1 Appendix

1.1 Background: ADAM optimizer

ADAM[7] computes adaptive learning rates η for all parameters from estimates of first and second moments of the gradients. The first moment($\hat{m}_t$) involves the exponentially decaying average of the previous gradients and the second moment($\hat{v}_t$) involves exponentially decaying average of the previous squared gradients. The update rule for epoch t during training is:

$$\Theta_t \leftarrow \Theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t \tag{1}$$

Here, ϵ is a very small number to prevent division by zero.

1.2 Background: Siamese Network in Deep Learning

The Siamese architecture has been used in many real applications, like face recognition [8] and dimension reduction [4]. A Siamese network contains two copies of a deep neural network(DNN) sharing the same weights(W). Figure 1 shows a general schema of a siamese architecture. Inputs are pairs of samples X^A and X^B. The two twin networks are tied by a distance measure($D_W(X^A, X^B)$) computed at the output representations of the two twin networks. A meaningful mapping maps similar input vectors to nearby points on the output manifold and dissimilar vectors to distant points. Inputs are pairs of samples. By forwarding a pair of similar samples into the Siamese network and penalizing the outputs (distance) of the pair, we can intuitively limit the distance between two similar samples in the learned embedding space to be small.

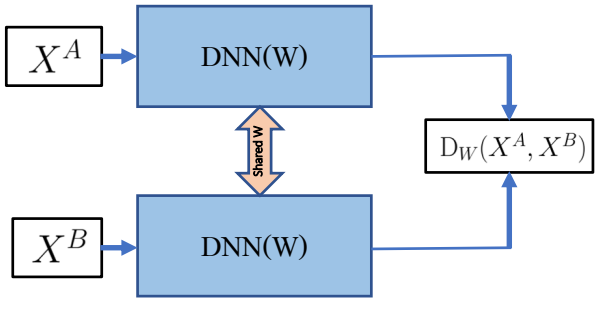

Fig. 1. Schematic of a general Siamese Network. Inputs are pairs of samples. By forwarding a pair into the Siamese network and penalizing the outputs of the pair, this training intuitively limits the D_W distance between two similar samples to be small. Backpropagation is used to train the network.

1.3 DeepDiff Variations Tried in our Experiments

We focus on the predictive modeling of *differential* gene expression given the histone modification profiles of a gene in two cell-types. To improve the prediction of differential gene expression, we use two types of auxiliary information. We use the cell-type specific expression prediction as an auxiliary task to the main task of differential gene expression prediction. Additionally, we also introduce a contrastive loss term as an auxiliary regularization term to further aid differential gene expression prediction. Combining these different auxiliary terms helps the model build powerful representations to improve differential gene expression prediction performance. Figure 3 in Section 3.6 presents an overview of our strategy. We use a number of variations of the DeepDiff model:

Raw Difference Features (Raw:d) First, we predict differential gene expression using the difference of the corresponding HM signals $\boldsymbol{X} = \boldsymbol{X}^A - \boldsymbol{X}^B$. We use $\boldsymbol{X}$ directly as input to the previously described Level I Embedding module f_1. The outputs from the Level I Embedding module are used as input to the Level II Embedding module f_2. This embedding $\boldsymbol{v}$ is passed through a linear layer for prediction. Intuitively, this is exactly like the AttentiveChrome model with input $\boldsymbol{X} = \boldsymbol{X}^A - \boldsymbol{X}^B$(shown in Figure 2 in Section 3.4).

Concatenation of Raw HM features (Raw:c) In this model, we treat the HM level signals a gene from the two cell-types as different HM features. We concatenate the HM profiles from the two cell-types into a single matrix $\boldsymbol{X} = [\boldsymbol{X}^A, \boldsymbol{X}^B]$ of size $(2 \times M) \times T$. This is used as input to the Level I Embedding module f_1 followed by the Level II Embedding module f_2. We use a Level I Embedding module that has one LSTM for each HM (from both cell-types), bin level attention weights $\alpha_{jt},\ j \in [1 \ldots 2 \times M]$ and $t \in [1 \ldots T]$ followed by a Level II Embedding module with HM level attention weights $\beta_j\ j \in [1 \ldots 2 \times M]$. Similar to *Raw:d* model, we only predict differential expression.

Concatenation and difference of raw HM features(Raw): In addition to the concatenated HM features, this variation uses an additional set of features corresponding to the difference of the HM profiles: $\boldsymbol{X}^A - \boldsymbol{X}^B$. Thus, the input matrix is now $\boldsymbol{X} = [\boldsymbol{X}^A, \boldsymbol{X}^B, \boldsymbol{X}^A - \boldsymbol{X}^B]$,a $(3 \times M) \times T$ matrix. We use this matrix as the input to the Level I Embedding module f_1 followed by the Level II Embedding module f_2. This Level II Embedding $\boldsymbol{v}$ is passed through a linear layer for prediction.

