The Effects of Nonlinear Signal on Expression-Based Prediction Performance

This manuscript ([permalink]) was automatically generated from [greenelab/linear_models_manuscript@bca5224] on June 9, 2022.

Authors

- **Benjamin J. Heil**
  0000-0002-2811-1031 · ben-heil · autobencoder
  Genomics and Computational Biology Graduate Group, Perelman School of Medicine, University of Pennsylvania · Funded by Grant XXXXXXXX

- **Jake Crawford**
  0000-0001-6207-0782 · jjc2718 · jjc2718
  Genomics and Computational Biology Graduate Group, Perelman School of Medicine, University of Pennsylvania · Funded by Grant XXXXXXXX

- **Casey S. Greene**
  0000-0001-8713-9213 · cgreene · greenescientist
  Department of Pharmacology, University of Colorado School of Medicine; Department of Biochemistry and Molecular Genetics, University of Colorado School of Medicine · Funded by Grant XXXXXXXX

Abstract

Those building predictive models from transcriptomic data are faced with two conflicting perspectives. The first, based on the inherent high dimensionality of biological systems, supposes that complex nonlinear models such as neural networks will better match complex biological systems. The second, imagining that complex systems will still produce low-rank decision surfaces prefers simpler models that are easier to interpret. We compare multi-layer neural networks and logistic regression across multiple prediction tasks on GTEx and Recount3 datasets and find evidence in favor of both possibilities. We verified the presence of nonlinear signal for transcriptomic prediction tasks by removing the predictive linear signal with Limma. This experiment ablates the performance of linear methods but not non-linear ones. However, we also found that the presence of nonlinear signal was not necessarily sufficient for neural networks to outperform logistic regression. Our results demonstrate that while multi-layer neural networks may be useful for making predictions from gene expression data, including a linear baseline model is critical because while biological systems are high-dimensional, many decision surfaces for predictive models may not be.

Introduction

Transcriptomic data contains a wealth of information about biology. Gene expression-based models are already being used for subtyping cancer [1], predicting transplant rejections [2], and uncovering biases in public data [3]. In fact, both the capability of machine learning models [4] and the amount of transcriptomic data available [5,6] are increasing rapidly. It makes sense, then, that neural networks are frequently being used to build predictive models from transcriptomic data [7,8,9].

However, there are two conflicting ideas in the literature regarding the utility of nonlinear models. One theory draws on prior biological understanding: the paths linking gene expression to phenotypes are complex [10,11], and nonlinear models like neural networks should be more capable of learning that complexity. Unlike purely linear models such as logistic regression, nonlinear models should learn more sophisticated representations of the relationships between expression and phenotype. Accordingly, many have used nonlinear models to learn representations useful for making predictions of phenotypes from gene expression [12,13,14].

The other supposes that even high-dimensional complex systems may produce linear decision surfaces. This is supported empirically: linear models seem to do as well as or better than nonlinear ones in many cases [15]. While papers of this sort are harder to come by — perhaps scientists do not tend to write papers about how their deep learning model was worse than logistic regression — other complex biological problems have also seen linear models prove equivalent to nonlinear ones [16].

We design experiments to ablate linear signal and find merit to both hypotheses. We construct a system of binary and multi-class classification problems on the GTEx and Recount3 compendia [17,18] that shows linear and nonlinear models have similar accuracy on several prediction tasks. However, we then remove the linear signals relating the phenotype to gene expression and find nonlinear signal in the data even when the linear models outperform the nonlinear ones. Given the unexpected nature of these findings, we evaluated independent tasks, examined different problem formulations, and constructed simulated data. Results were consistent with this model across these settings.

In reconciling these two ostensibly conflicting theories, confirm the importance of implementing and optimizing a linear baseline model before deploying a complex nonlinear approach. While nonlinear models may outperform simpler models at the limit of infinite data, they do not necessarily do so even when trained on the largest datasets publicly available today.

Results

Linear and nonlinear models have similar performance in many tasks

We compared the performance of linear and nonlinear models across multiple datasets and tasks (fig. 1 top). We examined using gene expression to predict tissue labels from GTEx [17], tissue labels from Recount3 [18], and labels of genetic sex from Flynn et al. [19]. We filtered scRNA samples, and also applied zero-one standardization after TPM normalization. To avoid leakage between cross-validation folds, we placed entire studies into single folds (fig. 1 bottom). We evaluated methods on subsampled datasets to determine the extent to which performance was affected by the amount of training data.

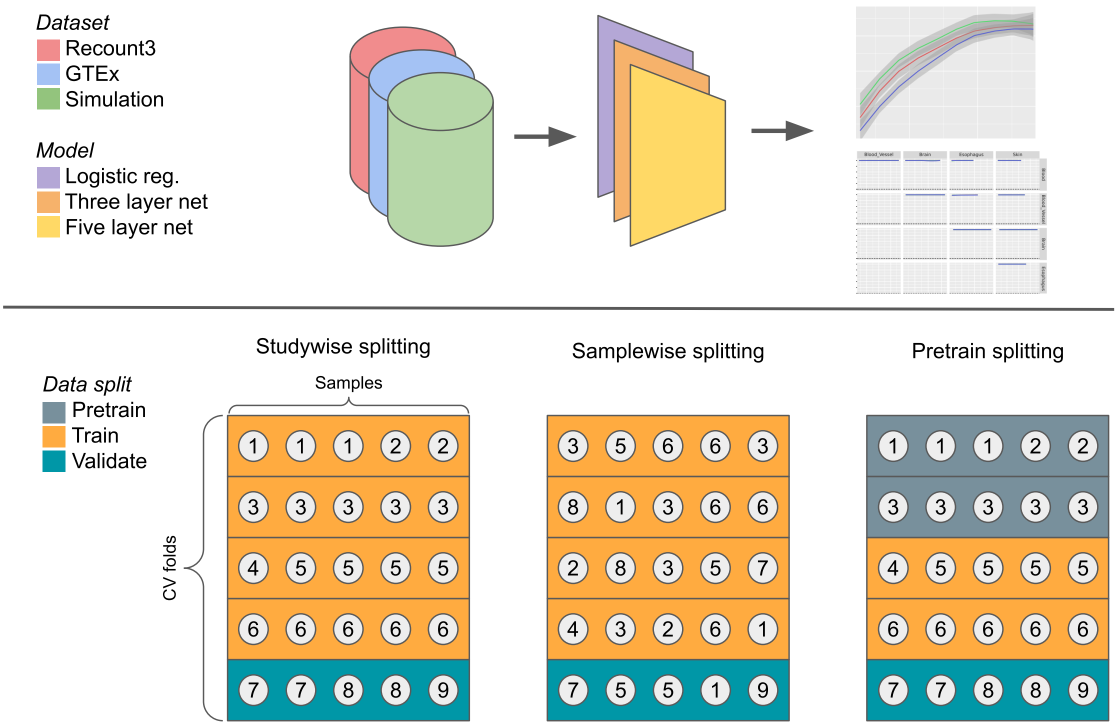

Figure 1: Schematic of the model analysis workflow. We evaluate three models on multiple classification problems in three datasets (top). We use study-wise splitting by default and evaluate the effects of sample-wise splitting and pre-training.

