Automated distillation of genomic equations governing single cell gene expression

Edouardo Honig
Department of Statistics & Data Science
University of California, Los Angeles
e.honig@ucla.edu

Frederique Ruf-Zamojski
Department of Medicine
Cedars-Sinai Medical Center
Frederique.Ruf-Zamojski@cshs.org

Stuart Sealfon
Department of Neurology
Icahn School of Medicine at Mount Sinai
stuart.sealfon@mssm.edu

Ying Nian Wu
Department of Statistics & Data Science
University of California, Los Angeles
ywu@stat.ucla.edu

Zijun Zhang
Division of Artificial Intelligence in Medicine
Cedars-Sinai Medical Center
Zijun.Zhang@cshs.org

Abstract

Gene expression is an essential cellular process that is controlled by a complex and orchestrated regulatory network of transcription factors and epigenetic modifications. The advancement in single-cell RNA sequencing enables the investigation of gene expression control at an unprecedented fine resolution and large scale. Yet, understanding the sequence determinants underlying distinct primary cell types remains elusive and challenging. While deep neural networks have shown strong performance in predicting gene expression, the lack of meaningful explanations of predictions, especially in systematic understanding of the molecular mechanisms, motivates the search for more transparent models. We present an automated model that predicts gene expression from genetic sequences while providing both strong performance and direct interpretations of predictions. Our model combines a pretrained genetic sequence class model and neural architecture search with symbolic regression to distill explainable genomic equations. We applied our method to an inhouse human pituitary (a specialized gland in the brain that controls the endocrine system) single-cell gene expression data. The distilled genomic equation prediction accuracy (Pearson r=0.713) is comparable to other explainable models, without artificially introducing strong inductive bias that may not hold for the complex and potentially non-linear cellular system. The genomic equations shed light on how sequence classes interact and regulate the cell type-specific, finely-controlled transcriptomic program in the human endocrine system. To our knowledge, this is the first attempt at distilling genomic equations from neural networks using symbolic regression.

1 Introduction

The conversion of genetic information to instructions for synthesizing RNA resulting in proteins is known as gene expression [1]. As a fundamental process conserved in all known lifeforms, gene expression is controlled by a complex and orchestrated regulatory network of transcription factors

NeurIPS 2023 AI for Science Workshop.

and epigenetic modifications [2]. Collectively, a coordinated set of gene expression changes form a cellular transcription program, that defines the identity and function of a cell. Within an organism, despite all cells sharing the same genome encoded in DNA, the transcription program are diverse and highly specific for cells of different types [3]. Understanding the finely-controlled gene transcription program has been challenging – historically, experimental technologies are restricted to profile bulk tissues with a mixture of different cell types, therefore losing the signals from individual cells and cell types [4]. Recent single-cell and single-nuclei RNA sequencing (scRNA-seq/snRNA-seq) advances have increased both the amount and granularity of available gene expression data. Unlike conventional experiments conducted in bulk tissues, scRNA-seq profiles the relative gene expression levels in each individual cell [5]. Typically, each scRNA-seq library measures approximately 3,000-4,000 genes per cell across tens of thousands of cells [6]. The high-dimensional and data-rich scRNA-seq provides suitable basis for machine learning to decipher the complex transcriptional control.

Many recent advances in predicting gene expression use deep neural networks [7–10] to improve predictive performance while providing genomic insights on tasks such as variant effect prediction and identification of functional elements in non-coding regions of genes. While these models are increasingly powerful, it is also more difficult to explain the predictions of so-called black box models. Black box models such as Enformer [10] may output highly correlated predictions for some genomic variants, yet Sasse et al. [11] found that predictions for up to 43% of genes were anti-correlated with the measured gene expression. While explainable artificial intelligence (XAI) methods can be applied to interpret incorrect predictions of such black box models, the output from the XAI process is only feature attributions [12]. Without more explicit relational explanations for predictions, it is difficult for these empirical feature importance findings to be used to construct general theories or inform scientific understanding. Despite empirical advances, the details of the relationship between genetic sequence and gene expression are still a mystery.

To address this challenge, explainable symbolic models hold the promise of discovering systematic knowledge on cellular transcription programs. A more explainable alternative exists in linear or symbolic models, but these generally are outperformed by deep neural networks. In light of this, [13, 14] have successfully distilled known and novel equations governing physical laws in the field of astrophysics from deep neural networks using symbolic regression. However, to our knowledge, there have not yet been any attempts to do the same in high-dimensional genomic sequences.

We propose a model that automatically produces genomic equations predicting gene expression from sequences by distilling a neural network [13, 14] trained on genetic sequence classes generated from a pre-trained model [15]. We use Neural Architecture Search (NAS) to tune and train a deep neural network with a latent bottleneck. Two sets of symbolic regression are then performed, to explain the mapping from input to latent space, and latent space to gene expression, which are joined together to result in genomic equations that predict gene expression from sequence classes. Our method is applied to human pituitary single-cell gene expression data, and aims to improve the understanding of the interaction between sequence classes and genes actively expressed in the human endocrine system.

