obtain model weights from the iteration with lowest test set accuracy.

Model Architectures in Genomics

- DeepSEA[2]
- DeepBind [3]
- Basset [4]

[2]Zhou, Jian, and Olga G. Troyanskaya. "Predicting effects of noncoding variants with deep learningbased sequence model." Nature methods 12.10 (2015): 931.

[3]Alipanahi, Babak, et al. "Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning." Nature biotechnology 33.8 (2015): 831.

[4]Kelley, David R., Jasper Snoek, and John L. Rinn. "Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks." Genome research 26.7 (2016): 990-999.

Model Architectures in Genomics

- DanQ [5]
- Separable fully connected layers [6]
- Reverse complement parameter sharing [7]

[5] Quang, Daniel, and Xiaohui Xie. "DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences." Nucleic acids research 44.11 (2016): e107-e107.

[6] Alexandari, A. M., Shrikumar, A., & Kundaje, A. (2017). Separable Fully Connected Layers Improve Deep Learning Models For Genomics. BioRxiv, 146431. https://doi.org/10.1101/146431

[7] Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Reverse-complement parameter sharing improves deep learning models for genomics. BioRxiv, 103663. https://doi.org/10.1101/103663

Model Interpretation: Overview

Understanding model rationale is an active field of research.

> What has my black box learned?

A first and easy distinction to make is between:

1. encouraging intepretable learning **while training a model**
   - L1/L2 regularization, weight constraints, etc.
2. interpreting learned knowledge **with a trained model**

Model Interpretation: Importance Scores

Given a trained model, model intepretation may be performed by computing **importance scores**.

> How important is nucleotide *n* in contributing to the final model prediction?

There are two methodological approaches for computing such scores, or rather, *visualizing learning*:

1. Forward- or *perturbation*-based
2. Backward- or *backpropagation/gradient*-based[8]

[8] Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning Important Features Through Propagating Activation Differences. ArXiv:1704.02685 [Cs]. Retrieved from http://arxiv.org/abs/1704.02685

Model Interpretation: Forward-based

Forward-based approaches are quite simple:

1. For a given observation, obtain a predicted value
2. Modify the value of a single feature (e.g. nucleotide A $\rightarrow$ C)
3. Obtain a new prediction
4. Calculate the difference, either at the network level or node level

Model Interpretation: Learning Motifs

Alternatively, given the trained network, which observations maximize network activations (either individual network nodes or final network output)?

What sequence(s) has the network learned to recognize?

1. Pass test observations through the first convolutional layer
2. Per filter, zero out low values below threshold (noise)
3. Extract motif-length sequences around non-zero activations
4. Use sequences to compute position-weight matrix (PWM)

Model Interpretation: Learning Motifs

For motif m of length l, define the **information** (height) at position $j \in \{1, l\}$ as:

$$R_j = \log_2 4 - H_j.$$

with $H_j. = \sum_{n \in \{A,C,G,T\}} H_{j,n}$ defined as the total **entropy** at position j over nucleotides $n \in \{A, C, G, T\}$.

Write the entropy at position j for nucleotide n as:

$$H_{j,n} = \bar{f}_{j,n} \log_2 \bar{f}_{j,n}$$

for **relative frequency**, $\bar{f}_{j,n}$, of nucleotide n at position j. $\bar{f}_{j,n}$ is calculated from the sequences surrounding the non-zero activations.