We used GTEx [17] to determine whether linear and nonlinear models performed similarly on a well-characterized dataset with consistent experimental protocols across samples. We first trained our models to differentiate between tissue types on pairs of the five most common tissues in the dataset. Likely due to the clean nature of the data, all models were able to perform perfectly on these binary classification tasks (fig. 2 top right). Because binary classification was unable to differentiate between models, we evaluated the models on a more challenging task. We tested the models on their ability to perform multiclass classification on all 31 tissues present in the dataset. In the multitask setting, logistic regression slightly outperformed the five-layer neural network, which in turn slightly outperformed the three-layer net (fig. 2 top left).

We then evaluated the same approaches in a dataset with very different characteristics: Sequence Read Archive [20] samples from Recount3 [18]. We compared the models' ability to differentiate between pairs of tissues (supp. fig. 5) and found their performance was roughly equivalent. We also evaluated the models' performance on a multiclass classification problem differentiating between the 21 most common tissues in the dataset. As in the GTEx setting, the logistic regression model outperformed the five-layer network, which outperformed the three-layer network (fig. 2 bottom left).

To examine whether these results held in a problem domain other than tissue type prediction, we tested performance on metadata-derived sex labels (fig. 2 bottom right), a task previously studied by Flynn et al. [19]. We used the same experimental setup as in our other binary prediction tasks to train the models, but rather than using tissue labels we used sex labels from Flynn et al.. In this setting we found that while the models all performed similarly, the nonlinear models tended to have a slight edge over the linear one.

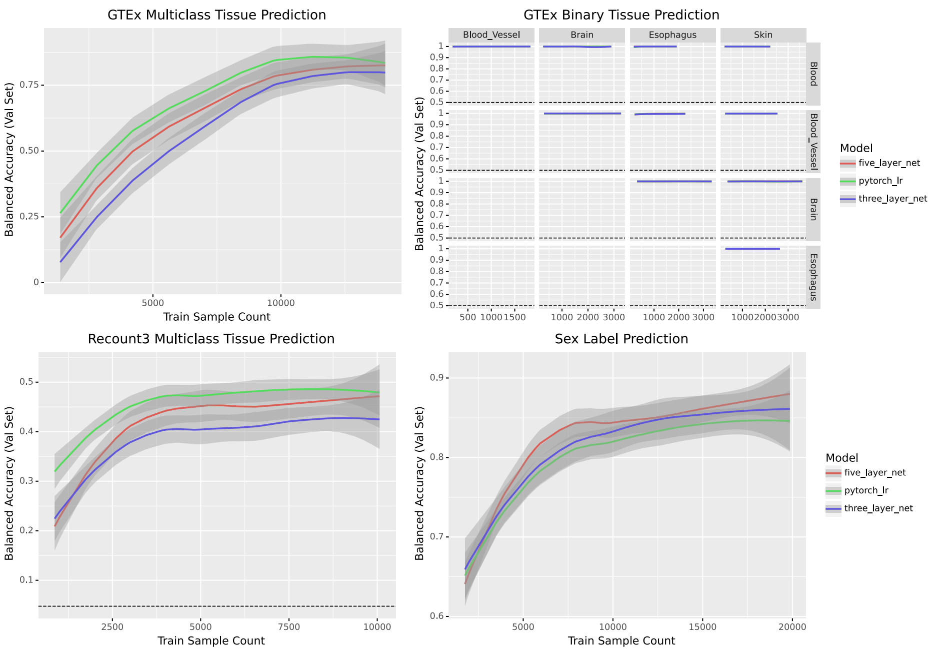

Figure 2: Performance of models across four classification tasks. In each panel the loess curve and its 95% confidence interval are plotted based on points from three seeds, ten data subsets, and five folds of studywise cross-validation (for a total of 150 points per model per panel).

There is predictive nonlinear signal in transcriptomic data

Our results to this point are consistent with a model where the predictive signal present in transcriptomic data is entirely linear. If that were the case, nonlinear models like neural networks would fail to give any substantial advantage. We first simulated three datasets to better understand model performance for a variety of linear or nonlinear data generating processes. We created data with both linear and nonlinear signal by generating two types of features: half of the features with a linear dividing line between the simulated classes and half with a nonlinear dividing line (see Methods for more detail). After training to classify the simulated dataset, all models effectively predicted the simulated classes. To determine whether or not there was nonlinear signal, we then used Limma [21] to remove the linear signal associated with the endpoint being predicted. After removing the linear signal from the dataset, nonlinear models correctly predicted classes, but logistic regression performed no better than random (fig 3 middle).

To confirm that non-linear signal was key to the performance of non-linear methods, we generated another simulated dataset consisting solely of features with a linear dividing line between the classes. As before, all models were able to predict the different classes well. However, once the linear signal was removed, all models performed no better than random guessing (fig 3 left). That the nonlinear models only achieved baseline accuracy also indicated that the signal removal method was not injecting nonlinear signal into data where nonlinear signal did not exist. We also trained the models on a dataset where all features were Gaussian noise as a negative control. As expected, the models all performed at baseline accuracy both before and after the signal removal process (fig. 3 right). This experiment supported our decision to perform signal removal on the training and validation sets separately. Removing the signal in the full dataset introduced predictive signal into this setting (supp. fig. 6).