2 Related Work

Predicting Gene Expression from Sequence. While gene expression prediction from sequence data has been studied using conventional machine learning methods, including Bayesian networks and Bayes classifiers [16, 17], deep learning approaches have increased in recent years [8, 9, 18]. Deep convolutional neural networks have been trained on increasingly long-range genetic sequences to directly predict gene expression, allowing the model to take into account both the coding and non-coding regions of input genes [7–9]. More recently, Transformer-based architectures [19] such as Enformer [10] have improved over deep convolutional neural networks by further expanding input sequence size and improving explainability of results with attention visualization. However, this direct sequence to gene expression method has been shown to fail to generalize in certain cases [11], motivating the search for other models that may generalize better.

Regulatory Sequence Models. In contrast to directly modeling primary DNA sequences for gene expression prediction, an alternative strategy uses a two-stage approach: a deep neural network is used to obtain epigenomic features such as chromatin profiles [18, 20, 21], which are then used to train additional models such as linear models to predict gene expression. In comparison to neural networks trained directly on sequence data without a meaningful latent space, the representations learned by regulatory sequence models can be used to identify changes in genetic sequence that may exist, but not affect gene expression [18]. While not useful for predicting gene expression, this model still offers insight into genome understanding. In particular, Chen et al. [15] introduce Sei, a deep model that takes as input 4,096 nucleotide sequences and outputs 21,907 chromatin profiles, which can be directly converted into 40 regulatory activities (sequence classes), identified by clustering the chromatin profile predictions from 30 million sequence comprising the human genome. Compared to existing regulatory sequence models [18, 20, 21], Sei has substantially improved the quality and abstraction of learned representations from a perspective of global classification and quantification of sequence activities.

Neural Architecture Search. Neural Architecture Search (NAS) is a method to automatically identify high performing neural network model architectures, removing the need to manually adjust model layers and continually train new models to find the best-performing configuration. Various searching algorithms have been developed to efficiently search optimal architectures from a predefined model space, including reinforcement learning-based (RL-based) [22], black-box optimizers of Bayesian optimization and evolutionary algorithms [23], and differentiable methods [24]. Searchers that use a RL-based controller are straightforward to generalize to composite reward functions [25, 26], while other searchers (e.g. differentiable) are potentially restricted to using accuracy as a reward.

Symbolic Regression. By searching a space of variables, mathematical operators, functions, and constants, symbolic regression aims to identify mathematical equations that explain the relationship between inputs and outputs by generating and recombining equations. Models learned with symbolic regression are more powerful and potentially generalize better than linear regression. There exist many processes that do not have a direct linear relationship between their independent and dependant variables. Since linear regression assumes a linear relationship between inputs and outputs, it can be difficult to obtain strong models of non-linear processes using linear regression. In contrast, symbolic regression does not make any model assumptions, and is more suitable for modeling non-linear relationships. The search in symbolic regression may be conducted with Bayesian methods [27] or neural networks [28], but is often performed using a genetic algorithm approach [29, 30]. When fitting equations with symbolic regression, random sets of expressions are generated and evaluated in an iterative fashion, such that the highest performing expressions are mutated until a pre-determined condition is met. It should be noted that the search space grows exponentially with the number of input variables, potentially drastically increasing the amount of time required for search to obtain a strong model. In this work we distill genomic equations from a neural network using symbolic regression.

3 Methods

Data collection and preprocessing. The data used to train our model was obtained from a collaborator in the form of single-cell gene expression data. Gene expression was averaged over cell types to produce the initial target gene expression values. Since many genes are not expressed for certain cell types, including the one of focus in our study, we transform the target data so that many targets are not effectively zero to improve training. We apply a log-transform then normalize the initial target gene expression values to the range [-1, 1] to produce the actual targets used during training, denoted by g, where g_0 represents the initial target gene expression: $g_1 = \log_2(g_0 + pseudocount)$, $g = \frac{g_1 - \min(g_1)}{\max(g_1) - \min(g_1)} \times 2 - 1$, where $pseudocount = 0.001$.

To obtain the genetic sequence data for each gene, the Matched Annotation from NCBI and EMBL-EBI (MANE) [31] transcription start sites (TSSes) were identified for each gene using the GRCh38 reference genome assembly. A 4,096-nucleotide window centered on the TSS of each gene was then extracted to be used as input to our model.

Feature construction and model space definition. To predict gene expression from sequence data, we can construct a neural network that takes one-hot encoded nucleotides from gene sequences and outputs gene expression predictions, as in [7–10]. However, since we aim to distill more explainable genomic equations using symbolic regression, it is necessary to reduce the dimensionality of the inputs due to the exponential complexity of symbolic regression with respect to number of input variables. Following Zhou et al. [18] and [21], we choose to use a two-stage process to train a model that predicts gene expression. Therefore, we use the pre-trained Sei [15] to obtain 40 sequence class embeddings of 4,096-nuclueotide genetic sequence data centered on gene transcription start sites. These sequence classes are used to train a neural network with a latent bottleneck to predict gene expression. The latent bottleneck reduces the dimension of the input from 40, decreasing the size of the search space of a symbolic regression from the latent space to the gene expression (output) space. In order to distill genomic equations from the neural network, symbolic regression is also performed on the sequence class (input) data to predict the latent representations from the trained neural network.

