A scalable approach to investigating sequence-to-expression prediction from personal genomes

Anna E. Spiro[1,*], Xinming Tu[1,*], Yilun Sheng[1], Alexander Sasse[1], Rezwan Hosseini[2], Maria Chikina[2], Sara Mostafavi[1,$]

[1]Paul G. Allen School of Computer Science and Engineering, University of Washington, WA, USA, 98195
[2]Department of Computational and Systems Biology, University of Pittsburgh, Pittsburgh, PA, USA, 15260
*These authors contributed equally
[$]To whom correspondence should be addressed to: saramos@cs.washington.edu

Abstract

A key promise of sequence-to-function (S2F) models is their ability to evaluate arbitrary sequence inputs, providing a robust framework for understanding genotype-phenotype relationships. However, despite strong performance across genomic loci , S2F models struggle with inter-individual variation. Training a model to make genotype-dependent predictions at a single locus—an approach we call personal genome training—offers a potential solution. We introduce SAGE-net, a scalable framework and software package for training and evaluating S2F models using personal genomes. Leveraging its scalability, we conduct extensive experiments on model and training hyperparameters, demonstrating that training on personal genomes improves predictions for held-out individuals. However, the model achieves this by identifying predictive variants rather than learning a *cis*-regulatory grammar that generalizes across loci. This failure to generalize persists across a range of hyperparameter settings. These findings highlight the need for further exploration to unlock the full potential of S2F models in decoding the regulatory grammar of personal genomes. Scalable software and infrastructure development will be critical to this progress.

Main

Sequence-to-function (S2F) deep neural networks have recently emerged as powerful tools for predicting how genomic DNA maps to cell-type-specific regulatory functions (1–7). Unlike statistical genetic models that rely on large cohorts to associate observed genetic variation with gene expression, S2F models take a mechanistic approach to learning gene regulation. This allows them to predict the impact of unseen genetic variants while also providing insights into the regulatory grammar encoded in genomic DNA that drives their predictions. By bridging

prediction and explanation, S2F models offer a coherent framework for interpreting genetic variants in personal genomes.

However, recent studies, including our own, have revealed that current S2F models trained on a single reference genome (which we refer to as reference-S2F) struggle to accurately predict gene expression from personal genomes (8,9). This observation suggests that while reference-S2F models capture certain aspects of *cis*-regulatory grammar that drive gene expression differences across genes, their ability to grasp the finer rules governing inter-individual variation in gene expression remains limited. Closing this gap requires higher-resolution training data, and a compelling approach is to train or fine-tune S2F models on cohort datasets with paired whole-genome sequencing (WGS) and RNA-seq, allowing them to learn how subtle genetic variation impacts gene expression (10–12).

Initial efforts have begun to explore this direction, but a significant bottleneck remains in computational scalability, which impedes rapid cycles of experimentation and result analysis. State-of-the-art models, such as Borzoi (5), are very large convolutional neural networks (CNNs) with >100M parameters. Training these models on a single reference genome requires weeks of GPU time. Even though fine-tuning on each additional genome can take much less time, the time still quickly becomes prohibitive (fine-tuning on 1000 genes across a large cohort of individuals takes weeks of GPU time (11)). This limitation has constrained current explorations, for example, restricting analyses to a small and potentially biased subset of genomic regions (e.g., genes selected based on linear heritability) (10).

Here, we developed a scalable model and software framework for training sequence-to-expression (S2E) deep neural networks using personal genomes, which we call SAGE-net (small and good enough CNN). Our approach addresses important gaps in existing software by incorporating key components: 1) an "on-the-fly" personal dataset that efficiently converts personal genomes into one-hot-encoded matrices for S2F training; 2) a contrastive learning model architecture that enable models to learn a refined *cis*-regulatory grammar predictive of subtle variation in gene expression across individuals; and 3) the use of compact CNNs to predict gene expression, which deliver performance comparable to fine-tuning large S2F models like Enformer (4), but with significantly lower computational cost.

At its core, SAGE-net builds on a convolutional neural network (CNN) that uses multiple rounds of convolutional layers followed by pooling to make predictions for long input sequences. Here, we explored a model that takes as input 40 kb of genomic DNA centered at gene transcription start sites (TSSs), although we note that the software package is flexible to model input length and various architecture choices. First, we compared the performance of this reference-only-trained CNN (r-SAGE-net) with that of Enformer in predicting mean (i.e., averaged across individuals) cortex gene expression from unseen genomic locations (**Fig. S1a,b**), using RNA-seq data from the ROSMAP cohort (13). While both models performed well,

Enformer clearly outperformed r-SAGE-net (we note that the predictions of Enformer were fine-tuned to gene expression in cortex, and without this fine-tunning its performance is worse than r-SAGE-net, see **Fig. S1c**, Methods). Nevertheless, r-SAGE-net achieved strong performance while reducing inference time by 70-fold (**Fig. S1d**).

Encouraged by this result, we created personal-SAGE-net (p-SAGE-net), which uses a CNN in a contrastive learning framework to decouple the "mean" and the "personal" components of gene expression per locus (**Fig. 1a**). Following previous work (10,11), we trained and tested our model on personal genomes (ROSMAP n=859 individuals, GTEx (14) n=205 individuals; **Fig. 1b**) using the top 1000 genes ranked by linear estimate of heritability (as determined by PrediXcan (15)). P-SAGE-net achieved performance comparable to PrediXcan when evaluated on unseen alleles for the *training* genes (ROSMAP n=85, GTEx n=205 individuals per 734 training genes; **Fig. 1c**)—a modest but critical benchmark for evaluating model effectiveness. This performance is consistent when the model is tested on a separate cohort (training on ROSMAP and predicting on GTEx; **Fig. 1c**). We note that p-SAGE-net achieves similar performance to models that fine-tune Borzoi or Enformer (10,11).

We then examined whether p-SAGE-net learns additional *cis*-regulatory grammar to more accurately predict gene expression variation across individuals compared to reference-S2F models. Using in-silico mutagenesis (ISM), we identified cases where it captures regulatory grammar underlying inter-individual expression differences—patterns often missed by reference-S2F models. For instance, for the *GSTM3* gene, reference-S2F models incorrectly predict expression changes with a negative correlation, whereas the personal counterpart model produces accurate, positively correlated predictions (**Fig. S2**). ISM analysis for p-SAGE-net near the Susie-fine-mapped causal variant (16) reveals a repressive motif, matching the transcription factor HLF, that is disrupted by the variant (**Fig. S3**)—a pattern absent in both r-SAGE-net (**Fig. S4**) and Enformer (**Fig. S5**). Beyond analyzing specific examples, we also systematically compared seqlets—short sequence motifs driving model predictions—between p-SAGE-net and its reference-only counterpart (see Methods). Across 466 well-predicted genes, seqlets from p-SAGE-net are distributed farther from TSSs compared to those from r-SAGE-net (**Fig. S6**). This suggests that training on personal genomes helps mitigate models' previously observed under-reliance on distal variants (17). A similar shift in attention to distal variants with cross-individual training was observed in previous studies using a variant-based rather than seqlet-based attribution analysis (10,11).

We next evaluated p-SAGE-net's ability to predict unseen alleles from unseen genes. This represents an important but significantly more challenging evaluation, as it requires the model's learned *cis*-regulatory grammar to generalize across both loci and individuals. However, as also previously shown for other personal-S2F models (10,11), we observed that p-SAGE-net (fine-tuned or trained from scratch, see **Fig. S7d**) does not generalize to alleles in unseen genes

(**Fig. 1d**). Moreover, we noticed that p-SAGE-net's performance on predicting differences between genes is worse than r-SAGE-net's (**Fig. S8**), suggesting that the model may be "forgetting" some generalizable cross-gene *cis*-regulatory grammar as it trains on personal genomes. To understand this further, we examined the ability of models to predict mean gene expression with each fine-tuning epoch. Interestingly, we observed that as training epochs on personal genomes increased, the performance of p-SAGE-net in predicting mean gene expression for unseen genes progressively declined (**Fig. 2a**), likely showing the model's tendency to overfit to training genes rather than learn a coherent regulatory grammar that generalizes across loci and individuals.

We performed an ablation analysis to understand the relationship between the model's architecture and these observations above. We observed that pre-training by initializing weights from r-SAGE-net (as opposed to training p-SAGE-net from scratch) only modestly improves cross-individual predictions (**Fig. S7a)**. Consistent with previous work (10,11), we see that using a loss function that emphasizes individual differences within genes is important for model performance across individuals (**Fig. S7a**). We also explored key modifications to the loss function (**Fig. S7b,e**), input sequence length (**Fig. S7c,f)**, and inclusion/exclusion of transformer blocks (**Fig. S7a,d)**. In short, while some of these variations affected seen gene per-gene performance, none allowed the model to generalize to unseen alleles in unseen genes.

Our flexible on-the-fly personal dataset, which constructs sequences directly from VCF files given a specified minor allele frequency (MAF) threshold, readily enables examining model performance on variants filtered by MAF. We assessed two gene sets: the top 1000 genes and the top 4000-5000 genes (as ranked by PrediXcan). We included this lower-ranked gene set out of concern that when heritability is estimated using linear, population-level statistical models, genes appearing to have higher heritability may actually be those primarily driven by a single common causal variant with a large effect. When we set MAF >0.05 (only including variants in model evaluation that are present at least 69 times in the 689 training individuals), model performance does not decrease for the top 1000 gene set, but starts to decrease slightly for the 4000-5000 gene set (**Fig. 2b**). This suggests that low frequency variants (MAF<0.05) are not major contributors to model predictions for the most heritable genes, and more generally that focusing on gene sets chosen by statistical heritability analysis presents its own biases.