Example: Learning Motifs

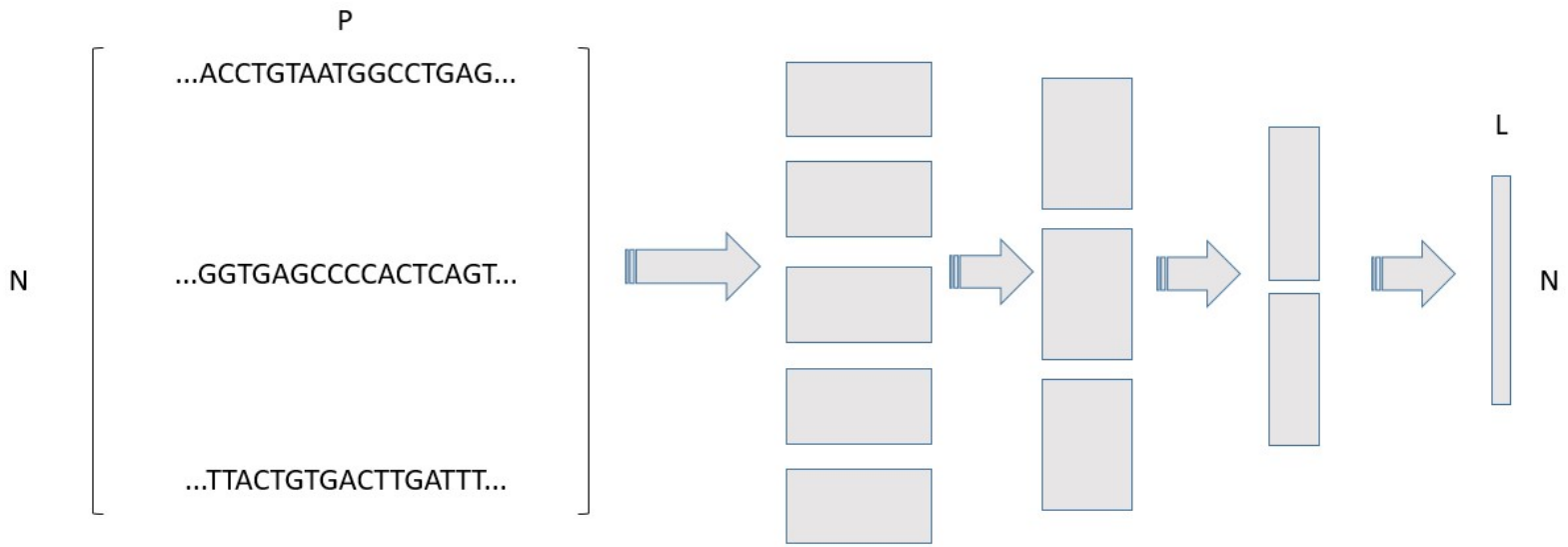
Consider a set of genomic reads S in which half of the set contains some motif ($Y = 1$) and half of the set does not contain the given motif ($Y = 0$).

The motif in our example will be a very simple string of seven C nucleotides.

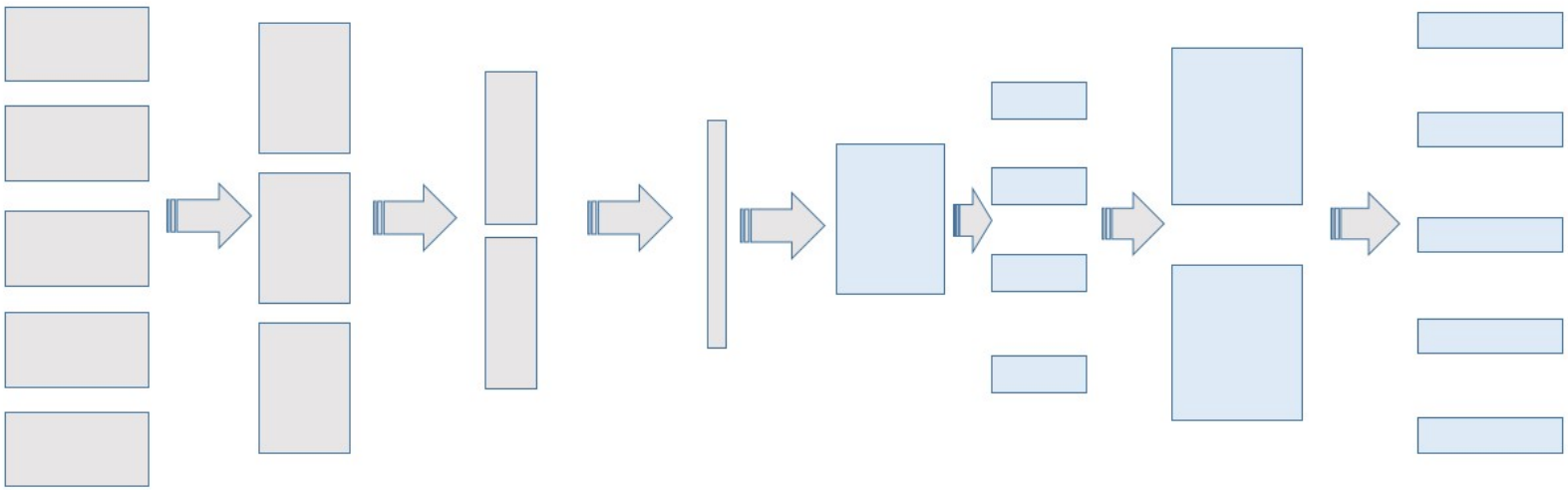
The goal is to train a binary classifier on the genomic sequences and understand to what extent the classifier has learned the inserted motif.