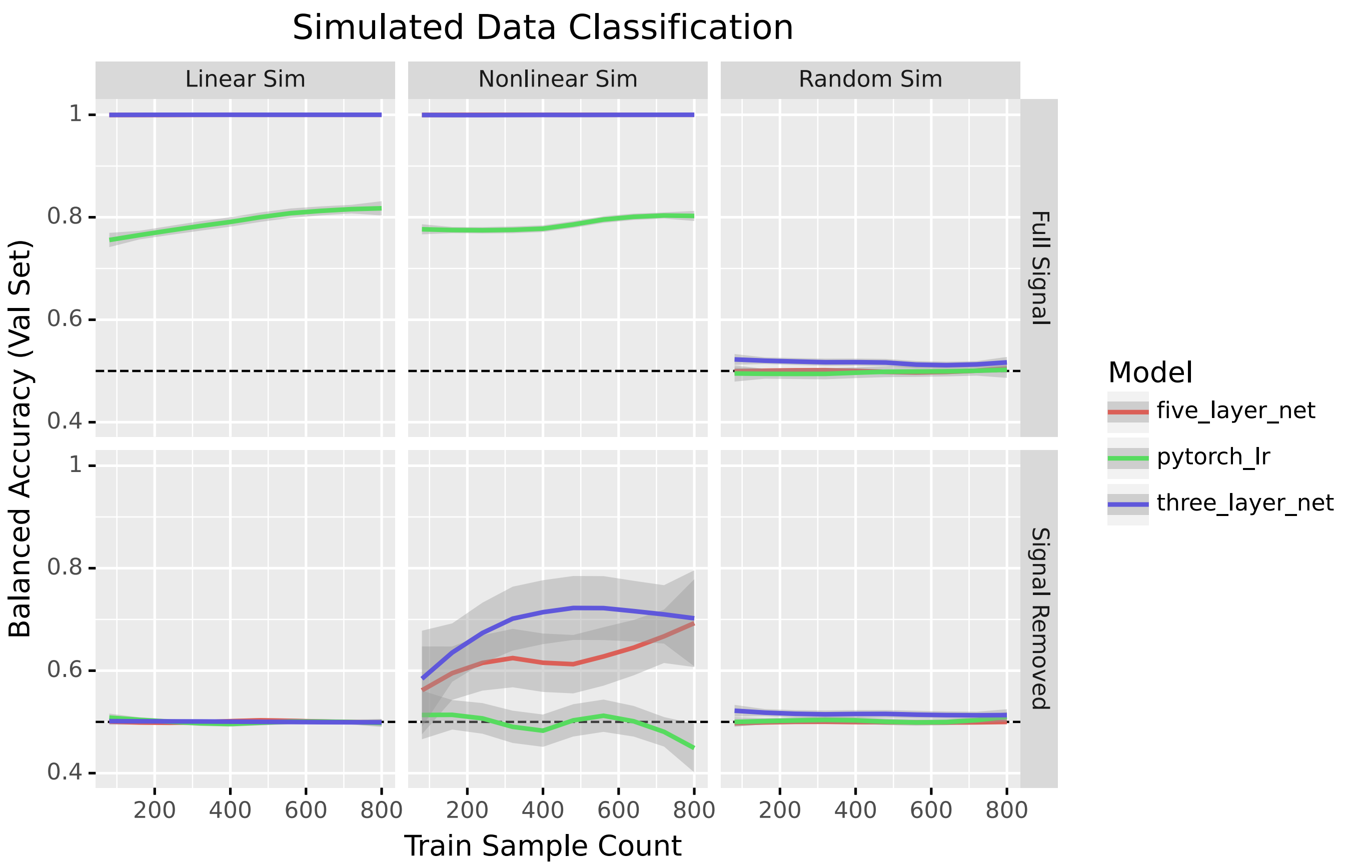

Figure 3: Performance of models in binary classification of simulated data before and after signal removal. Dotted lines indicate expected performance for a naive baseline classifier that predicts the most frequent class.

We next removed linear signal from GTEx and Recount3. We found that the neural nets performed better than the baseline while logistic regression did not (fig. 4 top right, supp. fig. 7). Similarly, for multiclass problems logistic regression performing poorly, while the nonlinear models had performance that increased with an increase in data while remaining worse than before the linear signal was removed (fig. 4 left). Likewise, the sex label prediction task showed a marked difference between the neural networks and logistic regression: only the neural networks could learn from the data (fig. 4 bottom right). In each of the settings, the models performed less well than when run on data without signal removal, indicating an increase in the problem's difficulty, and logistic regression, in particular, performed no better than random.

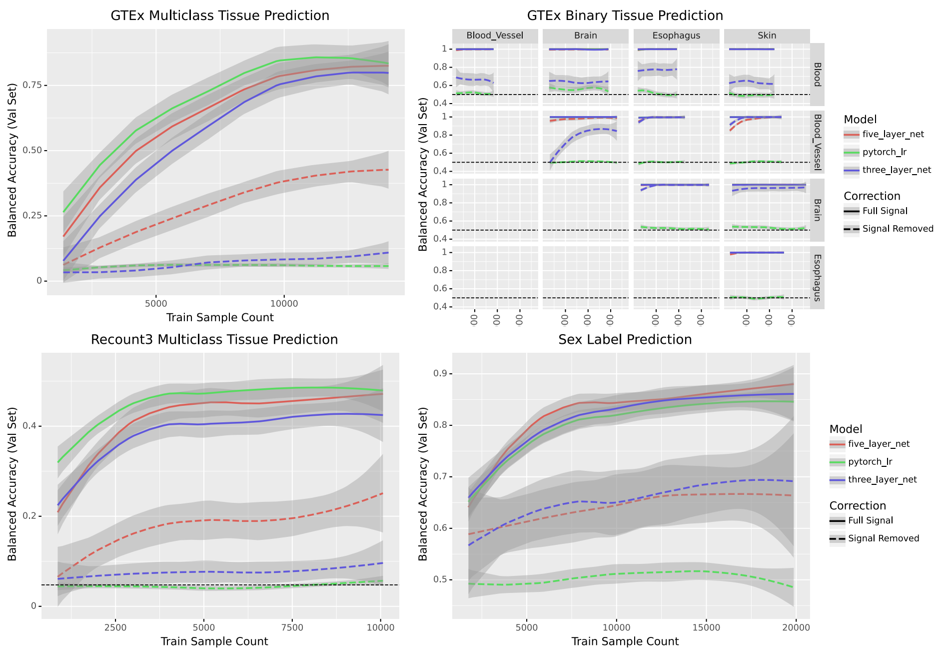

Figure 4: Performance of models across four classification tasks before and after signal removal

To verify that our results were not an artifact of our decision to assign studies to cross-validation folds, we compared the study-wise splitting that we used with an alternate method called samplewise splitting. Samplewise splitting (see Methods) is common in machine learning, but can leak information between the training and validation sets when samples are not independently and identically distributed among studies - a common feature of data in biology. We found that sample-wise splitting induced substantial performance inflation (supp. fig. 8). The relative performance of each model stayed the same regardless of the data splitting technique, so the results observed were not dependent on the choice of splitting technique.

Another growing strategy in machine learning, especially on biological data where samples are limited, is training models on a general-purpose dataset and fine-tuning them on a dataset of interest. We examined the performance of models with and without pretraining (supp. fig 9). We split the Recount3 data into three sets: pretraining, training, and validation (fig. 1 bottom), then trained two identically initialized copies of each model. One was trained solely on the training data, while the other was trained on the pretraining data and fine-tuned on the training data. The pretrained models showed high performance even when trained with small amounts of data from the training set. However, the nonlinear models did not have a greater performance gain from pretraining than logistic regression, and the balanced accuracy was similar across models.

Methods

Datasets

GTEx

We downloaded the 17,382 TPM-normalized samples of bulk RNA-seq expression data available from version 8 of GTEx. We zero-one standardized the data and retained the 5000 most variable genes. The tissue labels we used for the GTEx dataset were derived from the 'SMTS' column of the sample metadata file.

Recount3

We downloaded RNA-seq data from the Recount3 compendium [22] during the week of March 14, 2022. Before filtering, the dataset contained 317,258 samples, each containing 63,856 genes. To filter out single-cell data, we removed all samples with greater than 75 percent sparsity. We also removed all samples marked 'scrna-seq' by Recount3's pattern matching method (stored in the metadata as 'recount_pred.pattern.predict.type'). We then converted the data to transcripts per kilobase million using gene lengths from BioMart [23] and performed standardization to scale each gene's range from zero to one. We kept the 5,000 most variable genes within the dataset.