We also examined training sample size requirements using the same two gene sets. While model performance for the top 1000 gene set plateaued after ~400 training individuals, the 4000-5000 gene set exhibited a more gradual performance increase, suggesting that these genes require larger datasets to capture meaningful regulatory patterns (**Fig. 2c**). This aligns with our intuition that "easier" genes—those whose expression can be well-predicted from just a few variants—require fewer training samples, whereas "harder" genes demand more data to uncover the larger set of weaker factors driving inter-individual variation.

Finally, we hypothesized that fine-tuning p-SAGE-net on many genes (i.e., >1k) should provide additional training data for learning generalizable regulatory grammar, leading to improved performance as the number of training genes increases. However, when we add more training genes—either randomly ("random genes") or ranked by decreasing PrediXcan performance ("top genes")—we observe a drop in performance for the core smallest set of genes (**Fig. 2d**). This drop in performance remains even if we increase model capacity in intermediate model layers (**Fig. S9**). This decline suggests that rather than learning a generalizable regulatory grammar, the model may be merely identifying predictive variants from the training data.

In summary, our findings reveal critical limitations that are broadly relevant to current personal-S2E modeling efforts. Specifically, p-SAGE-net's inability to generalize to unseen genes and its reliance on common variants highlight the need for further refinement in capturing the full complexity of *cis*-regulatory grammar across individuals. The scalability of p-SAGE-net has enabled us to systematically explore training gene set biases and sample size requirements, underscoring the importance of expanding training datasets and improving model architectures to better capture the nuanced impact of genetic variation on gene expression.

The availability of scalable software like p-SAGE-net provides a robust foundation for advancing personalized deep genomics through the exploration of diverse training strategies. For instance, pre-training on random DNA sequences, which are not constrained by natural selection, may offer a promising avenue for uncovering broader regulatory principles (18). Additionally, integrating additional data modalities paired with WGS and RNA-seq—such as those capturing post-transcriptional regulatory processes—could provide a more comprehensive understanding of inter-individual expression differences. Moreover, we hypothesize that active learning strategies, where training is done on a varied number of individuals per gene but considers a larger set of genes, may mitigate model overfitting to training genes. These directions, combined with continued innovation in model design, will be essential for fully realizing the potential of sequence-to-function models in interpreting the regulatory code of personal genomes.

Methods

WGS and RNA-seq datasets

ROSMAP*:*

We use n=859 individuals with available WGS and RNA-seq (dorsolateral prefrontal cortex) from the ROS and MAP cohort studies (13), as in our previous work (9). We take log(TPM+1) and preprocess by regressing out known covariates and top expression principal components (PCs), following the process recommended by recent work (19). We regress out top 10 expression PCs as well as the following set of technical and phenotype covariates (as done in previous work (9): batch, estimated library size, PF reads aligned, percent coding bases, percent intergenic bases,

percent PF reads aligned, percent ribosomal bases, percent UTR bases, percent duplication, median 3prime bias, median 5prime to 3prime bias, median CV coverage, RIN score, post mortem interval, age of death, sex, and top 3 genotype PCs (from PLINK (20)). We retain gene means.

As in previous work (9), we use phased WGS data. The variant call files for WGS data from the ROSMAP in variant call format (VCF) were obtained from the Synapse repository (syn117074200). The coordinates of variant calls (GRCh37) were converted to GRCh38 coordinates using the Picard LiftoverVcf tool (http://broadinstitute.github.io/picard). The Eagle software2 version 2.4.1 was used to phase the genotypes with the default setting.

GTEx*:*

We use n=205 individuals with available WGS and RNA-seq (cortex), GTEx V8. As with ROSMAP, we take log(TPM+1) (from GTEx_Analysis_2017-06-05_v8_RNASeQCv1.1.9_gene_tpm.gct) and regress out top 10 expression PCs, known technical and phenotype covariates (from GTEx_Analysis_v8_Annotations_SampleAttributesDS.txt), and top 3 genotype PCs.

We use phased WGS data (from GTEx_Analysis_2017-06-05_v8_WholeGenomeSeq_838Indiv_Analysis_Freeze.SHAPEIT2_phased.vcf.gz).

Gene sets

From all genes in the ROSMAP expression data, we filter using annotations from Gencode Release 27 (GRCh38.p10). We filter by “gene_type” = “protein_coding”, “feature” = “gene”, and “seqname” in chr{1…22} or X, Y, yielding 19805 unique ENSGs. We use TSS definitions from Gencode Release 27. We use the chromosome split:

train chromosomes = range(1, 17), validation chromosomes = [17, 18, 21, 22], test chromosomes = [19,20], which allocates 14786 genes (78%) to train, 2126 (11%) to validation, and 2008 (11%) to test.

Enformer gene sets

We downloaded Enformer’s human gene splits from: https://console.cloud.google.com/storage/browser/basenji_barnyard/data. We consider a gene to be in Enformer’s “train”, “validation”, or “test” set only if our entire input window (TSS-20000 to TSS+20000) falls within an Enformer “train”, “validation”, or “test” window. This yields 15335 train genes, 1252 validation genes, and 1516 test genes.

Individual sets

ROSMAP individuals are split randomly into train (n=689), validation (n=85), and test (n=85) sets to achieve an 80%/10%/10% split. GTEx individuals (n=205) are only used in model evaluation, not in training.

PrediXcan

We train one PrediXcan model (implemented using scikit-learn ElasticNet) for each of the 19805 genes on 689 ROSMAP training individuals. We set the input window to be 40 kb, centered on gene TSS, as with other models. We set l1_ratio (which determines the balance between L1 and L2 penalties within the elastic net penalty) to 0.5 and use ROSMAP validation individuals (n=85) to select alpha (the weight on the elastic net penalty) for each gene using ElasticNetCV. We use Pearson correlation for ROSMAP validation individuals to create the ranked gene list used throughout our analyses. We use a MAF threshold of 0.01.

Sequence inputs

To construct sequence inputs to the S2F models, we use the package pysam to insert all variants (or, those with MAF above a specified threshold) into reference sequence centered on gene TSS. Our personal dataset defaults to inserting all variants (SNPs and indels), but this can be specified by the user. For r-SAGE-net and p-SAGE-net we use a 40 kb input window, for Enformer we use Enformer's full 196,608 bp input. If a gene is on the negative strand, we take the reverse-complement after one-hot encoding. Including this reverse-complementing in model training and evaluation improved model performance on predicting mean expression but not on predicting differences across individuals (**Fig. S10**).

R-SAGE-net training

We use Pytorch for all S2F model training. R-SAGE-net is trained on reference sequence and mean expression (averaged across ROSMAP training individuals) for n=14786 training genes.
For the hyperparameter tuning process, we grid search over:

- first_layer_kernel_number=(256, **900**): input channels for the first convolutional layer
- int_layers_kernel_number=(**256**, 512, 900): input & output channels for all convolutional layers after the first
- first_layer_kernel_size=(25, **10**): kernel size for the first convolutional layer
- int_layers_kernel_size=(**5**): kernel size for all convolutional layers after the first
- n_conv_blocks=(**5**, 8): number of convolutional blocks (convolutional layer, activation) with dilation=1
- pooling_size=(25, **10**, 5): pooling kernel size
- pooling_type=("max", **"avg"**): pooling kernel type
- n_dilated_conv_blocks=(**0**, 3, 5): number of convolutional blocks with dilation!=1
- dropout=(**0**, 0.2): dropout in fully connected layers
- h_layers=(2, **1**): number of hidden layers
- increasing_dilation=(**False**, True): whether to exponentially increase dilation in dilated_conv_layers (or keep at 2)

- batch_norm=(**True**, False): whether to add batch normalization at the beginning of each convolutional block
- hidden_size=(**256**, 512, 900): number of nodes in fully connected layers
- learning_rate=(**5e-4**, 1e-3): training learning rate

Our selection of hyperparameters is bolded above.

We train for a maximum of 100 epochs, with early stopping (patience=10) based validation gene (n=2126) Pearson correlation. We use Adam optimizer with weight decay = 1e-5, learning rate scheduler "cycle" with base_lr=learning_rate/2, max_lr=learning_rate*2,cycle_momentum=False. We use gradient clipping with gradient_clip_val=1, ReLU activation functions, and batch size of 16.

P-SAGE-net training

P-SAGE-net is trained on ROSAMP training individuals and genes from the specified gene set that are in training chromosomes. It takes as input reference and personal sequence (phased to produce two haplotypes) and predicts two outputs: mean expression (across ROSMAP training individuals) and individual gene expression z-score (with respect to training individuals). All model hyperparameters are the same as for r-SAGE-net. In addition to the convolutional and pooling layers of r-SAGE-net, p-SAGE-net also has one internal fully connected layer that the flattened personal and reference tensor pass through and separate output heads of fully connected layers to predict the two outputs. Each epoch entails training on all training genes and training individuals and validating on all validation genes and validation individuals. When p-SAGE-net is initialized from r-SAGE-net, all parameters in the convolutional and pooling layers of r-SAGE-net are loaded into p-SAGE-net. We train for a maximum of 25 epochs for the "best" p-SAGE-net model shown in Figure 1 and 10 epochs for all other model variations, selecting best epoch based on train gene per-gene Pearson correlation. We note that test gene per-gene Pearson correlation remains centered at 0 even if model epoch is instead chosen based on validation gene per-gene Pearson correlation. We use batch size of 6.