Data such as these could come from peak sequences called from a ChIP-seq experiment.

Problem Formulation: Generative Learning

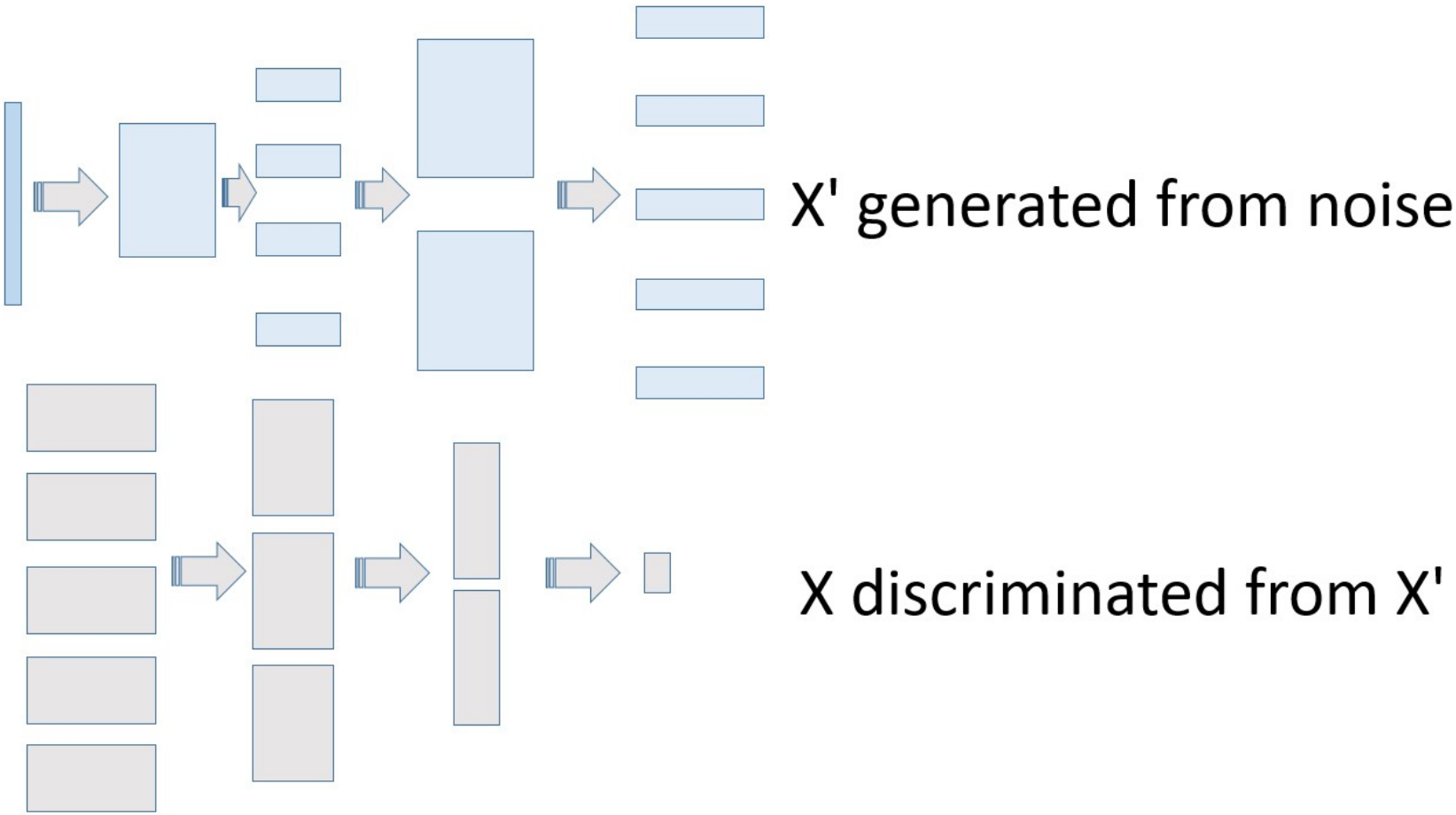


Problem Formulation: Autoencoders