NAS to optimize gene expression prediction. We use NAS to train the neural network to optimize for the latent space size, activation functions used in each layer, degree of dropout, and to prevent model collapse in early stages of model tuning. The model space is comprised by a set of 3 fully connected layers with ReLU, tanh, GeLU, or sigmoid activation followed by dropout, a fully connected layer to a lower latent dimension with the aforementioned activation functions, followed by another set of 3 fully connected layers identically defined as prior to the latent layer. For the NAS, the Adam optimizer [32] was used with learning rate equal to 0.001, and the loss function was mean-squared error (MSE). The reward optimized for during the NAS is the Pearson correlation of the network's predictions with the measured values, which is also used to evaluate our model. We implement NAS in Python using AMBER [33] with PyTorch [34].

Genomic equation distillation. Symbolic regression is used to obtain equations that explicitly relate the Sei sequence classes to gene expression. Specifically, two symbolic regressions are performed sequentially, the first of which distills equations relating the sequence classes to the latent representation of the neural network. The second symbolic regression maps from the predictions of the first symbolic regression of the neural network, to the output gene expression space. While it is feasible to perform the two symbolic regressions in parallel by training the latter regression on the latent representations from the neural network, we decided against this. Since the distilled equations from the first regression may not exactly match the network from which they were distilled, but will be part of the final model, we choose to use the distilled predictions for the latent space to distill the final equations from latent to output space. We implement both symbolic regressions using PySR [30] with the following operators: addition, multiplication, subtraction, square, negation, exponential, and inverse, and select the equations with lowest MSE loss. We limit the complexity of distilled equations to 20, where one unit of complexity is defined as the instance of a single variable, operator, or constant in an equation, the default in PySR [30].

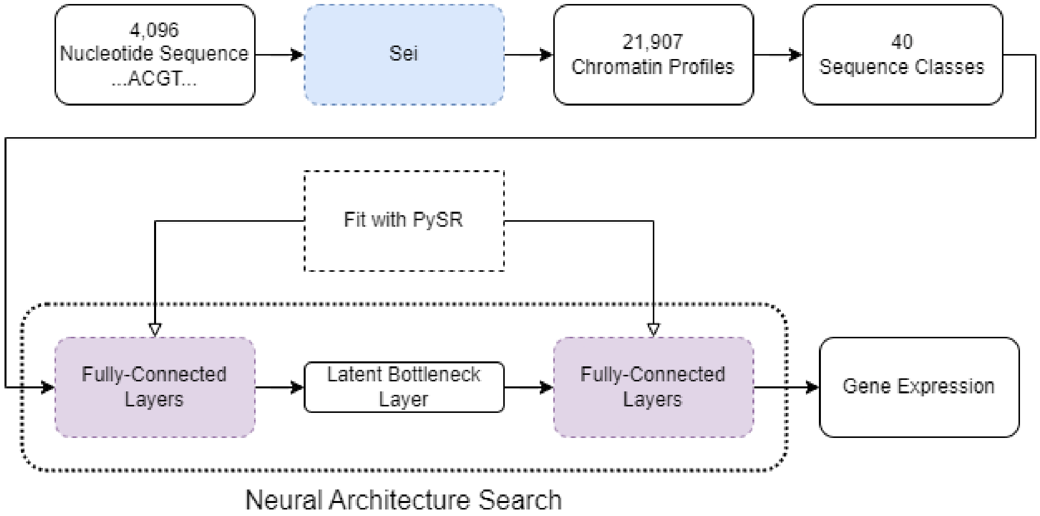

Figure 1: Overview of the automated genomic equation distillation framework

Baselines and Experimental details. We evaluate our results using the Pearson correlation coefficient between the predicted and measured gene expression values for each method. We first automatically select the best performing neural network architecture with NAS, using the Pearson correlation of the network directly as the reward for the NAS. Then, we distilled genomic equations from this neural network using symbolic regression **(SR(NAS | NN))**. Additionally, we report the Pearson correlation coefficient for several baselines: **ExPectoSC** [21], a regularized linear regression **(Ridge)**, a hand-tuned neural network without NAS (**NN**), the NAS-trained neural network the genomic equations are distilled from **(NAS | NN)**, and genomic equations from a symbolic regression trained on the 40 sequence classes to directly predict gene expression **(SR)**.