P-SAGE-net ablation experiments

For "non-zscore", instead of predicting individual z-score, we instead predict difference between mean expression (across training individuals) and the individual's gene expression. For "transformer block", the last convolutional block (at the lowest resolution) is replaced with a transformer block (with n_filters=kernel_number, nhead=kernel_size). For "non contrastive", instead of splitting personal gene expression prediction into mean and difference, the model is trained to predict a single output (an individual's gene expression) from a single sequence input (the individual's personal genome sequence). In practice, we do this using the same p-SAGE-net model architecture, by using a parameter that controls not splitting personal gene expression

and not subtracting the intermediate personal and reference tensors, and by setting the hyperparameter weight on the "mean expression" portion of the loss to zero.

Predicting gene expression with Enformer

We use GTEx mean gene expression to assign weights to each of Enformer's 5313 human output tracks. To go from Enformer's outputs to fine-tuned predictions, we use these weights to transform the log(prediction + 1) of Enformer's 3 center bins (447,448,449). We then sum these three center bins to get the final fine-tuned prediction. We use these fine-tuned predictions for all analyses using Enformer.

For the non-fine-tuned version shown in Fig. S1c, we take log(prediction + 1) of Enformer's 3 center bins (447,448,449) for this track and sum.

We load the Enformer model from https://tfhub.dev/deepmind/enformer/1 and use model.predict_on_batch to make predictions.

Model attributions

To identify seqlets in our model attributions, we use the tangermeme recursive_seqlets simplification of TF-MoDISco (21) with additional_flanks=2. We use the tangermeme annotate_seqlets implementation of TOMTOM (22) to match motifs to the HOCOMOCO v12 database (23).

To perform ISM with p-SAGE-net, we mutate both haplotypes of the personal sequence and record the difference output of the model for each mutated base.

We zero-center both ISM and gradients by subtracting the mean at each position (24).

Code availability

Code to reproduce our analyses and instructions for using the software package we describe is available at:
https://github.com/mostafavilabuw/SAGEnet.

Data availability

Genotype and RNA-seq data for the Religious Orders Study and Rush Memory and Aging Project (ROSMAP) samples are available from the Synapse AMP-AD Data Portal (accession code syn2580853) as well as the RADC Research Resource Sharing Hub at www.radc.rush.edu.

GTEx genotype and RNA-seq data are available from dbGaP (accession number phs000424.vN.pN).

References

1. Alipanahi B, Delong A, Weirauch MT, Frey BJ. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nat Biotechnol. 2015 Aug;33(8):831–8.
2. Zhou J, Troyanskaya OG. Predicting effects of noncoding variants with deep learning–based sequence model. Nat Methods. 2015 Oct;12(10):931–4.
3. Kelley DR, Snoek J, Rinn JL. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome Res. 2016 Jul;26(7):990–9.
4. Avsec Ž, Agarwal V, Visentin D, Ledsam JR, Grabska-Barwinska A, Taylor KR, et al. Effective gene expression prediction from sequence by integrating long-range interactions. Nat Methods. 2021 Oct;18(10):1196–203.
5. Linder J, Srivastava D, Yuan H, Agarwal V, Kelley DR. Predicting RNA-seq coverage from DNA sequence as a unifying model of gene regulation. Nat Genet. 2025 Jan 8;1–13.
6. Avsec Ž, Weilert M, Shrikumar A, Krueger S, Alexandari A, Dalal K, et al. Base-resolution models of transcription-factor binding reveal soft motif syntax. Nat Genet. 2021 Mar;53(3):354–66.
7. Zhou J. Sequence-based modeling of three-dimensional genome architecture from kilobase to chromosome scale. Nat Genet. 2022 May;54(5):725–34.
8. Huang C, Shuai RW, Baokar P, Chung R, Rastogi R, Kathail P, et al. Personal transcriptome variation is poorly explained by current genomic deep learning models. Nat Genet. 2023 Dec;55(12):2056–9.
9. Sasse A, Ng B, Spiro AE, Tasaki S, Bennett DA, Gaiteri C, et al. Benchmarking of deep neural networks for predicting personal gene expression from DNA sequence highlights shortcomings. Nat Genet. 2023 Dec;55(12):2060–4.
10. Rastogi R, Reddy AJ, Chung R, Ioannidis NM. Fine-tuning sequence-to-expression models on personal genome and transcriptome data [Internet]. 2024 [cited 2024 Oct 17]. Available from: http://biorxiv.org/lookup/doi/10.1101/2024.09.23.614632
11. Drusinsky S, Whalen S, Pollard KS. Deep-learning prediction of gene expression from personal genomes [Internet]. 2024 [cited 2024 Aug 2]. Available from: http://biorxiv.org/lookup/doi/10.1101/2024.07.27.605449
12. Sasse A, Chikina M, Mostafavi S. Unlocking gene regulation with sequence-to-function models. Nat Methods. 2024 Aug;21(8):1374–7.
13. De Jager PL, Ma Y, McCabe C, Xu J, Vardarajan BN, Felsky D, et al. A multi-omic atlas of the human frontal cortex for aging and Alzheimer's disease research. Sci Data. 2018 Aug 7;5(1):180142.
14. The GTEx Consortium. The GTEx Consortium atlas of genetic regulatory effects across human tissues. Science. 2020 Sep 11;369(6509):1318–30.
15. Gamazon ER, Wheeler HE, Shah KP, Mozaffari SV, Aquino-Michaels K, Carroll RJ, et al. A gene-based association method for mapping traits using reference transcriptome data. Nat Genet. 2015 Sep;47(9):1091–8.
16. Wang QS, Kelley DR, Ulirsch J, Kanai M, Sadhuka S, Cui R, et al. Leveraging supervised learning for functionally informed fine-mapping of cis-eQTLs identifies an additional 20,913 putative causal eQTLs. Nat Commun. 2021 Jun 7;12(1):3394.
17. Karollus A, Mauermeier T, Gagneur J. Current sequence-based models capture gene

expression determinants in promoters but mostly ignore distal enhancers. Genome Biol. 2023 Mar 27;24(1):56.
18. de Boer CG, Taipale J. Hold out the genome: a roadmap to solving the cis-regulatory code. Nature. 2024 Jan;625(7993):41–50.
19. Zhou HJ, Li L, Li Y, Li W, Li JJ. PCA outperforms popular hidden variable inference methods for molecular QTL mapping. Genome Biol. 2022 Oct 11;23(1):210.
20. Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MAR, Bender D, et al. PLINK: A Tool Set for Whole-Genome Association and Population-Based Linkage Analyses. Am J Hum Genet. 2007 Jul 25;81(3):559.
21. Shrikumar A, Tian K, Avsec Ž, Shcherbina A, Banerjee A, Sharmin M, et al. Technical Note on Transcription Factor Motif Discovery from Importance Scores (TF-MoDISco) version 0.5.6.5 [Internet]. arXiv; 2020 [cited 2023 Oct 6]. Available from: http://arxiv.org/abs/1811.00416
22. Gupta S, Stamatoyannopoulos JA, Bailey TL, Noble WS. Quantifying similarity between motifs. Genome Biol. 2007 Feb 26;8(2):R24.
23. Vorontsov IE, Eliseeva IA, Zinkevich A, Nikonov M, Abramov S, Boytsov A, et al. HOCOMOCO in 2024: a rebuild of the curated collection of binding models for human and mouse transcription factors. Nucleic Acids Res. 2024 Jan 5;52(D1):D154–63.
24. Sasse A, Chikina M, Mostafavi S. Quick and effective approximation of in silico saturation mutagenesis experiments with first-order taylor expansion. iScience [Internet]. 2024 Sep 20 [cited 2025 Feb 20];27(9). Available from: https://www.cell.com/iscience/abstract/S2589-0042(24)02032-7