X encoded to L ⇨ L decoded to X'

Problem Formulation: Adversarial Learning [9]



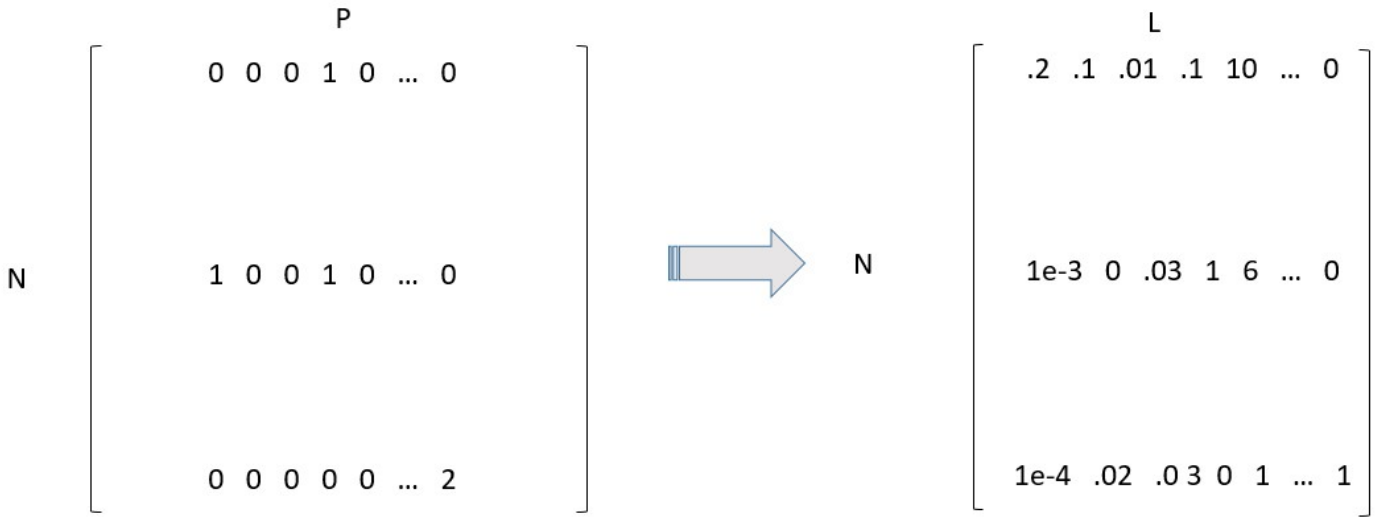
[9] Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.