We labeled samples with their corresponding tissues using the 'recount_pred.curated.tissue' field in the Recount3 metadata. These labels were based on manual curation by the Recount3 authors. A total of 20324 samples in the dataset had corresponding tissue labels. Samples were also labeled with their corresponding sex using labels from Flynn et al. [3]. These labels were derived using pattern matching on metadata from the European Nucleotide Archive [24]. A total of 23,525 samples in our dataset had sex labels.

Data simulation

We generated three simulated datasets. The first dataset contained 1000 samples of 5000 features corresponding to two classes. Of those features, 2500 contained linear signal. That is to say that the feature values corresponding to one class were drawn from a standard normal distribution, while the feature values corresponding to the other were drawn from a Gaussian with a mean of 6 and unit variance.

We generated nonlinear features similarly. The values for the nonlinear features were drawn from a standard normal distribution for one class, while the second class had values drawn from either a mean six or negative six Gaussian with equal probability. These features are referred to as "nonlinear" because two dividing lines are necessary to perfectly classify such data, while a linear classifier can only draw one such line per feature.

The second dataset was similar to the first dataset, but it consisted solely of 2500 linear features. The final dataset contained only values drawn from a standard normal distribution regardless of class label.

Model architectures

We used three representative models to demonstrate the performance profiles of different model classes. Each model was implemented in Pytorch [25], used the same optimizer, and was trained for at most 50 epochs.

The nonlinear models were fully connected neural networks. The first was a three-layer network with hidden layers of sizes 2500 and 1250. Our second was a five-layer network, with hidden layers of sizes 2500, 2500, 2500, and 1250. Both models used ReLU nonlinearities [26].

The final model was an implementation of logistic regression, a linear model. As there are known differences in performance between implementations of logistic regression [27], we implemented ours in PyTorch as similarly to the neural nets as possible to allow for a fair comparison.

Model training

Optimization
Our models minimized the cross-entropy loss using an Adam [28] optimizer on mini-batches of data. They also used inverse frequency weighting to avoid giving more weight to more common classes.

Regularization
The models used early stopping and gradient clipping to regularize their training. Both neural nets used dropout [29] with a probability of 0.5. The deeper network used batch normalization [30] to mitigate the vanishing gradient problem.

Signal removal
We used Limma[21] to remove linear signal associated with tissues in the data. More precisely, we ran the 'removeBatchEffect' function from Limma on the training and validation sets separately, using the tissue labels as batch labels.

Hyperparameters
The learning rate and weight decay hyperparameters for each model were selected via nested cross-validation over the training folds at runtime.

Determinism
Model training was made deterministic by setting the Python, NumPy, and PyTorch random seeds for

each run, as well as setting the PyTorch backends to deterministic and disabling the benchmark mode.

Logging
Model training progress was tracked and recorded using Neptune [31].

Model Evaluation

In our analyses we use five-fold cross-validation with studywise data splitting. In a studywise split, the studies are randomly assigned to cross-validation folds such that all samples in a given study end up in a single fold (fig. 1 bottom).

Hardware
Our analyses were performed on an Ubuntu 18.04 machine and the Colorado Summit compute cluster. The desktop CPU used was an AMD Ryzen 7 3800xt processor with 16 cores and access to 64 GB of RAM, and the desktop GPU used was an Nvidia RTX 3090. The Summit cluster used Intel Xeon E5-2680 CPUs and NVidia Tesla K80 GPUs. From initiating data download to finishing all analyses and generating all figures, the full Snakemake [32] pipeline took around one month to run.

Recount3 tissue prediction
In the Recount3 setting, the multi-tissue classification analyses were trained on the 21 tissues (see Supp. Methods) that had at least ten studies in the dataset. Each model was trained to determine which of the 21 tissues a given expression sample corresponded to.

To address class imbalance, our models' performance was then measured based on the balanced accuracy across all classes. Unlike raw accuracy, balanced accuracy (the mean across all classes of the per-class recall) isn't predominantly determined by performance on the largest class in an imbalanced class setting. For example, in a binary classification setting with 9 instances of class A and 1 instance of class B, successfully predicting 8 of the 9 instances of class A and none of class B yields an accuracy of 0.8 but a balanced accuracy of 0.44.

The binary classification setting was similar to the multi-class one. The five tissues with the most studies (brain, blood, breast, stem cell, and cervix) were compared against each other pairwise. The expression used in this setting was the set of samples labeled as one of the two tissues being compared.

The data for both settings were split in a stratified manner based on their study.

GTEx classification
The multi-tissue classification analysis for GTEx used all 31 tissues. The multiclass and binary settings were formulated and evaluated in the same way as in the Recount3 data. However, rather than being split studywise, the cross-validation splits were stratified according to the samples' donors.

Simulated data classification/sex prediction
The sex prediction and simulated data classification tasks were solely binary. Both settings used balanced accuracy, as in the Recount3 and GTEx problems.

Pretraining
When testing the effects of pretraining on the different model types, we split the data into three sets. Approximately forty percent of the data went into the pretraining set, forty percent went into the training set, and twenty percent went into the validation set. The data was split such that each study's samples were in only one of the three sets to simulate the real-world scenario where a model is trained on publicly available data and then fine-tuned on a dataset of interest.

To ensure the results were comparable, we made two copies of each model with the same weight initialization. The first copy was trained solely on the training data, while the second was trained on the pretraining data, then the training data. Both models were then evaluated on the validation set. This process was repeated four more times with different studies assigned to the pretraining, training, and validation sets.

Conclusion

We performed a series of analyses to determine the relative performance of linear and nonlinear models across multiple tasks. Consistent with previous papers [15,16], linear and nonlinear models performed roughly equivalently in a number of tasks. That is to say that there are some tasks where linear models perform better, some tasks where nonlinear models have better performance, and some tasks where both model types are equivalent.

However, When we removed all linear signal in the data, we found that residual nonlinear signal remained. This was true in simulated data as well as GTEx and Recount3 data across several tasks. These results also held in altered problem settings, such as using a pretraining dataset before the training dataset and using samplewise data splitting instead of studywise splitting. This consistent presence of nonlinear signal demonstrated that the similarity in performance across model types was not due to our problem domains having solely linear signals.

Given that nonlinear signal is present in our problem domains, why doesn't that signal allow nonlinear models to make better predictions? It is possible that the nonlinear signal is either entirely redundant with the linear signal or unreliable enough that nonlinear methods prioritize learn the linear signal when it is present.

One limitation of our study is that the results likely do not hold in an infinite data setting. Deep learning models have been shown to solve complex problems in biology and tend to significantly outperform linear models when given enough data. However, we do not yet live in a world in which millions of well-annotated examples are available in many areas of biology. Our results are generated on some of the largest labeled expression datasets in existence (Recount3 and GTEx), but our tens of thousands of samples are far from the millions or billions used in deep learning research.