Figure 1

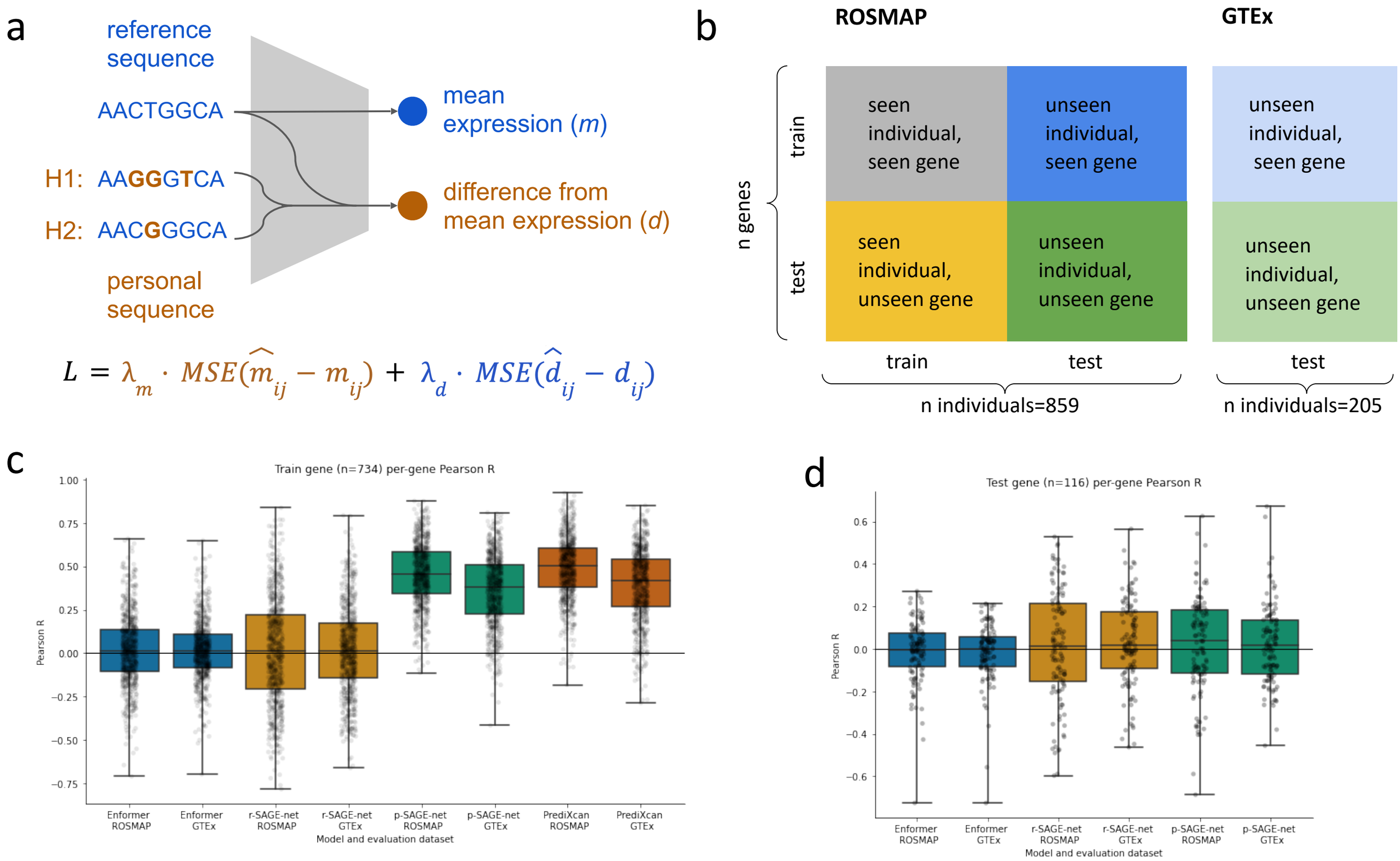


Figure 1: P-SAGE-net model and its performance on personal gene expression prediction.
(a) Schematic of p-SAGE-net. For a given data point (one gene, one individual), the model takes as input three 40 kb one-hot-encoded sequences of genomic DNA centered at gene TSS: reference sequence and personal sequences from the individual's two haplotypes (in practice, we use phased WGS data). The model uses reference sequence to predict mean gene expression (measured as the average log-transformed gene expression across individuals in a cohort, here ROSMAP), and a combination of reference and personal sequence to predict personal difference from mean gene expression. The loss function combines the MSE loss of "mean" and the MSE loss of "difference" using a hyperparameter. (b) Data splits across genes and individuals for the two datasets used, ROSMAP and GTEx. For clarity, the validation split is not shown in the figure. (c) P-SAGE-net compared to baselines on per-gene Pearson correlation for 734 seen genes (measured across 85 (ROSMAP) and 205 (GTEx) unseen individuals) from the top 1000 gene set. Each dot is one gene, boxplots show interquartile range with whiskers extending to minimum and maximum. (d) Same comparisons as in (c), but for 116 unseen genes from the same top 1000 gene set.

Figure 2

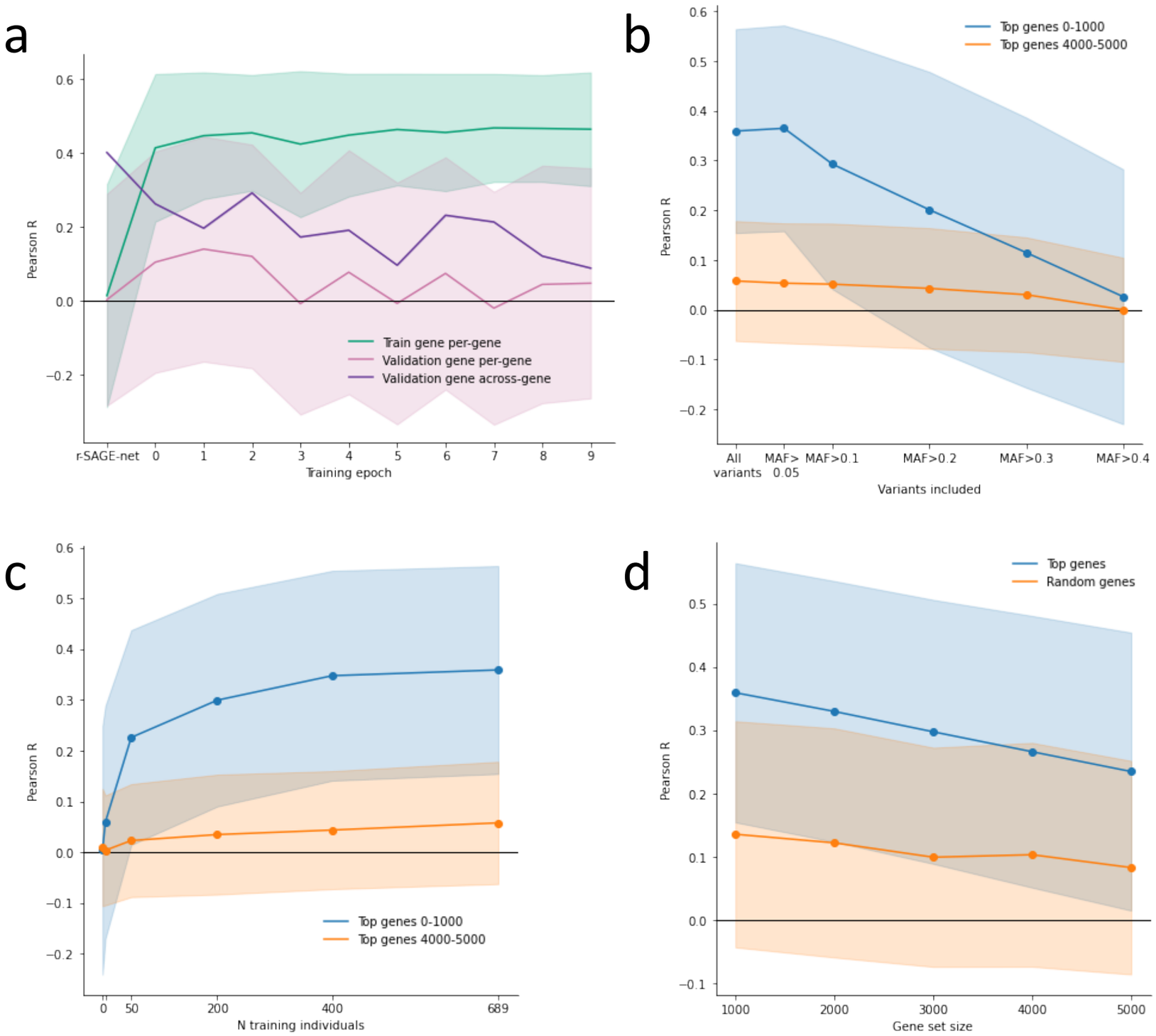
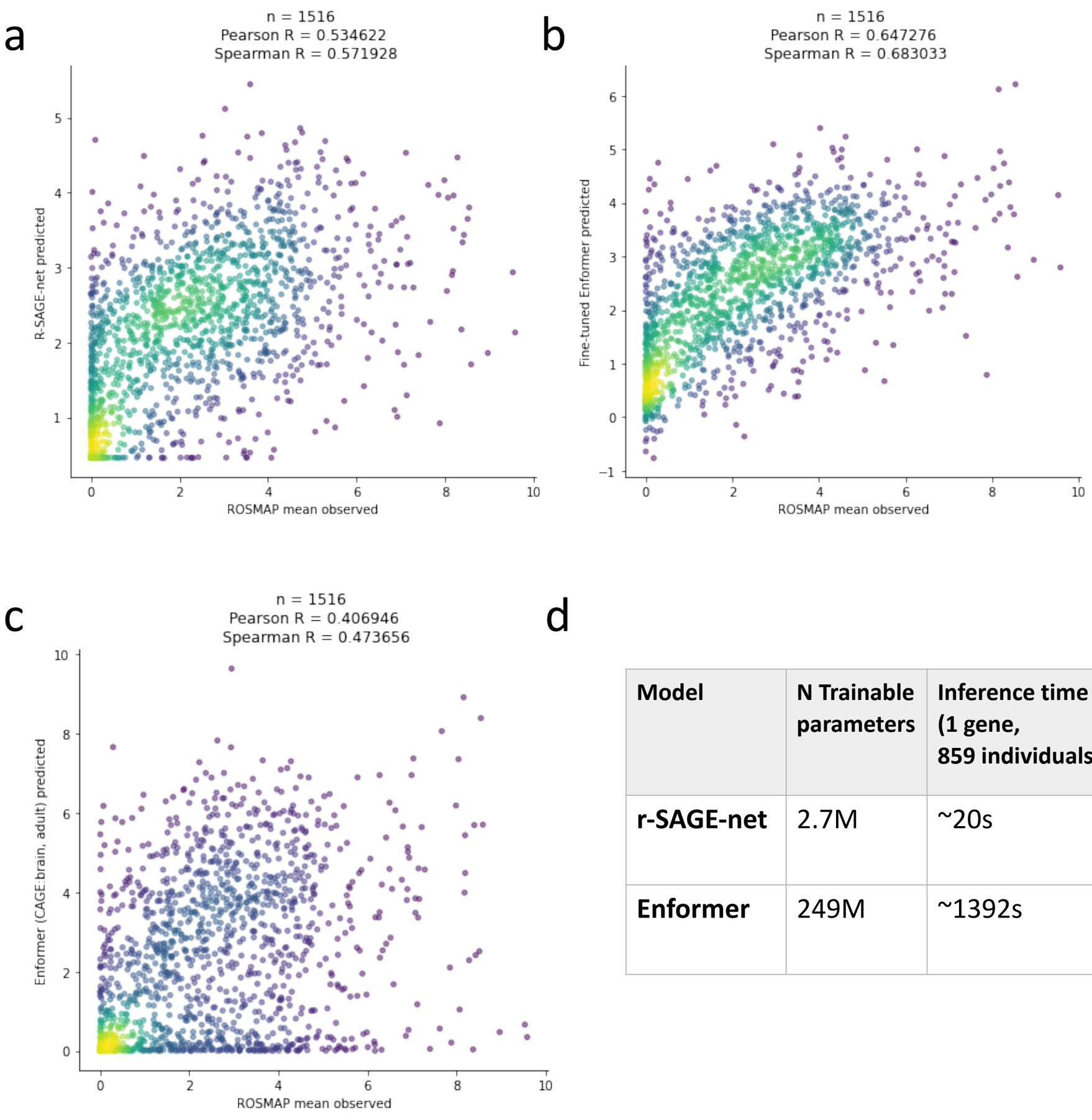