4 Results

We apply our method to gene expression data from our in-house pituitary snRNA-seq data, which consists of 76,929 cells with gene expression sparsely measured for 33,330 genes, such that most genes are not expressed in each cell. Specifically, we use one of the most abundant cell types in human pituitary, Gonadotropes, and took the average gene expression over Gonadotropes cells for robust cell type expression quantification. After matching genes from our data to the MANE TSSes [31], we obtain a total of 17,771 genes for which appropriate genetic sequences were available. A random subset of 20% of genes are held out for final model evaluation, and the other 80% of genes are randomly split for training (72%) and validation (8%).

Table 1: Performance Evaluation

Model	Explainable	Pearson Correlation (↑)
ExPectoSC [21]	✓	0.718
Ridge Regression	✓	0.717
SR	✓	0.696
NN	✗	0.717
NAS | SR	✗	0.742
NAS | NN	✗	0.751
SR(NAS | SR)	✓	0.709
SR(NAS | NN)	✓	0.713

4.1 NAS optimizes gene expression prediction

Automation via NAS makes it more feasible to explore a larger set of architectures from a model space of n=170,859,375 combinations. We use NAS to search for the optimal number of parameters in both the latent space and the fully connected layers, as well as to identify the highest performing activation functions for each layer and the number of layers. Compared to a naive manually-selected architecture, NAS improved neural network test Pearson correlation coefficient from 0.717 to 0.751, as seen in Table 1.

Since a key motivation of our work is to generate genomic equations explaining neural network predictions, we also include an ablation where the NAS is informed by distilled genomic equations for each architecture explored in the model space. The NAS is performed with its reward being the Pearson correlation from distilled genomic equations during the architecture search (**NAS | SR**). In this way, the final architecture found by the NAS will have been selected based on the predictive power of the genomic equations that could be distilled from it. However, in practice, this method of performing symbolic regression to obtain reward during NAS has limited benefits. Since the search space for symbolic regression is too expansive to quickly fit meaningful equations for the model architectures explored during NAS, this is more computationally expensive than the other methods and does not offer any clear benefits in terms of quality of distilled equations (**SR(NAS | SR)**).

4.2 Distilled symbolic regression is predictive

In the process of searching for the best equations using symbolic regression, many candidate models are explored. At the end of the search, equations of several different complexities are available to be selected. PySR [30] offers multiple suggested methods to select the best equations among the different complexities. While we select the genomic equations that minimize the MSE loss (the *accuracy* criterion), the PySR defined *best* criterion selects equations that have the highest score, defined as the negated derivative of the log-loss with respect to complexity, among equations with a loss at least 50% better than the model with highest accuracy [30].

Table 2: Latent Equation Complexity Analysis and Test Performance

Complexity (per latent equation)	Number of Sequence Classes	Pearson Correlation (↑)
3	4	0.643
5	4	0.636
6	6	0.662
7	11	0.662
9	13	0.680
18	19	0.706
best	8	0.676
accuracy	**17**	**0.713**

Table 2 displays the complexity, number of input variables, and performance of the distilled equations that model the neural network's latent space from genetic sequence classes (**SR(NAS | NN)**). The number of sequence classes and Pearson correlation are calculated on test data using the lowest loss distilled equation that models gene expression given the network's latent representations. The *accuracy* or lowest loss selection criteria results in the strongest performance, and it is shown in Table 1 that less than half of the 40 sequence classes are necessary to reach comparable performance to other explainable models.

The symbolic regression conducted directly on the sequence classes was the simplest and lowest performing model, resulting in an equation that uses only 6/40 sequence classes, below.

$$\hat{g} = 0.193 \cdot (X_{TF4} - 1.16)\left((X_{E10} - 0.725)^2 + X_{PC3} + X_{L6}\right) + 0.193 X_{L4} + 0.193 X_{HET6} - 0.0121$$

Notably, X_{E10} represents the enhancer sequence class in Brain; X_{TF4} represents a specific transcription factor, OTX2, that is essential for the normal development of brain, eye [35] and pituitary gland [36]. These terms are consistent with the biological origin of Gonadotropes cell type in the pituitary gland. Future investigations will determine the specificity of these sequence classes. The labels for each sequence class are in Appendix A, and the distilled genomic equations are in Appendix B.

The above equation is less complex than the equations distilled from the neural networks, but the best performing distilled genomic equation is also a function of the six sequence classes above. However, the model distilled from the symbolic regression informed neural network contains only 5/6 of the sequence classes used in the direct symbolic regression, as its predictions do not depend on sequence class L4. Noting that L4 represents a low signal class, indicating low enrichment in the histone markers measured in [15], this difference in equation dependencies is understandable in context. The inclusion of another low signal class L5 may explain the absence of L4 in the aforementioned equation.