Adding Features from Auxiliary Tasks and Auxiliary Contrastive Loss: We propose using individual gene expression prediction as an auxiliary task to help the harder task of differential gene expression prediction. For this purpose, we propose the following variations:

Concatenation and Difference of HMs + Auxiliary features(Raw+Aux): This model aims at better feature representations for the *Raw* model. For this purpose, we use $\boldsymbol{X} = [\boldsymbol{X}^A, \boldsymbol{X}^B, \boldsymbol{X}^A - \boldsymbol{X}^B]$ as the input to a Level I Embedding module f_1^d, followed by a Level II Embedding module f_2^d for the DeepDiff main task. We add cell-type specific gene expression prediction for cell-type A and B as the Cell-Specific Auxiliary task(Auxiliary-Task-A and Auxiliary-Task-B, respectively). For this purpose, another Level I Embedding module f_1^A takes as input matrix $\boldsymbol{X}^A$ corresponding to the HM profile for cell-type A. This is followed by a Level II Embedding module f_2^A for cell-type A. Similarly, we use another Level I Embedding module f_1^B followed by the Level II Embedding module f_2^B for cell-type B. To leverage the information from the cell-type specific expression prediction tasks, additional auxiliary features are provided to the Level II Embedding module f_2^d. Concretely, in addition to the outputs of f_1^d, the Level II Embedding module f_2^d also takes as features outputs from the Level I Embedding modules f_1^A and f_1^B. Thus, f_2^d receives as input the output representations from both the f_1^A and f_1^B Level I Embedding units concatenated after the f_1^d Level I Embedding module outputs. Both the Cell-Specific Auxiliary task and the main difference tasks are trained end to end together.

Only Auxiliary Embedding as Features(Aux): For this variation, at the first level we use two Level I Embedding modules f_1^A and f_1^B corresponding to each cell-type. This is followed by two Level II Embedding modules f_2^A and f_2^B that take as input $f_1^A(\boldsymbol{X}^A)$ and $f_1^B(\boldsymbol{X}^B)$ respectively. The output of the Level II Embedding modules gives two final auxiliary embeddings $\boldsymbol{v}_A$, and $\boldsymbol{v}_B$ (Auxiliary-Task-A Embedding and Auxiliary-Task-B Embedding, respectively). These auxiliary embeddings are concatenated, $v = [\boldsymbol{v}_A, \boldsymbol{v}_B]$, and used as input to an MLP for the final prediction. For the auxiliary task predictions, the output $\boldsymbol{v}_A$ is passed

Model	Input Features	Auxiliary Task	Target Labels	Level I Embedding	Level II Embedding	Loss
Raw:d	$\boldsymbol{X}^A - \boldsymbol{X}^B$	-	differential expression	f_1	f_2	ℓ_{Diff}
Raw:c	$[\boldsymbol{X}^A, \boldsymbol{X}^B]$	-	differential expression	f_1	f_2	ℓ_{Diff}
Raw	$[\boldsymbol{X}^A, \boldsymbol{X}^B, \boldsymbol{X}^A - \boldsymbol{X}^B]$	-	differential expression	f_1	f_2	ℓ_{Diff}
Raw+Aux	$[\boldsymbol{X}^A, \boldsymbol{X}^B, \boldsymbol{X}^A - \boldsymbol{X}^B]$ and $\boldsymbol{X}^A, \boldsymbol{X}^B$	Cell-Specific Auxiliary	differential expression, gene expression A, gene expression B	f_1^d, f_1^A, f_1^B	f_2^d, f_2^A, f_2^B	$\ell_{Diff}+\ell_{CellAux}$
Aux	$\boldsymbol{X}^A, \boldsymbol{X}^B$	Cell-Specific Auxiliary	differential expression, gene expression A, gene expression B	f_1^A, f_1^B	f_2^A, f_2^B	$\ell_{Diff}+\ell_{CellAux}$
Aux+Siamese	$\boldsymbol{X}^A, \boldsymbol{X}^B$	Cell-Specific Auxiliary + Siamese Auxiliary	differential expression, gene expression A, gene expression B	f_1^A, f_1^B (shared weights)	f_2^A, f_2^B	$\ell_{Diff}+\ell_{CellAux}+\ell_{Siamese}$
Raw+Aux+Siamese	$[\boldsymbol{X}^A, \boldsymbol{X}^B, \boldsymbol{X}^A - \boldsymbol{X}^B], \boldsymbol{X}^A, \boldsymbol{X}^B$	Cell-Specific Auxiliary + Siamese Auxiliary	differential expression, gene expression A, gene expression B	f_1^d, f_1^A, f_1^B (shared weights for A and B)	f_2^A, f_2^B	$\ell_{Diff}+\ell_{CellAux}+\ell_{Siamese}$