We are also unable to make claims about all problem domains. There are many potential transcriptomic prediction tasks and many datasets to perform them on. While we show that nonlinear signal is not always helpful in tissue or sex prediction, and others have shown the same for various disease prediction tasks, there may be problems where nonlinear signal is more important.

Ultimately, our results show that task-relevant nonlinear signal in the data, which we confirm is present, does not necessarily lead nonlinear models to outperform linear ones. Additionally, our results suggest that scientists making predictions from expression data should always include simple linear models as a baseline to determine whether more complex models are warranted.

Acknowledgements

We would like to thank Alexandra Lee and Jake Crawford for reviewing code that went into this project. We would also like to thank the past and present members of GreeneLab who gave feedback on this project during lab meetings. This work utilized resources from the University of Colorado Boulder Research Computing Group, which is supported by the National Science Foundation (awards ACI-1532235 and ACI-1532236), the University of Colorado Boulder, and Colorado State University.

Supplementary Materials

Results

Recount binary classification

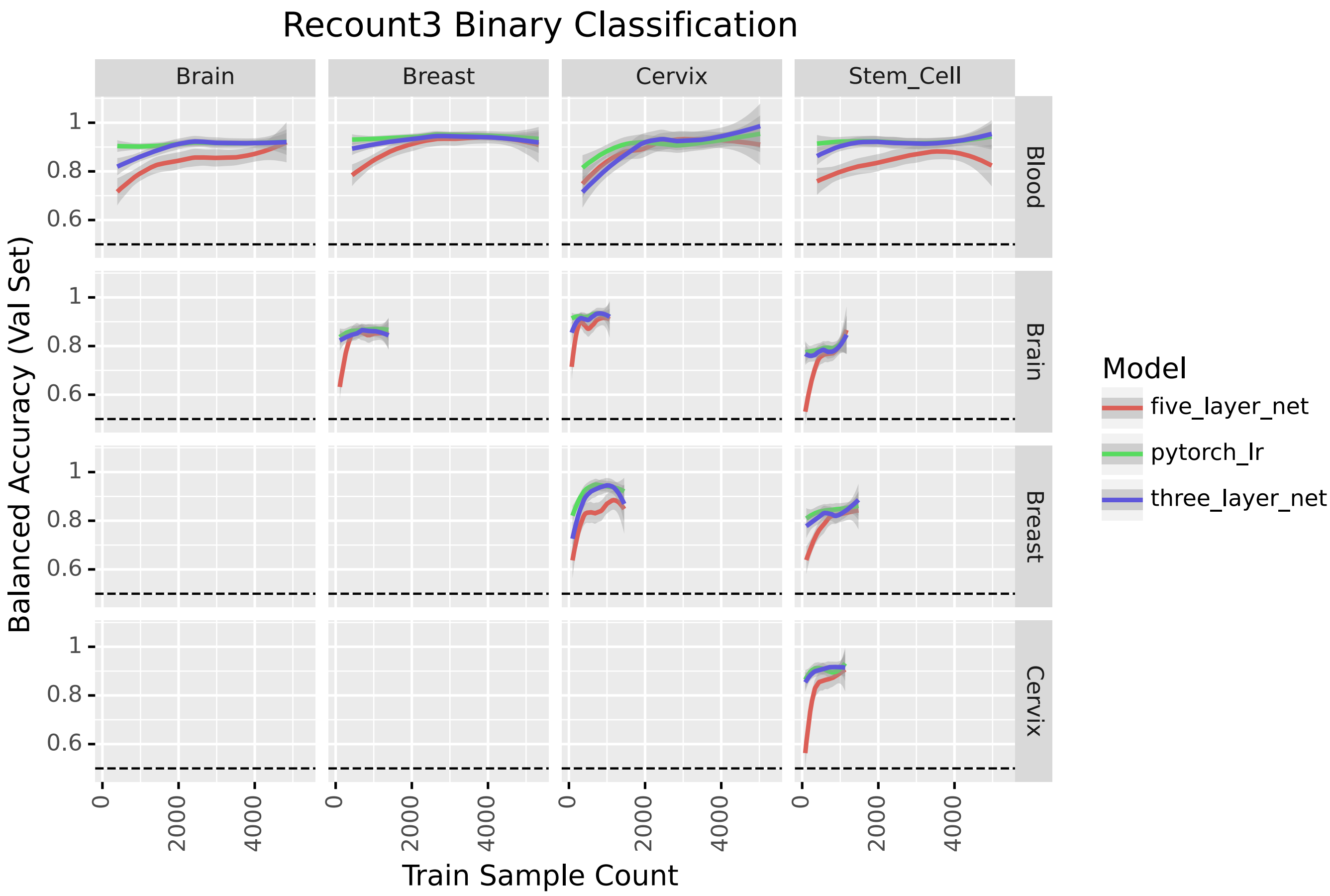

Figure 5:
Comparison of models' binary classification performance on Recount3 data

Signal removal

While it's possible to remove signal in the full dataset or the train and validation sets independently, we decided to do the latter. We made this decision because we observed potential data leakage when removing signal from the entire dataset in one go (supp. fig. 6).