4.3 Relaxation of linear assumption improves modeling of highly-expressed genes

We hypothesize that more expressive genes will have higher nonlinear interactions with underlying genetic sequence. Overall, highly-expressed genes are not well captured by either the linear or the nonlinear model, likely due to the training data imbalance; we do not observe statistical differences between the error residuals from ridge regression and our distilled genomic equation (Appendix Fig. 3). To investigate this hypothesis, we instead design two subsets of genes for which either the ridge regression (linear subset) or the genomic equation (nonlinear subset) has substantially higher predictive power. These subsets were identified by selecting genes for which a model's error was lower than 75% of genes, while the other model's error was higher than 50% of its predictions. The gene names within each subset for each model are listed in Appendix D. With this, we hope to identify genes where the relationship between sequence class and expression is either strongly linear or nonlinear. A two-sample t-test indicates there is a statistically significant (p=0.030) difference in the means of the two subsets, where the genes in the nonlinear subset have higher gene expression on average. This finding is in-line with our hypothesis, and potentially consistent with current biological understandings of synergistic gene expression regulation [37, 38].

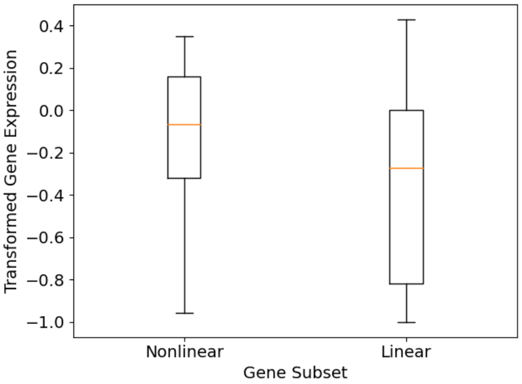

Figure 2: Boxplots of gene subset expression. Nonlinear subset has a significantly higher expression level than linear subset (p=0.030, t-test).

4.4 Additional Data

In addition to the Gonadotropes cell type, we test our method on two other cell types from our single-cell gene expression data: Somatotropes and Stem Cells. We find that our distilled equations perform similarly to ridge regression, while ExPectoSC performs more strongly. As can be seen in Appendix B, these genomic equations are also functions of the sequence classes that are related to the Brain and Promoter, similar to those distilled for the Gonadotropes. We hope to identify further commonalities between genomic equations across cell types in the future.

Table 3: Performance Evaluation: Additional Cell Types

		Somatotropes	Stem Cells
Model	Explainable	Pearson Correlation (↑)	
ExPectoSC [21]	✓	0.728	0.730
Ridge Regression	✓	0.720	0.708
NN	✗	0.720	0.714
NAS I NN	✗	0.746	0.749
SR(NAS I NN)	✓	0.709	0.702

5 Conclusion

In this work, we introduce an automated method of distilling genomic equations from a neural network predicting gene expression from gene sequence classes. We evaluated the performance of both the neural network and its genomic equations distilled via symbolic regression and found they were comparable. Our results imply that symbolic regression may have further use in understanding the relationship between genetic sequences and gene expression. The distilled genomic equations help explain the neural network predictions, and may improve understanding of the importance of different input sequence classes and their effect on gene expression. Furthermore, as the first work to our knowledge that applies symbolic regression to distill genomic equations from a neural network, we hope to introduce and motivate applications of similar methods to single-cell RNA sequencing tasks.

References

[1] Gary H Perdew, John P Vanden Heuvel, and Jeffrey M Peters. *Regulation of gene expression: molecular mechanisms*. Springer, 2006.

[2] Matthew V Rockman and Leonid Kruglyak. Genetics of global gene expression. *Nature Reviews Genetics*, 7(11):862–872, 2006.

[3] Lu Chen, Myrto Kostadima, Joost HA Martens, Giovanni Canu, Sara P Garcia, Ernest Turro, Kate Downes, Iain C Macaulay, Ewa Bielczyk-Maczynska, Sophia Coe, et al. Transcriptional diversity during lineage commitment of human blood progenitors. *Science*, 345(6204):1251033, 2014.

[4] Zhong Wang, Mark Gerstein, and Michael Snyder. Rna-seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57–63, 2009.

[5] Fuchou Tang, Catalin Barbacioru, Yangzhou Wang, Ellen Nordman, Clarence Lee, Nanlan Xu, Xiaohui Wang, John Bodeau, Brian B Tuch, Asim Siddiqui, et al. mrna-seq whole-transcriptome analysis of a single cell. *Nature methods*, 6(5):377–382, 2009.

[6] Xin Chen, Zhaowei Yang, Wanqiu Chen, Yongmei Zhao, Andrew Farmer, Bao Tran, Vyacheslav Furtak, Malcolm Moos Jr, Wenming Xiao, and Charles Wang. A multi-center cross-platform single-cell rna sequencing reference dataset. *Scientific Data*, 8(1):39, 2021.

[7] David R Kelley, Jasper Snoek, and John L Rinn. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome research*, 26(7):990–999, 2016.

[8] David R Kelley, Yakir A Reshef, Maxwell Bileschi, David Belanger, Cory Y McLean, and Jasper Snoek. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome research*, 28(5):739–750, 2018.

[9] Vikram Agarwal and Jay Shendure. Predicting mrna abundance directly from genomic sequence using deep convolutional neural networks. *Cell reports*, 31(7), 2020.