Table 1. DeepDiff Variations in detail: The columns represent (a) different combinations of input features, (b) the auxiliary tasks used in the multitasking framework (Cell-Specific Auxiliary includes both the Auxiliary-Task-A and Auxiliary-Task-B), (c) the corresponding target labels for the tasks, and (d),(e) the model architecture of the variations: f_1 represents Level I Embedding module, and f_2 represents Level II Embedding module, and (f) the corresponding loss used to train the models.

through a linear layer for the prediction for cell-type A. Similarly, $\boldsymbol{v}_B$ is used as input to a linear layer for the prediction for cell-type B.

Siamese Auxiliary with Siamese Contrastive Loss(Aux+Siamese): Using the siamese contrastive loss formulation[4], we introduce a notion of similarity and dissimilarity based on a gene's differential gene expression. We consider the histone modification profiles $\boldsymbol{X}^A$ and $\boldsymbol{X}^B$ of two differentially expressed genes(upregulated or downregulated) to be 'different'($S = 1$) and 'similar'($S = 0$) for genes not differentially expressed. We introduce a contrastive loss term $\ell_{Siamese}$ as a regularizer, based on whether a gene is differentially regulated or not at the output embedding of the Level I Embedding unit f_1. We use the following formulation $\ell_{Siamese}$:

$$\ell_{Siamese} = (1 - S) \times \frac{1}{2} \times R + S \times \frac{1}{2} \max(0, m - R)^2 \quad (2)$$

where:

$$R = \sqrt{(f_1^A(\boldsymbol{X}^A) - f_1^B(\boldsymbol{X}^B))^2} \quad (3)$$

In Eq. 2, $m > 0$ is the margin in the contrastive loss and S indicates similarity or dissimilarity of the inputs i.e.,$S = 1$ if the gene is differentially expressed, and $S = 0$ if not differentially regulated. Contrastive Loss encourages 'similar' inputs to map to nearby points in the output representation space and 'dissimilar' inputs to map to distant points in the representation space. We use this $\ell_{Siamese}$ as a regularizer. We classify genes based on log change in differential gene expression$<= -2$(downregulated) or differential gene expression$>= 2$(upregulated) as differentially regulated($S = 1$) and log change in $-2 <=$differential gene expression$<= 2$ as $S = 0$. For this model, we use the Level I embedding unit as Siamese twin networks, i.e. f_1^A and f_1^B share their weights, while f_2^A and f_2^B (similar to the *aux* model) do not share weights.

Raw and Auxiliary Features with Siamese Contrastive Loss(Raw+Aux+Siamese): We further add the above contrastive loss formulation to the *Raw+Aux* model. We use the Level I Embeddings f_1^A and f_1^B as Siamese twin networks that share weights, and use the concatenation of the output Level I embeddings for the contrastive loss $\ell_{Siamese}$ in Eq. 2. In addition to f_1^A and f_1^B, we use f_1^d for the *Raw* features, similar to *Raw+Aux* model.

For the models with auxiliary tasks, *Raw+Aux* and *Aux*, we use the total loss $\ell = \ell_{Diff} + \ell_{CellAux}$. For *Aux+Siamese*, we use $\ell = \ell_{Diff} + \ell_{CellAux} + \ell_{Siamese}$. For the *Raw*, *Raw:c* and *Raw:d* models, we only use ℓ_{Diff} for optimizing the network. Table 1 shows the DeepDiff variations with corresponding architecture, target labels and loss variations. Figure 3 in Section 3.6 presents the variations as a combination of the DeepDiff main and Cell-Specific Auxiliary tasks.

1.4 Related Work

Table 2 compares DeepDiff with all the related studies discussed in Section 2 for the task of quantifying gene expression using HMs.