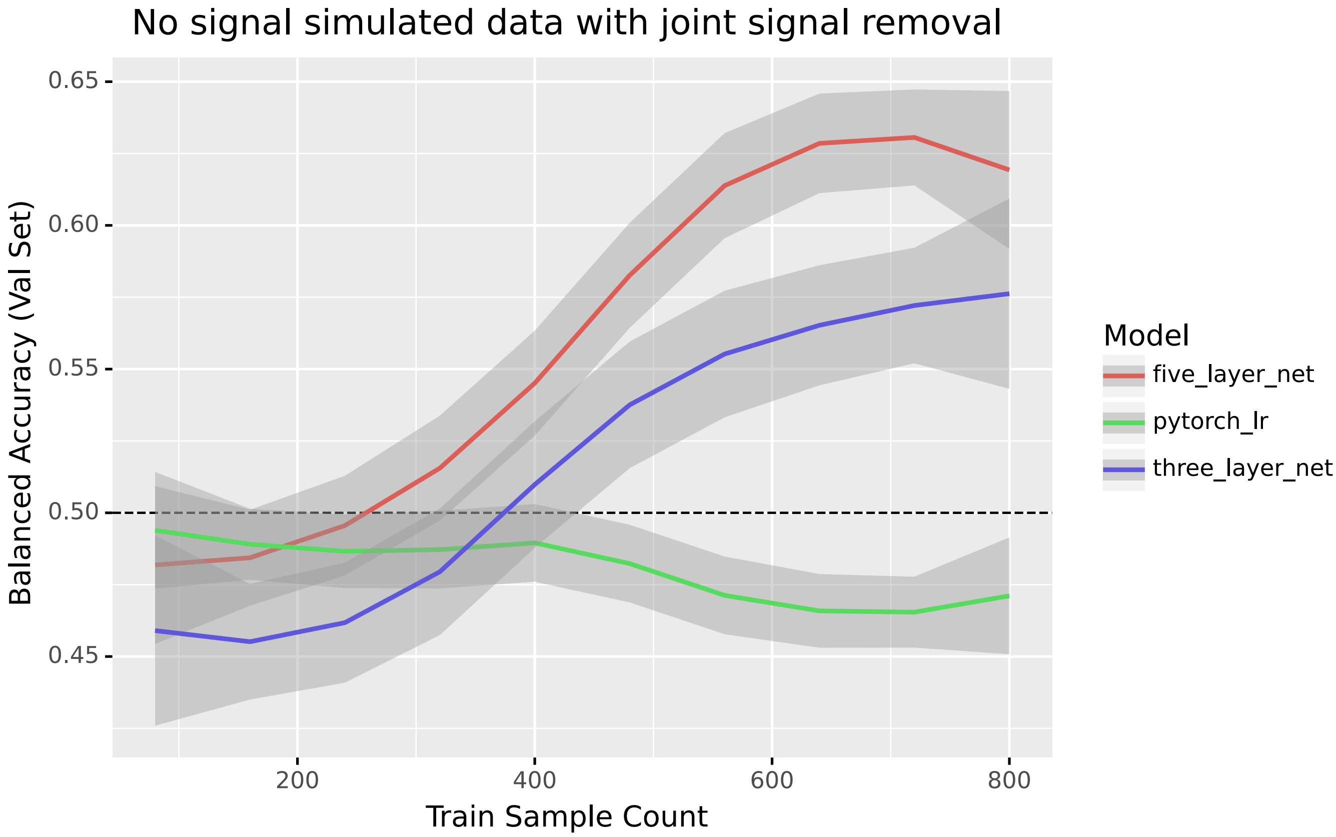

Figure 6:
Full dataset signal removal in a dataset without signal

Figure 7:
Comparison of models' binary classification performance before and after removing linear signal

Samplewise splitting

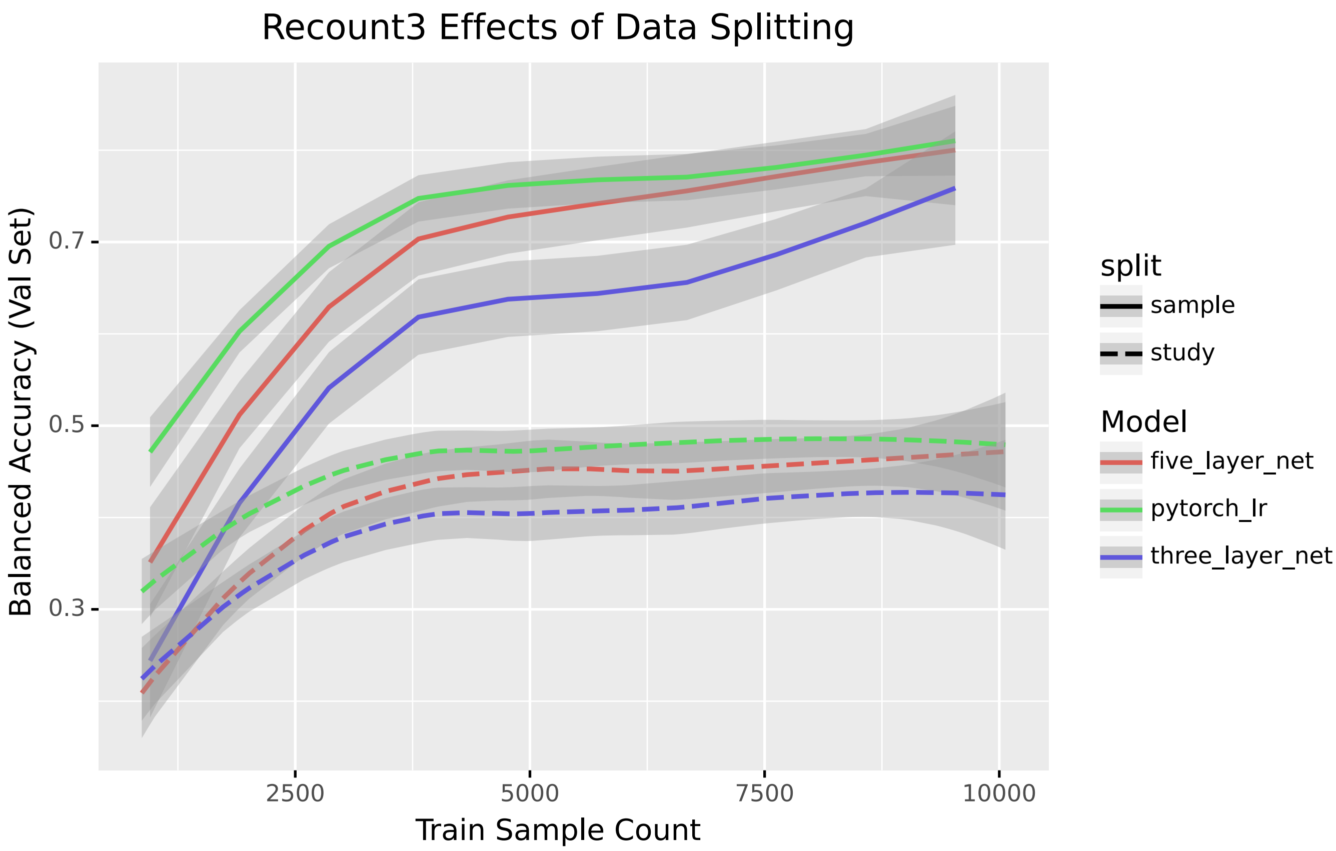

Figure 8:
Performance of Recount3 multiclass prediction with samplewise train/val splitting