Figure 2: P-SAGE-net performance across training epoch, MAF filter, number of training individuals, and gene set size.
(a) Three different metrics (train gene per-gene Pearson R, validation gene per-gene Pearson R, validation gene across-gene Pearson R) across 10 training epochs for p-SAGE-net. Since p-SAGE-net is initialized from r-SAGE-net, r-SAGE-net metrics are shown before epoch 0. Training genes are the same 734 as in Fig. 1c, validation genes are 114 genes from the same top 1000 set. Y-axis shows median Pearson R across genes, shading shows standard deviations. (b) P-SAGE-net evaluated with MAF filters for two different gene sets. MAF is calculated using training individuals. MAF filter is used in model evaluation, not model training—the model is trained using all variants. At each MAF threshold, the model is evaluated on all 205 (unseen) GTEx individuals for each gene present in the model training set: 734 genes for the 0-1000 gene set, 756 genes for the 4000-5000 gene set. (c) P-SAGE-net trained on different numbers of training individuals for two different gene sets (the same sets as in (b)). Models are trained on different numbers of ROSMAP training individuals – 0, 5, 50, 200, 400, 689 (all) – and evaluated on 205 GTEx individuals. (d) P-SAGE-net trained on different numbers of genes, with genes added either by PrediXcan rank or randomly. For “top genes”, gene set size=1000 means that the model has been trained on all genes in the top 1000 set that are in the training chromosomes split (n=734). The gene set size is then increased up to 5000 by adding genes by descending rank, up to n=3675 training genes for the top 5000 gene set. For “random genes”, the 1000 gene set is randomly selected from the top 5000 genes and then increased by randomly adding genes within this top 5000 gene set, such that larger sets contain all genes included in smaller sets. For both “top genes” and “random genes”, models are evaluated on 205 GTEx individuals for training genes in the relevant 1000 gene set (n=734 genes for top genes, n=756 genes for random genes).

Figure S1



Model	N Trainable parameters	Inference time (1 gene, 859 individuals)
r-SAGE-net	2.7M	~20s
Enformer	249M	~1392s

Figure S1: Comparisons between r-SAGE-net and Enformer.
(a) R-SAGE-net trained on genes in Enformer's training set (n=15335) and evaluated on genes in Enformer's test set (n=1516). R-SAGE-net is trained and evaluated on 40 kb reference sequence and average log-transformed gene expression across ROSMAP individuals. Scatterplot is colored by density. (b) Fine-tuned Enformer evaluated on Enformer's test set genes. Fine-tuning follows the procedure outlined in our previous work (9) (Methods). Evaluation is done with the same length input as in Enformer training (196,608 bp). (c) Enformer evaluated on Enformer's test set genes without fine-tuning. Insead of fine-tuning, we select the representative track ("CAGE:brain, adult"), as done previously (9). (d) Comparison between r-SAGE-net and Enformer model sizes and inference times. Both models are evaluated on one NVIDIA RTX A4000 GPU.

Figure S2

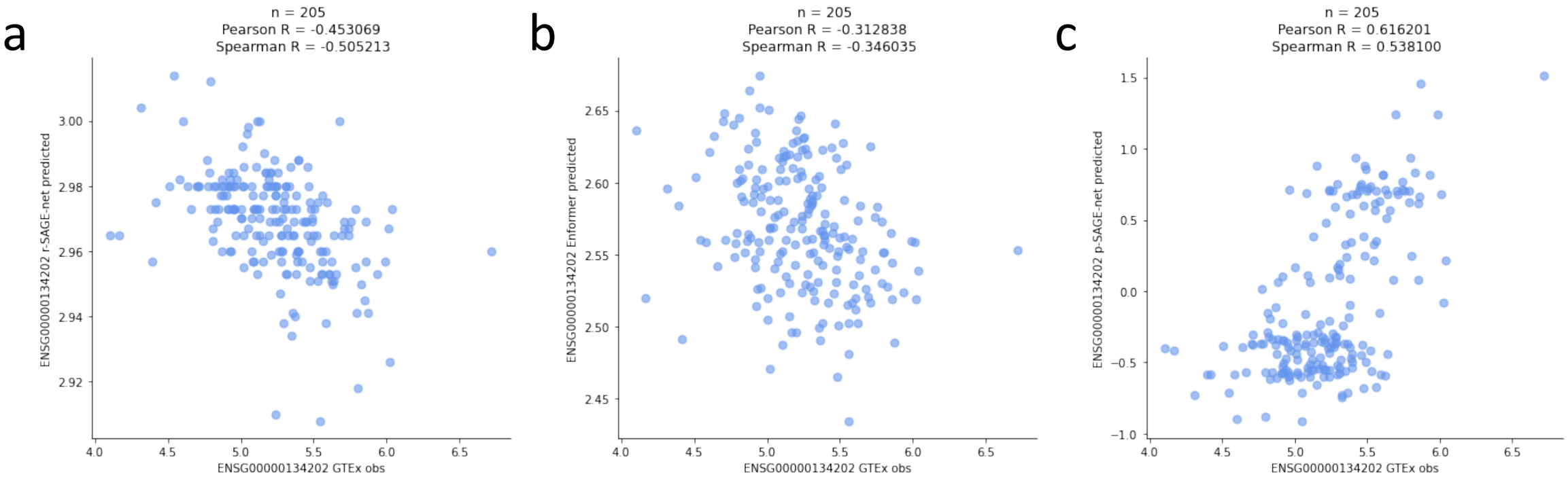


Figure S2: Model performance for GSTM3.
(a) R-SAGE-net GTEx predicted vs. observed. (b) Fine-tuned Enformer GTEx predicted vs. observed. (c) P-SAGE-net GTEx predicted vs. observed. Note that the p-SAGE-net model used here was trained on the top 1000 gene set, which includes GSTM3 (but the GTEx individuals shown here were not seen in model training).

Figure S3

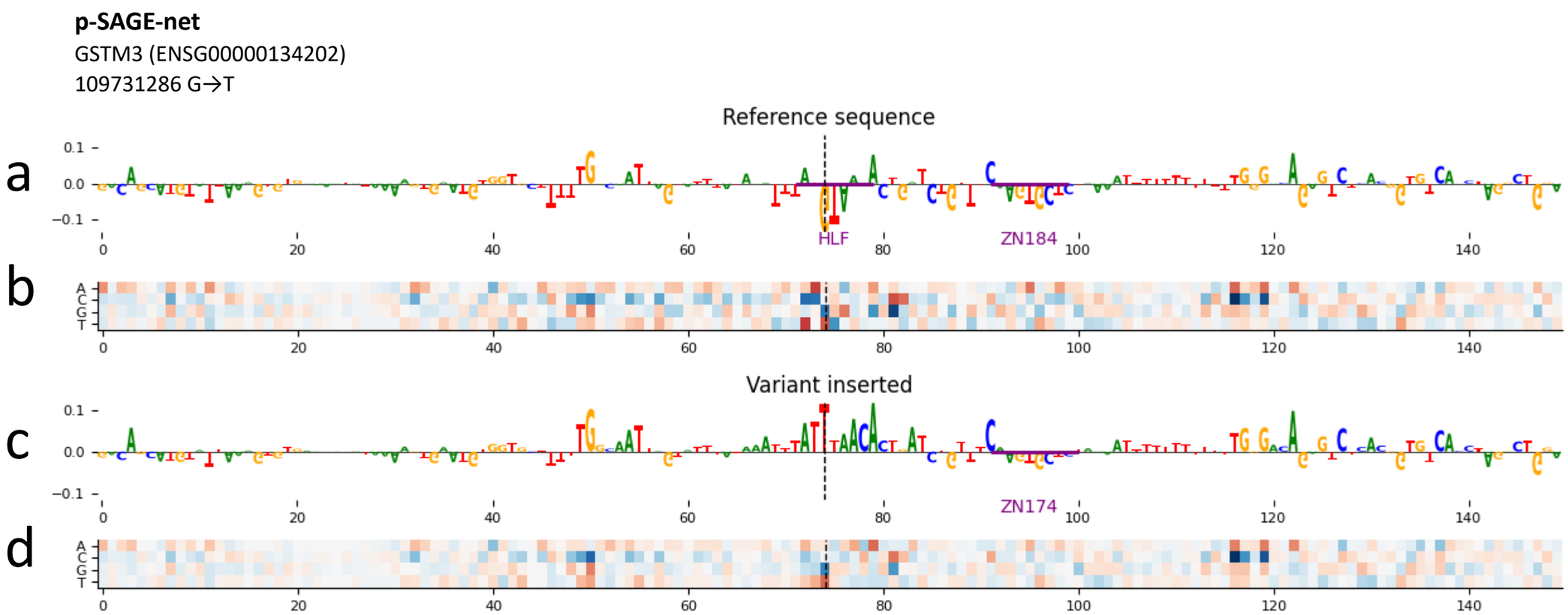


Figure S3: P-SAGE-net ISM for GSTM3.
(a) Zero-centered ISM for a 150 bp window around the variant of interest (chromosome 1, hg38 position 109731286 G → T), shown for the reference sequence. Motifs shown are from seqlets matched to the HOCOMOCO v12 database (19) by TOMTOM (20) with p<0.05. (b) Same ISM as in (a), but shown for all bases instead of only reference sequence. White represents value=0. (c) Same ISM as in (a), but done for reference sequence with variant inserted (T instead of G at 109731286). (d) Same ISM as in (c), but shown for all bases instead of only reference sequence.