Problem Formulation: single-cell RNA-seq

- In traditional RNA sequencing, one typically measures differential bulk gene expression across condition
  - *Gene X is active in cancer types but inactive in healthy types.*
  - It provides a picture of the average transcriptional activity in a tissue
- scRNA-seq, on the other hand, measures individual cells
  - It offers a snapshot of the transcripts at the *single cell* level
  - *Cell type A differentiates from cell type B due to gene X.*

Which cells belong to which cell type? Can we model these cell types with latent classes?

Problem Formulation: The Model



X

(cell-gene counts)

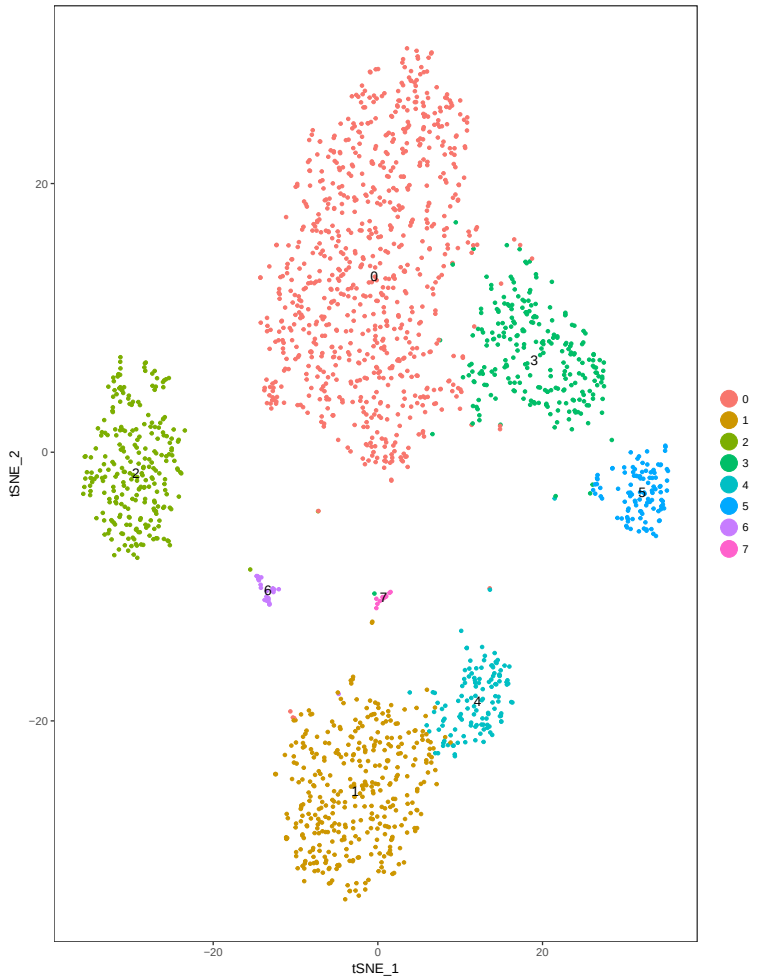
L

(latent states)

10

[10]Lopez, Romain, et al. "A deep generative model for gene expression profiles from single-cell RNA sequencing." arXiv preprint arXiv:1709.02082 (2017).

Model Interpretation: t-SNE clustering



11

[11]tSNE plot created with Seurat: Butler, Andrew, et al. "Integrating single-cell transcriptomic data across different conditions, technologies, and species." Nature biotechnology (2018).

Summary

- Deep learning for genomics is gaining increasing attention as the models continue to show improvements in predictive accuracy over past techniques
- Unfortunately model intepretation is still difficult and hinders applicability and adoption
- We saw how CNNs can learn the TF-binding profiles of cells
- Also discussed how deep generative modeling may be used to learn latent cell types
- Please remember this was just an overview and there is much that I glossed over or left out entirely
- If questions, you may email me at ploenzke@g.harvard.edu.