Recount3 Pretraining

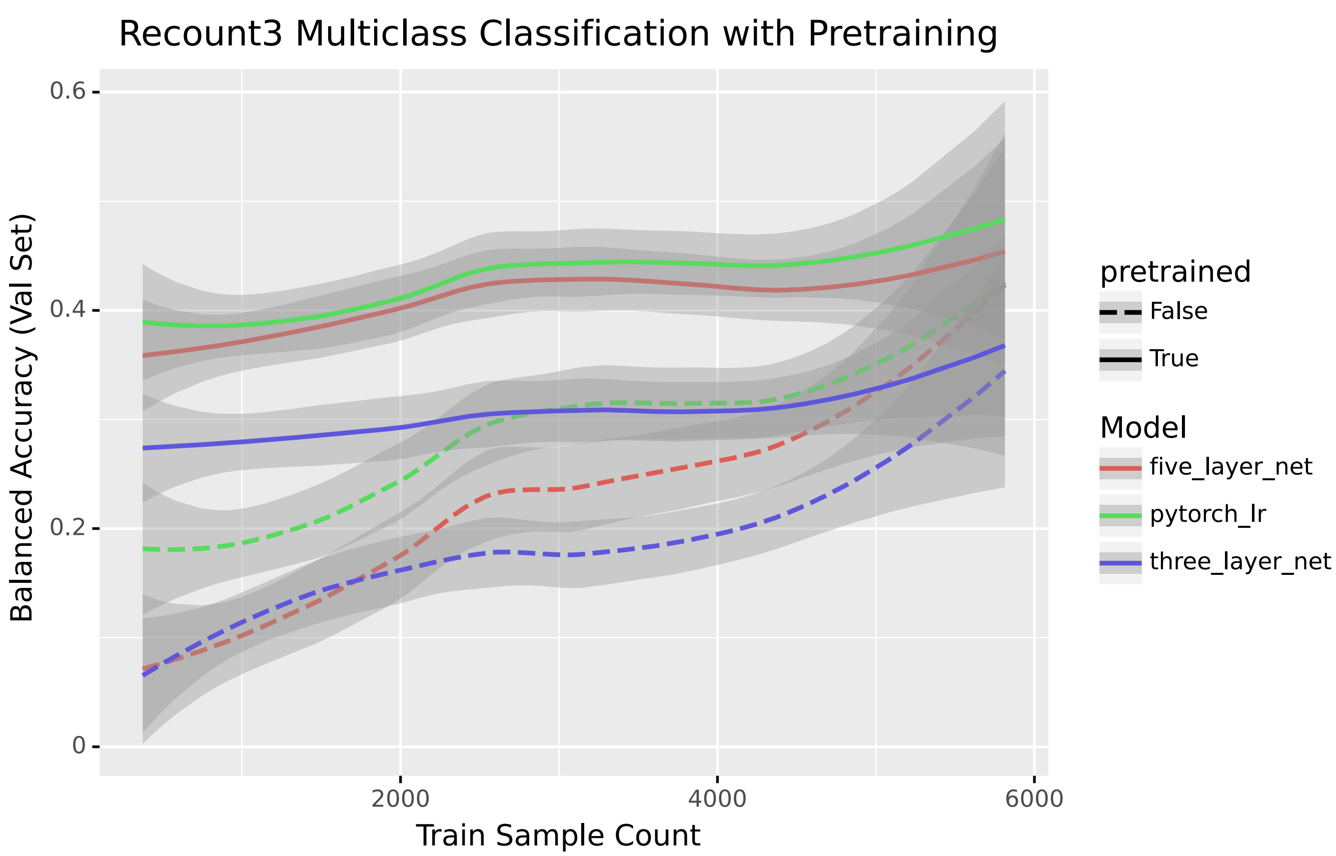

Figure 9:
Performance of Recount3 multiclass prediction with pretraining

Methods

Recount3 tissues used

The tissues used from Recount3 were blood, breast, stem cell, cervix, brain, kidney, umbilical cord, lung, epithelium, prostate, liver, heart, skin, colon, bone marrow, muscle, tonsil, blood vessel, spinal cord, testis, and placenta.

References

1. **Supervised Risk Predictor of Breast Cancer Based on Intrinsic Subtypes**
Joel S Parker, Michael Mullins, Maggie CU Cheang, Samuel Leung, David Voduc, Tammi Vickery, Sherri Davies, Christiane Fauron, Xiaping He, Zhiyuan Hu, … Philip S Bernard
Journal of Clinical Oncology (2009-03-10) https://doi.org/c2688w
DOI: 10.1200/jco.2008.18.1370 · PMID: 19204204 · PMCID: PMC2667820

2. **Gene Expression Profiling for the Identification and Classification of Antibody-Mediated Heart Rejection**
Alexandre Loupy, Jean Paul Duong Van Huyen, Luis Hidalgo, Jeff Reeve, Maud Racapé, Olivier Aubert, Jeffery M Venner, Konrad Falmuski, Marie Cécile Bories, Thibaut Beuscart, … Philip F Halloran
Circulation (2017-03-07) https://doi.org/f9vfvw
DOI: 10.1161/circulationaha.116.022907 · PMID: 28148598

3. **Large-scale labeling and assessment of sex bias in publicly available expression data**
Emily Flynn, Annie Chang, Russ B Altman
BMC bioinformatics (2021-03-30) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8011224/
DOI: 10.1186/s12859-021-04070-2 · PMID: 33784977 · PMCID: PMC8011224

4. **Compute Trends Across Three Eras of Machine Learning**
Jaime Sevilla, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, Pablo Villalobos
arXiv (2022-03-11) https://arxiv.org/abs/2202.05924

5. **Massive mining of publicly available RNA-seq data from human and mouse**
Alexander Lachmann, Denis Torre, Alexandra B Keenan, Kathleen M Jagodnik, Hoyjin J Lee, Lily Wang, Moshe C Silverstein, Avi Ma'ayan
Nature Communications (2018-04-10) https://doi.org/gc92dr
DOI: 10.1038/s41467-018-03751-6 · PMID: 29636450 · PMCID: PMC5893633

6. **A curated database reveals trends in single-cell transcriptomics**
Valentine Svensson, Eduardo da Veiga Beltrame, Lior Pachter
Database (2020) https://doi.org/gjnr3h
DOI: 10.1093/database/baaa073 · PMID: 33247933 · PMCID: PMC7698659

7. **DeePathology: Deep Multi-Task Learning for Inferring Molecular Pathology from Cancer Transcriptome**
Behrooz Azarkhalili, Ali Saberi, Hamidreza Chitsaz, Ali Sharifi-Zarchi
Scientific Reports (2019-11-11) https://doi.org/gpg7vc
DOI: 10.1038/s41598-019-52937-5 · PMID: 31712594 · PMCID: PMC6848155

8. **Bias-invariant RNA-sequencing metadata annotation**
Hannes Wartmann, Sven Heins, Karin Kloiber, Stefan Bonn
GigaScience (2021-09) https://doi.org/gph9xp
DOI: 10.1093/gigascience/giab064 · PMID: 34553213 · PMCID: PMC8559615

9. **Improved prediction of smoking status via isoform-aware RNA-seq deep learning models**
Zifeng Wang, Aria Masoomi, Zhonghui Xu, Adel Boueiz, Sool Lee, Tingting Zhao, Russell Bowler, Michael Cho, Edwin K Silverman, Craig Hersh, … Peter J Castaldi
PLOS Computational Biology (2021-10-11) https://doi.org/gph9xq
DOI: 10.1371/journal.pcbi.1009433 · PMID: 34634029 · PMCID: PMC8530282

10. **The evolution of gene expression and the transcriptome–phenotype relationship**

Peter W Harrison, Alison E Wright, Judith E Mank
Seminars in Cell & Developmental Biology (2012-04) https://doi.org/fxqd2g
DOI: 10.1016/j.semcdb.2011.12.004 · PMID: 22210502 · PMCID: PMC3378502

11. **Nonlinear Dynamics in Gene Regulation Promote Robustness and Evolvability of Gene Expression Levels**
Arno Steinacher, Declan G Bates, Ozgur E Akman, Orkun S Soyer
PLOS ONE (2016-04-15) https://doi.org/f8xrrq
DOI: 10.1371/journal.pone.0153295 · PMID: 27082741 · PMCID: PMC4833316