1.5 More about experimental setup

DeepDiff and baseline hyperparameters: For Level I Embedding, we use bidirectional LSTMs with hidden state size $D = 32$. Similarly, for bidirectional LSTMs in Level II Embedding modules, we use the hidden state size of 16. Since we implement a bi-directional LSTM, this results in each hidden state at Level I Embedding hidden state $\boldsymbol{h}_{jt}$ of size 64 and Level II Embedding hidden state s_j of size 32. Accordingly, we set the context vectors, $\boldsymbol{W}\boldsymbol{b}_j$ and $\boldsymbol{W}_h$, to size 64 and 32, respectively. We also use dropout, a regularization technique based on randomly dropping units from DNNs during training to prevent overfitting. We use a dropout probability of 0.5 for our experiments. We use hyperparameter $m = 2.0$ in our experiments for the *Aux+Siamese* and *Raw+Aux+Siamese* models with Siamese Auxiliary task (Equation (2)). For both the single and two-layer SVR models, we used cross-validation on varying hyperparameter values of $C \in \{0.1, 1, 10, 100\}$. We used radial basis kernel for SVR models. For the rest of the parameters, we used default settings in sklearn.

Evaluation Metric: We use Pearson Correlation Coefficient (PCC) to evaluate all our variations and baselines. PCC is a measure of the linear correlation between two continuous variables (predicted and target values in our experiments). It ranges between 1 and -1, where 1 is total positive linear correlation, 0 is no linear correlation, and -1 is total negative linear correlation.

1.6 More possible experiments: Classification as Cell-Specific Auxiliary Task

We also evaluate using classification labels for the Cell-Specific Auxiliary Task as opposed to regression. To formulate the labels in cell-type specific gene expression prediction in each cell-type as binary classification, we follow AttentiveChrome. In detail, for each cell type, we choose the cell type specific median of the rpkm gene expression as the threshold to classify the expression as 1 or -1. We use the log fold change in rpkm gene expression values as the regression label for the differential expression task. If the auxiliary task is classification, $v_A^{'}$, defined in Appendix Section 1.3 for Cell-Specific Auxiliary task, will be fed to a softmax output layer. To train this classification auxiliary task, we minimize the negative log likelihood loss. Figure 2 shows the PCC for all model variations for $classification$ of cell-type specific gene expression as auxiliary tasks and Table 3 shows the relative performance (%) with respect to Pearson Correlation Coefficient(PCC) when comparing *Aux* and *Raw+Aux* models to two-layer SVR. Because rpkm is a cell type specific normalization, we

Computational Study	Differential	Unified	Non-linear	Bin-Info	Representation Learning		Feature Inter.	Interpretable	Output
					Neighbor Bins	Whole Region			
Linear Regression ([6])	×	×	×	×	×	✓	×	✓	Regression
SVR (single layer) ([1])	✓	×	✓	Bin-specific	×	✓	✓	×	Regression
SVR (two layer) ([1])	✓	×	✓	×	×	✓	✓	×	Regression
SVM ([2])	✓	×	✓	Bin-specific	×	✓	✓	×	Classification
Random Forest ([3])	×	×	✓	Best-bin	×	✓	×	×	Classification/Regression
ReliefF+Random Forest ([10])	✓	×	✓	×	×	✓	×	×	Classification
Rule Learning ([5])	×	×	✓	×	×	✓	✓	✓	No prediction
DeepChrome-CNN [11]	×	✓	✓	Automatic	✓	✓	✓	×	Classification
AttentiveChrome[12]	×	✓	✓	Automatic	✓	✓	✓	✓	Classification
DeepDiff (this study)	✓	✓	✓	Automatic	✓	✓	✓	✓	Regression

Table 2. Comparison of previous studies for the task of quantifying gene expression using histone modification marks (adapted from [11]). The columns indicate (a) whether the it is a differential gene expression or cell type specific gene expression prediction study, (b) whether the study has a unified end-to-end architecture or not (c) if it captures non-linearity among features (d) how has the bin information been incorporated (e) if representation of features is modeled on local and global scales, (f) if combinatorial interactions among histone modifications are modeled, (h) if the model is interpretable, and (g) the output formulation of the study.

use rpkm as the target label in this case for consistency with the labels for Cell-Specific Auxiliary tasks.

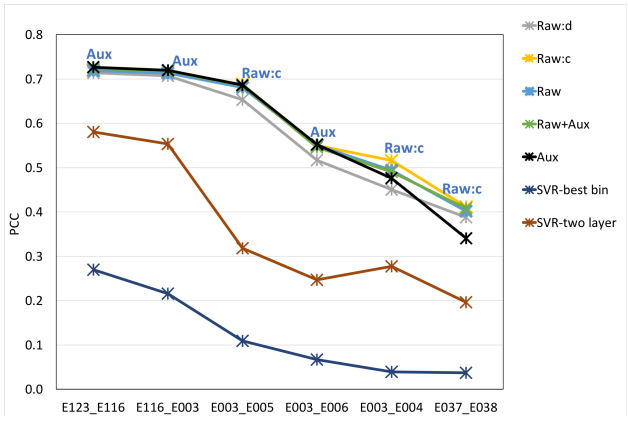

Fig. 2. Cell-Specific Auxiliary as $classification$: Pearson correlation (PCC) for DeepDiff main task and multi-tasking with Cell-Specific Auxiliary as classification for six cell-type pairs. The text label for each cell-type pair is the best performing DeepDiff variation.