[10] Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18(10):1196–1203, 2021.

[11] Alexander Sasse, Bernard Ng, Anna Spiro, Shinya Tasaki, David A. Bennett, Christopher Gaiteri, Philip L. De Jager, Maria Chikina, and Sara Mostafavi. How far are we from personalized gene expression prediction using sequence-to-expression deep neural networks? *bioRxiv*, 2023. doi: 10.1101/2023.03.16.532969. URL `https://www.biorxiv.org/content/early/2023/03/20/2023.03.16.532969`.

[12] Ian Covert, Scott Lundberg, and Su-In Lee. Feature removal is a unifying principle for model explanation methods. *arXiv preprint arXiv:2011.03623*, 2020.

[13] Miles D Cranmer, Rui Xu, Peter Battaglia, and Shirley Ho. Learning symbolic physics with graph networks. *arXiv preprint arXiv:1909.05862*, 2019.

[14] Miles Cranmer, Alvaro Sanchez Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. Discovering symbolic models from deep learning with inductive biases. *Advances in Neural Information Processing Systems*, 33:17429–17442, 2020.

[15] Kathleen M Chen, Aaron K Wong, Olga G Troyanskaya, and Jian Zhou. A sequence-based global map of regulatory activity for deciphering human genetics. *Nature genetics*, 54(7): 940–949, 2022.

[16] Michael A Beer and Saeed Tavazoie. Predicting gene expression from sequence. *Cell*, 117(2): 185–198, 2004.

[17] Yuan Yuan, Lei Guo, Lei Shen, and Jun S Liu. Predicting gene expression from sequence: a reexamination. *PLoS computational biology*, 3(11):e243, 2007.

[18] Jian Zhou, Chandra L Theesfeld, Kevin Yao, Kathleen M Chen, Aaron K Wong, and Olga G Troyanskaya. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nature genetics*, 50(8):1171–1179, 2018.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[20] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10):931–934, 2015.

[21] Ksenia Sokolova, Chandra L Theesfeld, Aaron K Wong, Zijun Zhang, Kara Dolinski, and Olga G Troyanskaya. Atlas of primary cell-type-specific sequence models of gene expression and variant effects. *Cell Reports Methods*, 2023.

[22] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

[23] Kirthevasan Kandasamy, Karun Raju Vysyaraju, Willie Neiswanger, Biswajit Paria, Christopher R Collins, Jeff Schneider, Barnabas Poczos, and Eric P Xing. Tuning hyperparameters without grad students: Scalable and robust bayesian optimisation with dragonfly. *The Journal of Machine Learning Research*, 21(1):3098–3124, 2020.

[24] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[25] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2820–2828, 2019.

[26] Zijun Zhang, Linqi Zhou, Liangke Gou, and Ying Nian Wu. Neural architecture search for joint optimization of predictive power and biological knowledge. *arXiv preprint arXiv:1909.00337*, 2019.

[27] Ying Jin, Weilin Fu, Jian Kang, Jiadong Guo, and Jian Guo. Bayesian symbolic regression. *arXiv preprint arXiv:1910.08892*, 2019.

[28] Silviu-Marian Udrescu and Max Tegmark. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.

[29] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85, 2009.

[30] Miles Cranmer. Interpretable machine learning for science with pysr and symbolicregression.jl, 2023.

[31] Joannella Morales, Shashikant Pujar, Jane E Loveland, Alex Astashyn, Ruth Bennett, Andrew Berry, Eric Cox, Claire Davidson, Olga Ermolaeva, Catherine M Farrell, et al. A joint ncbi and embl-ebi transcript set for clinical genomics and research. *Nature*, 604(7905):310–315, 2022.

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[33] Zijun Zhang, Christopher Y Park, Chandra L Theesfeld, and Olga G Troyanskaya. An automated framework for efficiently designing deep convolutional neural networks in genomics. *Nature Machine Intelligence*, 3(5):392–400, 2021.

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[35] Francis Beby and Thomas Lamonerie. The homeobox gene otx2 in development and disease. *Experimental eye research*, 111:9–16, 2013.

[36] Amanda H Mortensen, Vanessa Schade, Thomas Lamonerie, and Sally A Camper. Deletion of otx2 in neural ectoderm delays anterior pituitary development. *Human molecular genetics*, 24 (4):939–953, 2015.

[37] Jinmi Choi, Kseniia Lysakovskaia, Gregoire Stik, Carina Demel, Johannes Söding, Tian V Tian, Thomas Graf, and Patrick Cramer. Evidence for additive and synergistic action of mammalian enhancers during cell fate determination. *Elife*, 10:e65381, 2021.

[38] Jessica Zuin, Gregory Roth, Yinxiu Zhan, Julie Cramard, Josef Redolfi, Ewa Piskadlo, Pia Mach, Mariya Kryzhanovska, Gergely Tihanyi, Hubertus Kohler, et al. Nonlinear control of transcription through enhancer–promoter interactions. *Nature*, 604(7906):571–577, 2022.