Figure S4

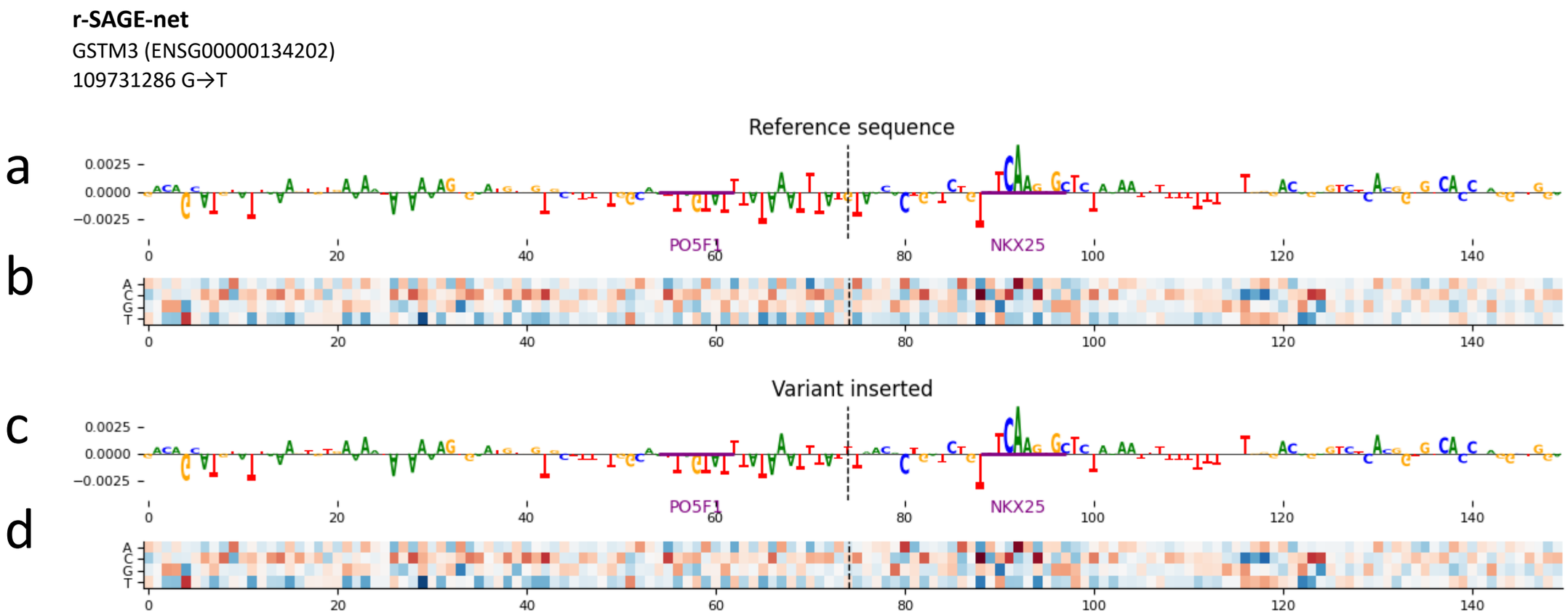


Figure S4: R-SAGE-net ISM for GSTM3. Same analysis as Fig. S3, but using r-SAGE-net instead of p-SAGE-net.

Figure S5

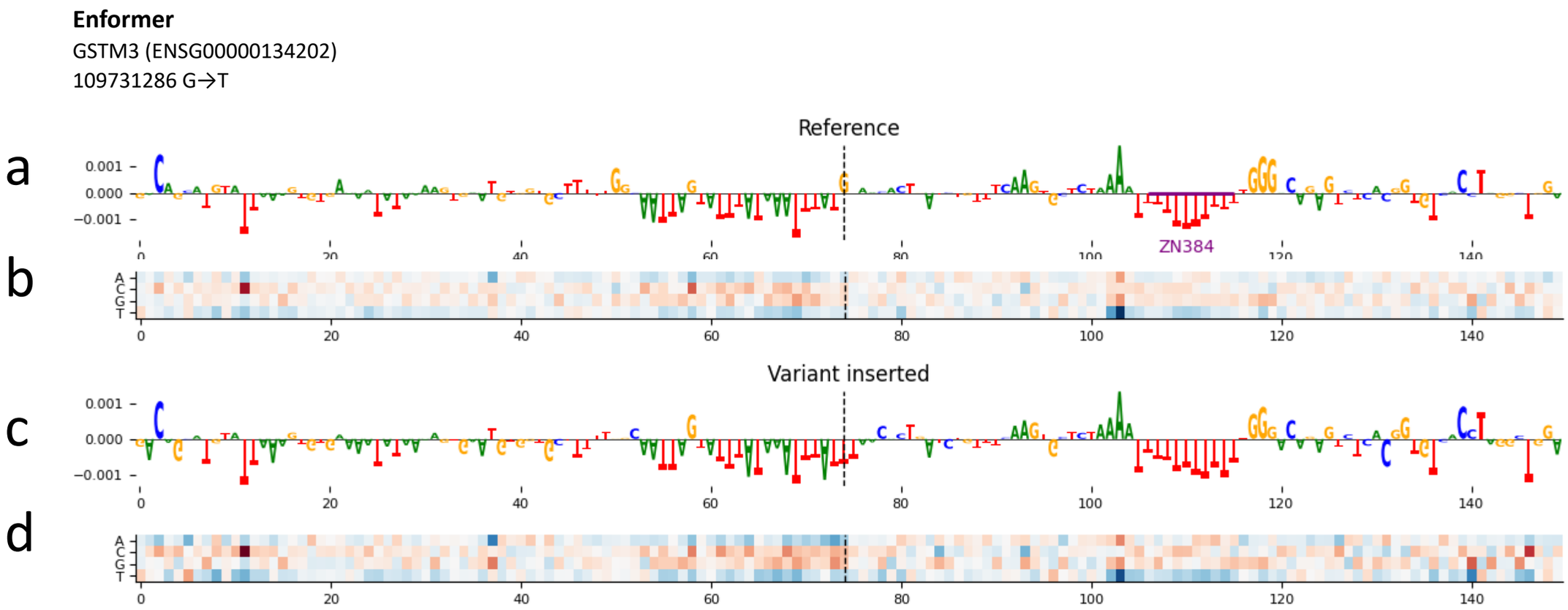


Figure S5: Enformer ISM for GSTM3. Same analysis as Fig. S3, but using fine-tuned Enformer instead of p-SAGE-net.