Method	Mean	Median
Aux	173.20	172.58
Raw+Aux	179.17	192.33

Table 3. Relative performance with Cell-Specific Auxiliary as $classification$: Mean and Median of the relative performance (%) with respect to Pearson Correlation Coefficient(PCC) when comparing DeepDiff multitasking with classification based Cell-Specific Auxiliary models to one of the best-performing baselines: two-layer SVR across six cell-type pairs.

Histone Mark	Associated with Regions
H3K4me3	Promoter
H3K4me1	Enhancer
H3K36me3	Transcribed
H3K9me3	Heterochromatin
H3K27me3	Polycomb Repression

Table 4. Five core histone modifications as defined by [9] with associated regions on the genome.

REMC Id	Cell type
E123	K562
E116	GM12878
E003	H1 Cell Line
E004	H1 BMP4 Derived Mesendoderm Cultured Cells
E005	H1 BMP4 Derived Trophoblast Cultured Cells
E006	H1 Derived Mesenchymal Stem Cells
E037	CD4 Memory Primary Cells
E038	CD4 Naive Primary Cells
E007	H1 Derived Neuronal Progenitor Cultured Cells

Table 5. The cell-types (and corresponding REMC ID) used in the experiments.

References

[1]Chao Cheng and Mark Gerstein. Modeling the relative relationship of transcription factor binding and histone modifications to gene expression levels

Pairs(Tasks)	Cell type A	Cell type B
1	E123	E003
2	E116	E003
3	E123	E116
4	E003	E005
5	E003	E006
6	E006	E007
7	E005	E006
8	E003	E004
9	E004	E006
10	E037	E038

Table 6. The Cell-type pairs we use in our experiments.

in mouse embryonic stem cells. *Nucleic acids research*, 40(2):553–568, 2011.

[2]Chao Cheng, Koon-Kiu Yan, Kevin Y Yip, Joel Rozowsky, Roger Alexander, Chong Shou, Mark Gerstein, et al. A statistical framework for modeling gene expression using chromatin features and application to modencode datasets. *Genome Biol*, 12(2):R15, 2011.

[3]Xianjun Dong, Melissa C Greven, Anshul Kundaje, Sarah Djebali, James B Brown, Chao Cheng, Thomas R Gingeras, Mark Gerstein, Roderic Guigó, Ewan Birney, et al. Modeling gene expression using chromatin features in various cellular contexts. *Genome Biol*, 13(9):R53, 2012.

[4]Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pp. 1735–1742. IEEE, 2006.

[5]Bich Hai Ho, Rania Mohammed Kotb Hassen, and Ngoc Tu Le. Combinatorial roles of dna methylation and histone modifications on gene expression. In *Some Current Advanced Researches on Information and Computer Science in Vietnam*, pp. 123–135. Springer, 2015.

[6]Rosa Karlić, Ho-Ryun Chung, Julia Lasserre, Kristian Vlahovičekand Martin Vingron. Histone modification levels are predictive for gene expression. *Proceedings of the National Academy of Sciences*, 107(7):2926–2931, 2010.

[7]Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2014.

[8]Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[9]Anshul Kundaje, Wouter Meuleman, Jason Ernst, Misha Bilenky, Angela Yen, Alireza Heravi-Moussavi, Pouya Kheradpour, Zhizhuo Zhang, Jianrong Wang, Michael J Ziller, et al. Integrative analysis of 111 reference human epigenomes. *Nature*, 518(7539):317–330, 2015.

[10]Jeffery Li, Travers Ching, Sijia Huang, and Lana X Garmire. Using epigenomics data to predict gene expression in lung cancer. In *BMC bioinformatics*, volume 16, p. S10. BioMed Central, 2015.

[11]Ritambhara Singh, Jack Lanchantin, Gabriel Robins, and Yanjun Qi. Deepchrome: deep-learning for predicting gene expression from histone modifications. *Bioinformatics*, 32(17):i639–i648, 2016.

[12]Ritambhara Singh, Jack Lanchantin, Arshdeep Sekhon, and Yanjun Qi. Attend and predict: Understanding gene regulation by selective attention on chromatin. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pp. 6785–6795. Curran Associates, Inc., 2017.