12. **ADAGE-Based Integration of Publicly Available Pseudomonas aeruginosa Gene Expression Data with Denoising Autoencoders Illuminates Microbe-Host Interactions**
Jie Tan, John H Hammond, Deborah A Hogan, Casey S Greene
mSystems (2016-02-23) https://doi.org/gcgmbq
DOI: 10.1128/msystems.00025-15 · PMID: 27822512 · PMCID: PMC5069748

13. **A semi-supervised deep learning method based on stacked sparse auto-encoder for cancer prediction using RNA-seq data**
Yawen Xiao, Jun Wu, Zongli Lin, Xiaodong Zhao
Computer Methods and Programs in Biomedicine (2018-11) https://doi.org/gfnm5c
DOI: 10.1016/j.cmpb.2018.10.004 · PMID: 30415723

14. **A biological network-based regularized artificial neural network model for robust phenotype prediction from gene expression data**
Tianyu Kang, Wei Ding, Luoyan Zhang, Daniel Ziemek, Kourosh Zarringhalam
BMC Bioinformatics (2017-12) https://doi.org/gf8cm6
DOI: 10.1186/s12859-017-1984-2 · PMID: 29258445 · PMCID: PMC5735940

15. **Standard machine learning approaches outperform deep representation learning on phenotype prediction from transcriptomics data**
Aaron M Smith, Jonathan R Walsh, John Long, Craig B Davis, Peter Henstock, Martin R Hodge, Mateusz Maciejewski, Xinmeng Jasmine Mu, Stephen Ra, Shanrong Zhao, … Charles K Fisher
BMC Bioinformatics (2020-03-20) https://doi.org/ggpc9d
DOI: 10.1186/s12859-020-3427-8 · PMID: 32197580 · PMCID: PMC7085143

16. **A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models**
Evangelia Christodoulou, Jie Ma, Gary S Collins, Ewout W Steyerberg, Jan Y Verbakel, Ben Van Calster
Journal of Clinical Epidemiology (2019-06) https://doi.org/gfzstd
DOI: 10.1016/j.jclinepi.2019.02.004 · PMID: 30763612

17. **The Genotype-Tissue Expression (GTEx) project**
John Lonsdale, Jeffrey Thomas, Mike Salvatore, Rebecca Phillips, Edmund Lo, Saboor Shad, Richard Hasz, Gary Walters, Fernando Garcia, Nancy Young, … Helen F Moore
Nature Genetics (2013-05-29) https://doi.org/gd5z68
DOI: 10.1038/ng.2653 · PMID: 23715323 · PMCID: PMC4010069

18. **recount3: summaries and queries for large-scale RNA-seq expression and splicing**
Christopher Wilks, Shijie C Zheng, Feng Yong Chen, Rone Charles, Brad Solomon, Jonathan P Ling, Eddie Luidy Imada, David Zhang, Lance Joseph, Jeffrey T Leek, … Ben Langmead
Genome Biology (2021-11-29) https://doi.org/gnm7zc
DOI: 10.1186/s13059-021-02533-6 · PMID: 34844637 · PMCID: PMC8628444

19. **Large-scale labeling and assessment of sex bias in publicly available expression data**

Emily Flynn, Annie Chang, Russ B Altman
BMC Bioinformatics (2021-03-30) https://doi.org/gpjt3n
DOI: 10.1186/s12859-021-04070-2 · PMID: 33784977 · PMCID: PMC8011224

20. **The Sequence Read Archive**
R Leinonen, H Sugawara, M Shumway
Nucleic Acids Research (2010-11-09) https://doi.org/c652z5
DOI: 10.1093/nar/gkq1019 · PMID: 21062823 · PMCID: PMC3013647

21. **limma powers differential expression analyses for RNA-sequencing and microarray studies**
Matthew E Ritchie, Belinda Phipson, Di Wu, Yifang Hu, Charity W Law, Wei Shi, Gordon K Smyth
Nucleic Acids Research (2015-01-20) https://doi.org/f7c4n5
DOI: 10.1093/nar/gkv007 · PMID: 25605792 · PMCID: PMC4402510

22. **Phosphorylation of ETS transcription factor ER81 in a complex with its coactivators CREB-binding protein and p300**
S Papoutsopoulou, R Janknecht
Molecular and cellular biology (2000-10) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC86284/
DOI: 10.1128/mcb.20.19.7300-7310.2000 · PMID: 10982847 · PMCID: PMC86284

23. **BioMart--biological queries made easy**
Damian Smedley, Syed Haider, Benoit Ballester, Richard Holland, Darin London, Gudmundur Thorisson, Arek Kasprzyk
BMC genomics (2009-01-14) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2649164/
DOI: 10.1186/1471-2164-10-22 · PMID: 19144180 · PMCID: PMC2649164

24. **The European Nucleotide Archive**
Rasko Leinonen, Ruth Akhtar, Ewan Birney, Lawrence Bower, Ana Cerdeno-Tárraga, Ying Cheng, Iain Cleland, Nadeem Faruque, Neil Goodgame, Richard Gibson, … Guy Cochrane
Nucleic acids research (2011-01) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3013801/
DOI: 10.1093/nar/gkq967 · PMID: 20972220 · PMCID: PMC3013801

25. **PyTorch: An Imperative Style, High-Performance Deep Learning Library**
Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, … Soumith Chintala
arXiv (2019-12-05) https://arxiv.org/abs/1912.01703

26. **Rectified linear units improve restricted boltzmann machines**
Vinod Nair, Geoffrey E Hinton
Proceedings of the 27th International Conference on International Conference on Machine Learning (2010-06-21) https://dl.acm.org/doi/10.5555/3104322.3104425
ISBN: 9781605589077

27. **Logistic regression implemented using pytorch performs worse than sklearn's logistic regression**
Tony_Wang (https://discuss.pytorch.org/u/tony_wang/summary)
(2019) https://discuss.pytorch.org/t/logistic-regression-implemented-using-pytorch-performs-worse-than-sklearns-logistic-regression/52447

28. **Adam: A Method for Stochastic Optimization**
Diederik P Kingma, Jimmy Ba
arXiv (2017-01-31) https://arxiv.org/abs/1412.6980

29. **Dropout: A Simple Way to Prevent Neural Networks from Overfitting**
Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov

Journal of Machine Learning Research (2014) http://jmlr.org/papers/v15/srivastava14a.html

30. **Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift**
Sergey Ioffe, Christian Szegedy
Proceedings of the 32nd International Conference on Machine Learning (2015-06-01)
https://proceedings.mlr.press/v37/ioffe15.html

31. **Neptune: Experiment management and collaboration tool**
neptune.ai
(2020) https://neptune.ai

32. **Snakemake--a scalable bioinformatics workflow engine**
J Koster, S Rahmann
Bioinformatics (2012-08-20) https://doi.org/gd2xzq
DOI: 10.1093/bioinformatics/bts480 · PMID: 22908215