Appendix

A Sei Sequence Classes [15]

Sequence class label	Sequence class name	Rank by size	Group
PC1	Polycomb / Heterochromatin	0	PC
L1	Low signal	1	L
TN1	Transcription	2	TN
TN2	Transcription	3	TN
L2	Low signal	4	L
E1	Stem cell	5	E
E2	Multi-tissue	6	E
E3	Brain / Melanocyte	7	E
L3	Low signal	8	L
E4	Multi-tissue	9	E
TF1	NANOG / FOXA1	10	TF
HET1	Heterochromatin	11	HET
E5	B-cell-like	12	E
E6	Weak epithelial	13	E
TF2	CEBPB	14	TF
PC2	Weak Polycomb	15	PC
E7	Monocyte / Macrophage	16	E
E8	Weak multi-tissue	17	E
L4	Low signal	18	L
TF3	FOXA1 / AR / ESR1	19	TF
PC3	Polycomb	20	PC
TN3	Transcription	21	TN
L5	Low signal	22	L
HET2	Heterochromatin	23	HET
L6	Low signal	24	L
P	Promoter	25	P
E9	Liver / Intestine	26	E
CTCF	CTCF-Cohesin	27	CTCF
TN4	Transcription	28	TN
HET3	Heterochromatin	29	HET
E10	Brain	30	E
TF4	OTX2	31	TF
HET4	Heterochromatin	32	HET
L7	Low signal	33	L
PC4	Polycomb / Bivalent stem cell Enh	34	PC
HET5	Centromere	35	HET
E11	T-cell	36	E
TF5	AR	37	TF
E12	Erythroblast-like	38	E
HET6	Centromere	39	HET

B Highest Accuracy Distilled Genomic Equations

SR(NAS | NN)

$$
\begin{aligned}
x_0 &= 0.174\left(X_{TN4} + X_{E10} - X_{PC4} + X_{HET5} - X_{E11} - X_{E12}\right) - 0.174\left(X_P - 0.206\right)^2 \\
x_2 &= \left(X_{PC3}^4 + X_{TN1} - X_{E10}^2\right)\left(1.42X_{L6}^2 - X_{E10}^2 + 1.12X_{E1}\right) \\
x_4 &= 0.114\left(X_{E10} - X_{TF4} - X_{E12} + X_{HET6} - e^{X_{L1} - X_P - X_{E10}}\right) + 0.0819 \\
x_6 &= 0.0753X_{L4} - 0.151X_{PC3} - 0.0753X_{HET2} - 0.0753\left(X_{E10} + X_{HET6} - e^{X_{E6}}\right)^2
\end{aligned}
$$

$$\hat{g} = x_0 + x_6 + \left(-x_4 - 0.450\right)\left(-1.43x_0e^{x_0} + x_2 + x_6 + 0.0639\right)$$

SR(NAS | SR)

$$
\begin{aligned}
x_0 &= e^{-2.01\left(0.996X_{E10}+e^{-X_{L1}+X_P}\right)^2\left(X_{TF2}^2-X_{E7}+e^{X_P}\right)^2} \\
x_2 &= e^{-0.360(0.882X_{E10}+1)^2\left(X_{HET4}+e^{X_{E1}+e^{-X_{L6}}}\right)^2} \\
x_3 &= e^{-X_{L2}+X_{E1}-\left(-X_{HET5}+\left(X_{L7}-X_{L3}+e^{X_{L5}}\right)^2+0.618\right)^4} \\
x_7 &= \left(X_{HET5} + 0.0632 + e^{-\left(-X_{E10}+X_{L7}+X_{E11}+e^{X_{TF4}-X_{L3}}\right)^2}\right)^2 \\
x_8 &= \left(X_{HET5} - X_{E12} + e^{X_{E6}-\left(X_{L5}-0.905X_{L6}^2+e^{-X_{TN1}}\right)^2}\right)^2
\end{aligned}
$$

$$\hat{g} = -0.434x_0 + 0.434x_7 - 0.144e^{x_2^2-x_3+x_7^2-x_8}$$

SR(NAS | NN): Somatotropes

$$
\begin{aligned}
x_3 &= -X_{E6} - X_{PC3} - 0.517X_{E10}^2 + 2X_{E10} + 0.517X_{HET4} + 0.248 \\
x_4 &= X_{HET6} - e^{-X_{HET1}-X_{PC2}+X_{L7}-e^{-X_{L5}-X_{PC4}-X_{E12}+X_{E1}^2+X_{L3}}} \\
x_6 &= 0.212X_{L5} - 0.212X_{TN4} + 0.212\left(X_{PC3} + \left(X_{E10} - 0.928\right)^2\right)^2 - 0.212e^{\frac{1}{0.702}} \\
x_7 &= -X_{HET6} + \left(-X_{E10} + X_{TF4}\right)\left(2.31X_{L4} + X_{E10} - 3.31X_{PC4} - 2.31X_{E12} - 0.305\right) \\
x_9 &= \left(-X_{TN4}\left(-X_{HET2}^2 + e^{X_{HET6}}\right) + 1.47\right)\left(-X_{E6} + X_P + X_{E10} + X_{PC4}X_{HET6}\right)
\end{aligned}
$$

$$\hat{g} = 0.161x_3 + 0.161x_4 - 0.161x_6 - 0.161x_7 + 0.0711\left(-x_7 - 0.700x_9\right)^2 - 0.292$$