Figure S6

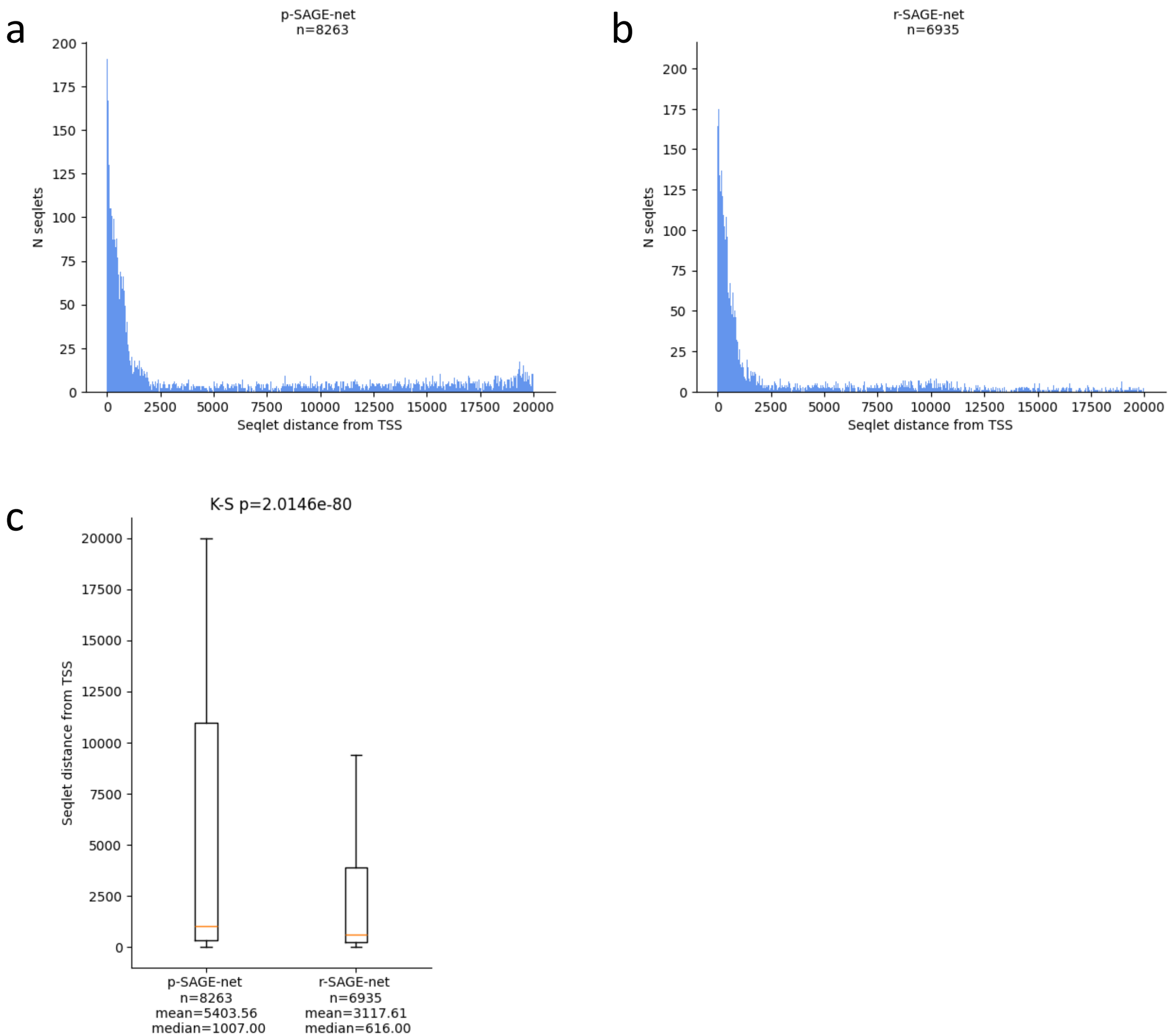


Figure S6: Seqlet distances from TSS for p-SAGE-net vs. r-SAGE-net. (a) Distribution of p-SAGE-net seqlet distances from TSS. Analysis shown (combined) for high performance genes in model training set (n=466 genes with GTEx per-gene correlation>0.3). We approximate ISM values using gradients (21) and then identify seqlets with p<0.005 (see Methods). (b) Same analysis as (a), but with r-SAGE-net instead of p-SAGE-net. (c) Comparison between seqlet distances from gene TSS for p-SAGE-net and r-SAGE-net, p-values from the two-sample Kolmogorov-Smirnov test for goodness of fit.

Figure S7

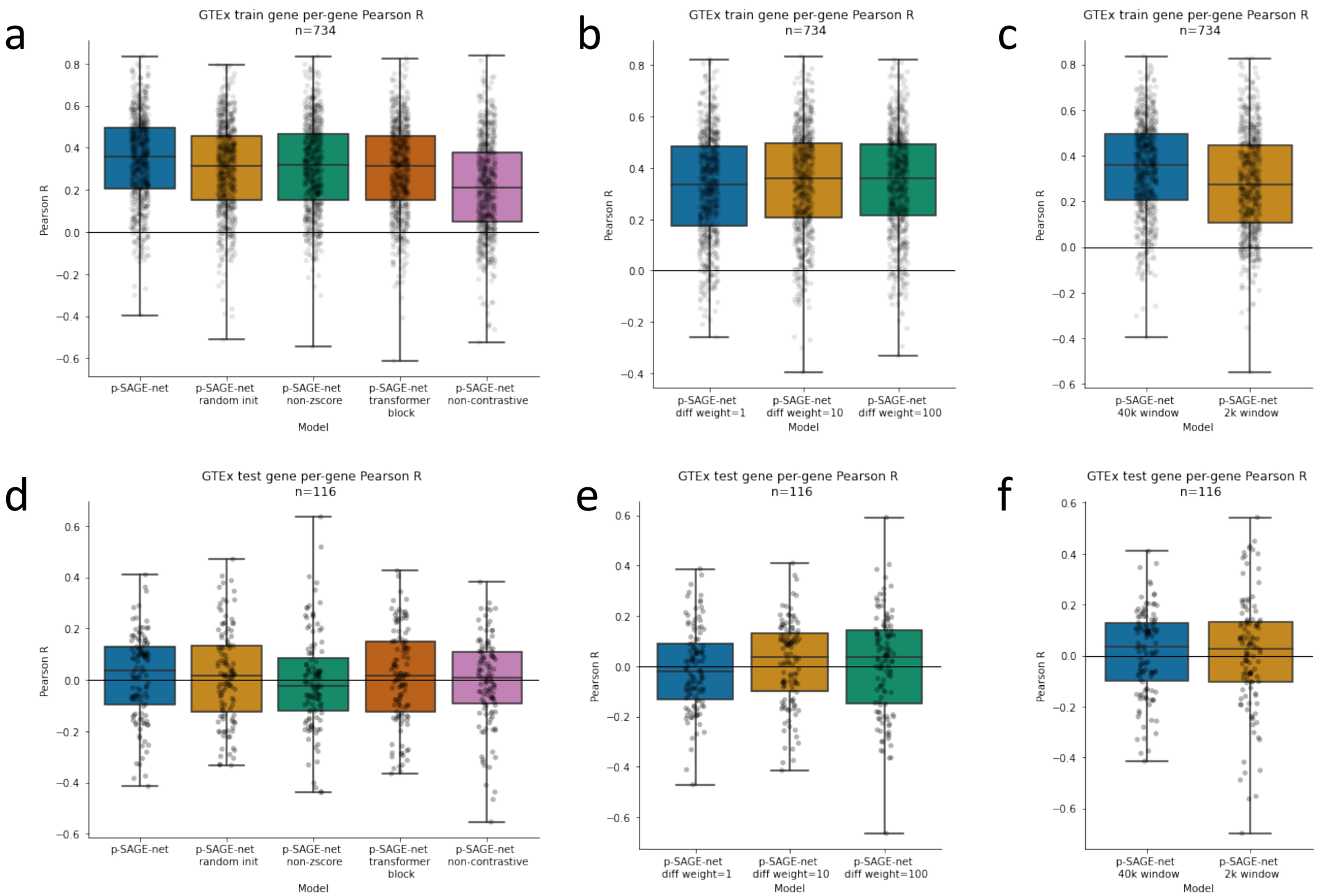


Figure S7: Model ablation analyses. (a) Comparison between best model (p-SAGE-net) and variations: randomly initializing model weights instead of initializing from r-SAGE-net ("random init"), using personal expression difference instead of z-score for model difference output ("non-zscore"), replacing the last convolutional block with a transformer block ("transformer block"), and predicting personal gene expression from personal sequence, without the contrastive approach ("non-contrastive"). See Methods for details on each variation. Each model is trained on 689 ROSMAP training individuals for 734 training genes from the top 1000 gene set and evaluated on 205 GTEx individuals for the same gene set. Each dot is one gene, boxplots show interquartile range with whiskers extending to minimum and maximum. (b) Modifications to loss function hyperparameters. For each weight on the "difference" portion of the loss function (diff weight = 1, 10, 100), the weight on the "mean expression" portion of the loss function = 1—so when diff weight = 1, the two portions of the loss function are equally weighted. The difference weight selected for all other analyses is 10. Model training and evaluation gene and individual sets are the same as in (a). (c) Comparison between models with 40 kb vs. 20 kb input window. Window size applies to both model training and evaluation. Model training and evaluation gene and individual sets are the same as in (a). (d) Same analysis as (a), but shown for unseen test genes (n=116). (e) Same analysis as (b), but shown for unseen test genes (n=116). (f) Same analysis as (c), but shown for unseen test genes (n=116).

Figure S8

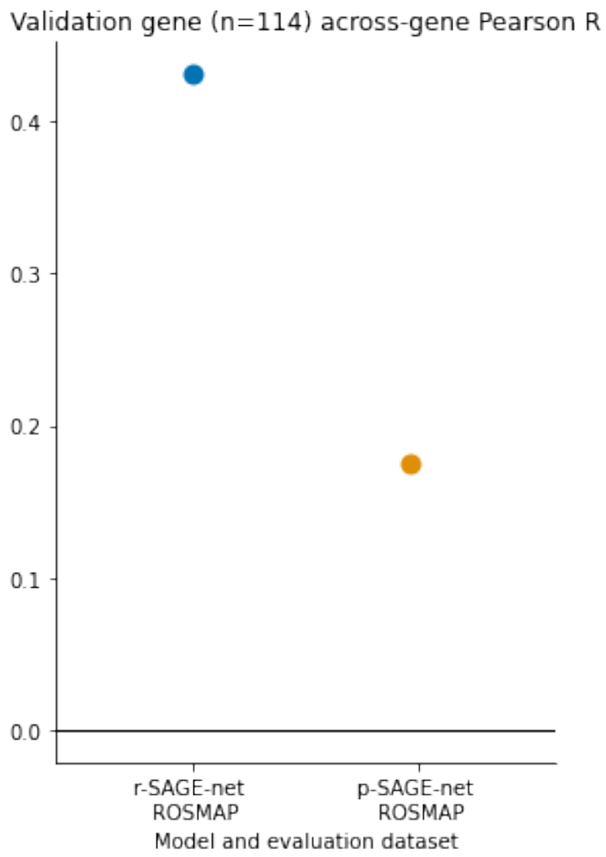


Fig. S8: Validation gene across-gene performance for p-SAGE-net and r-SAGE-net. 114 validation genes are from the top 1000 gene set used in Fig. 1c,d. P-SAGE-net and r-SAGE-net models are the same as in Fig. 1c,d.