SR(NAS | NN): Stem Cells

$$
\begin{aligned}
x_2 &= 9.82\left(0.603X_{HET5} - 0.603\left(-X_{TN4} + X_{HET4}\right)e^{X_{HET6}} - 0.603e^{X_{E12}} + 1\right)^4 - 0.161 \\
x_3 &= \left(X_{E10} + \left(-0.157X_{L1} - X_{E12} + X_{HET6} + e^{-4\left(-X_{TN4}+X_{E11}\right)^2}\right)^2\right)^2 \\
x_6 &= -1.44X_{HET3}^4 + \left(X_{E6} + X_{PC3} - X_{E10} + \left(0.724 - X_{E1}\right)^2\right)^2 - 0.157 \\
x_8 &= \left(X_{PC3} - X_{E1} + \frac{1}{1.59}\right)^2\left(-X_{E10}^4 + X_{E1} - e^{X_{L5}}\right)^2 - 0.164
\end{aligned}
$$

$$\hat{g} = e^{-0.685x_8-0.685e^{2x_2x_6\left(x_2-0.927\right)-x_3}} - 0.816$$

C Model Residual Plots

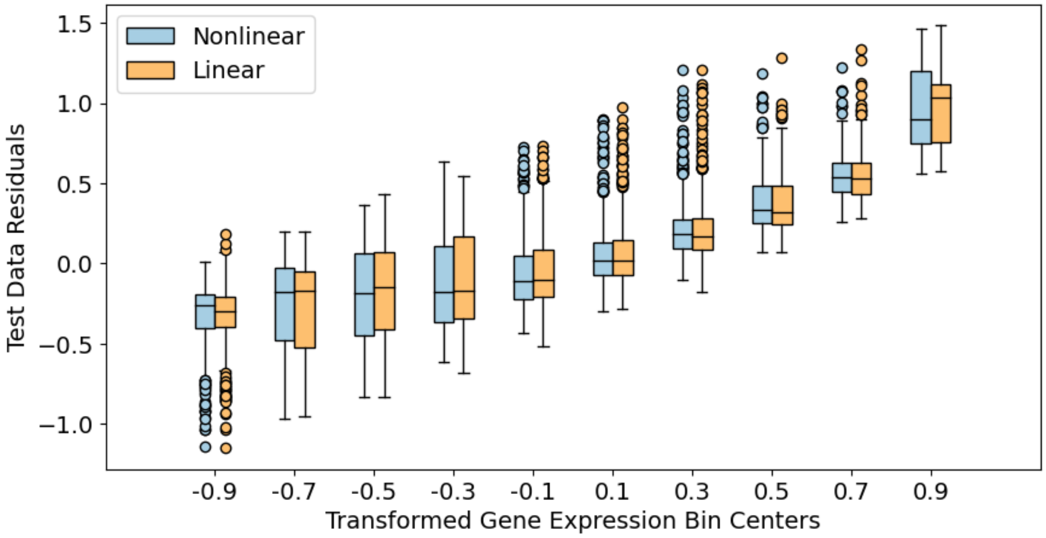

Figure 3: Examination of model residuals, bin width=0.2. No significant differences were identified in residuals across all bins.

D Gene subsets

D.1 Nonlinear (Genomic Equation)

GBP1, HEPN1, ZNF793, ACSL1, PLCH2, UBE2F, SYCE1L, NUDT19, PIP, COTL1, CMTM1, LIME1, ASB15, C2orf49, RNASEL, ZNF583, SDC4, MYH9, PRR15L, SUPV3L1, GCNT1, ERAP2, ETV3, RSKR, GRAMD1C, CTXN2, OSBPL2, TKT, NLRP1

D.2 Linear (Ridge Regression)

CD164, NAA25, IL4R, FAM124B, CGB7, EBAG9, KCTD17, KLF10, LSMEM2, FCER1G, CARD6, HINT1, POLM, RNASEH2B, ZNF418, ACADM, GPR155, ACTL8, TTYH1, MBNL3, HBM, SRSF8, LEP, LACTBL1, FGF7, KLK4, CADM4, NBPF20, CWH43, SH2D3A, FLRT3, HOPX, POLR1E, RETN, RHOG, ITK, TSPAN33, TGFB1, RAB14, F2RL3, ANGPTL7, ZNF555, PLD6, SLC22A8, GP1BA, DDX19B, SLC16A14, AGAP2, FGF19