Figure S9

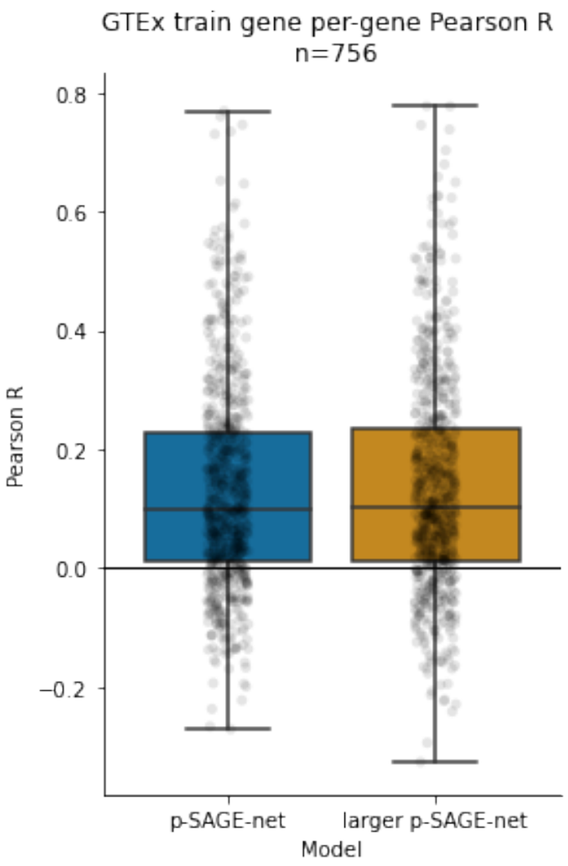


Figure S9: Training on a larger gene set with increased model capacity. The evaluation shown for p-SAGE-net is the same as Fig. 2d, "Random genes", gene set size = 3000. The evaluation shown for "larger p-SAGE-net" is the same but for a model with double the number of convolutional kernels in all convolutional layers after the first convolutional layer (512 instead of 256).

Figure S10

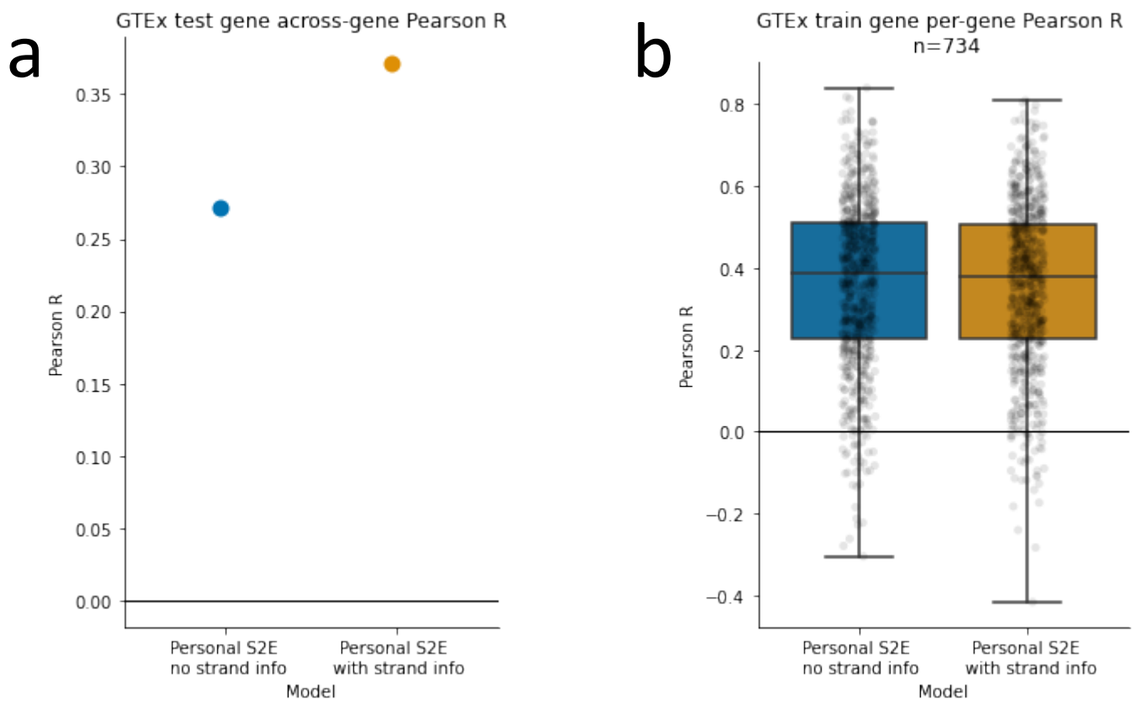


Figure S10: Effect of using gene strand information. Genes are from the top 1000 gene set. For "with strand info", we take the reverse complement of the sequence for genes on the negative strand, while for "no strand info" we do not.

Figure S11

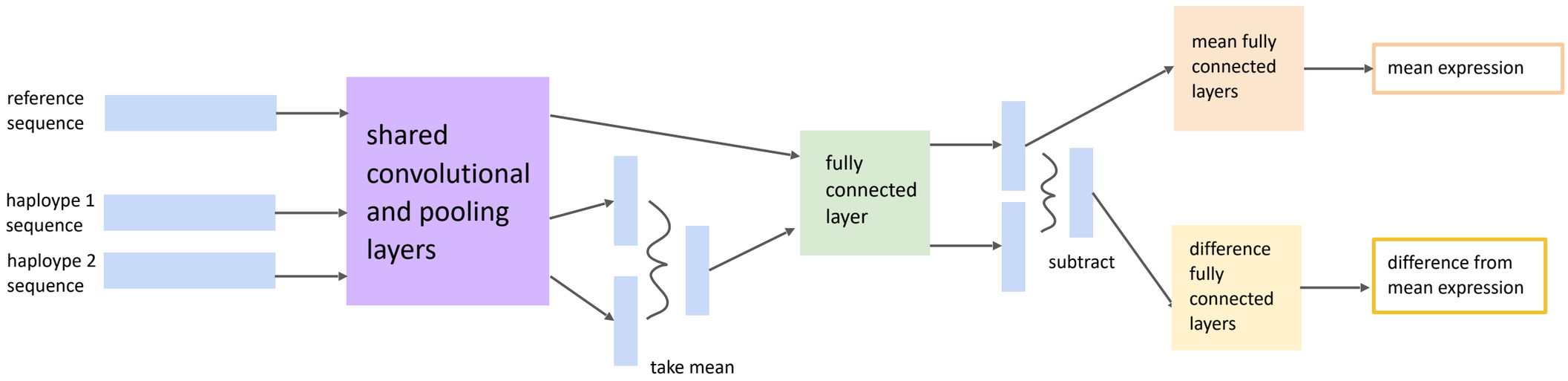


Figure S11: P-SAGE-net model architecture. Model input is reference sequence for a given gene as well as an individual's two haplotypes for that gene. These three sequences pass through the same shared convolutional layers, after which the two haplotypes are averaged. The averaged personal tensor and the reference tensor then pass through the same fully connected layer and different output heads to produce mean expression and difference from mean expression. Mean expression is predicted from reference sequence alone, while difference from mean expression is predicted using all three sequences, specifically by subtracting the "reference" tensor intermediate output from the "personal". See Methods for model layer specifics.

Figure S12

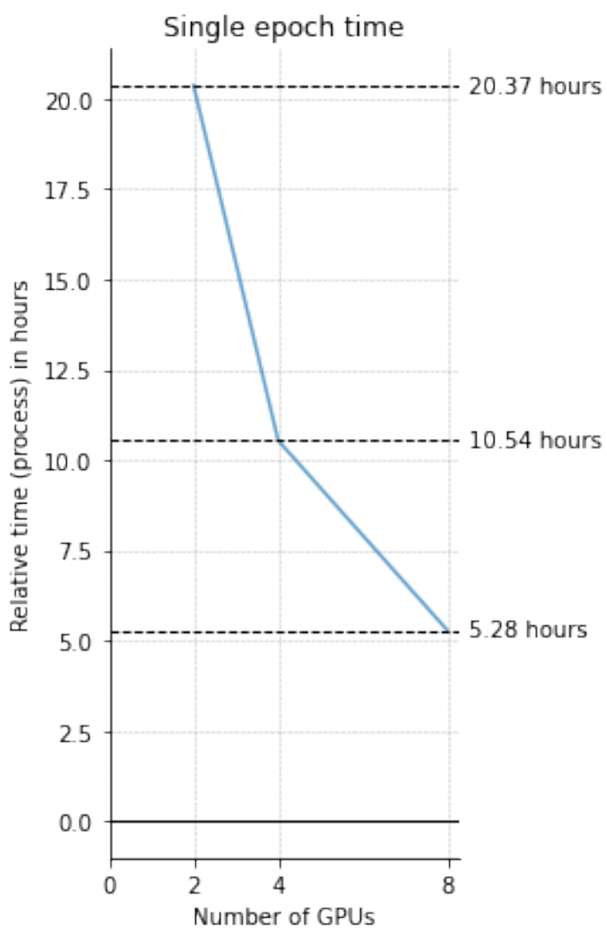


Figure S12: P-SAGE-net training time. P-SAGE-net single epoch training time, shown for training parallelized over 2, 4, 8 NVIDIA A40 GPUs. For one epoch, the model is trained on 689 individuals x 741 genes, evaluated on 85 individuals x 114 genes.

Figure S13

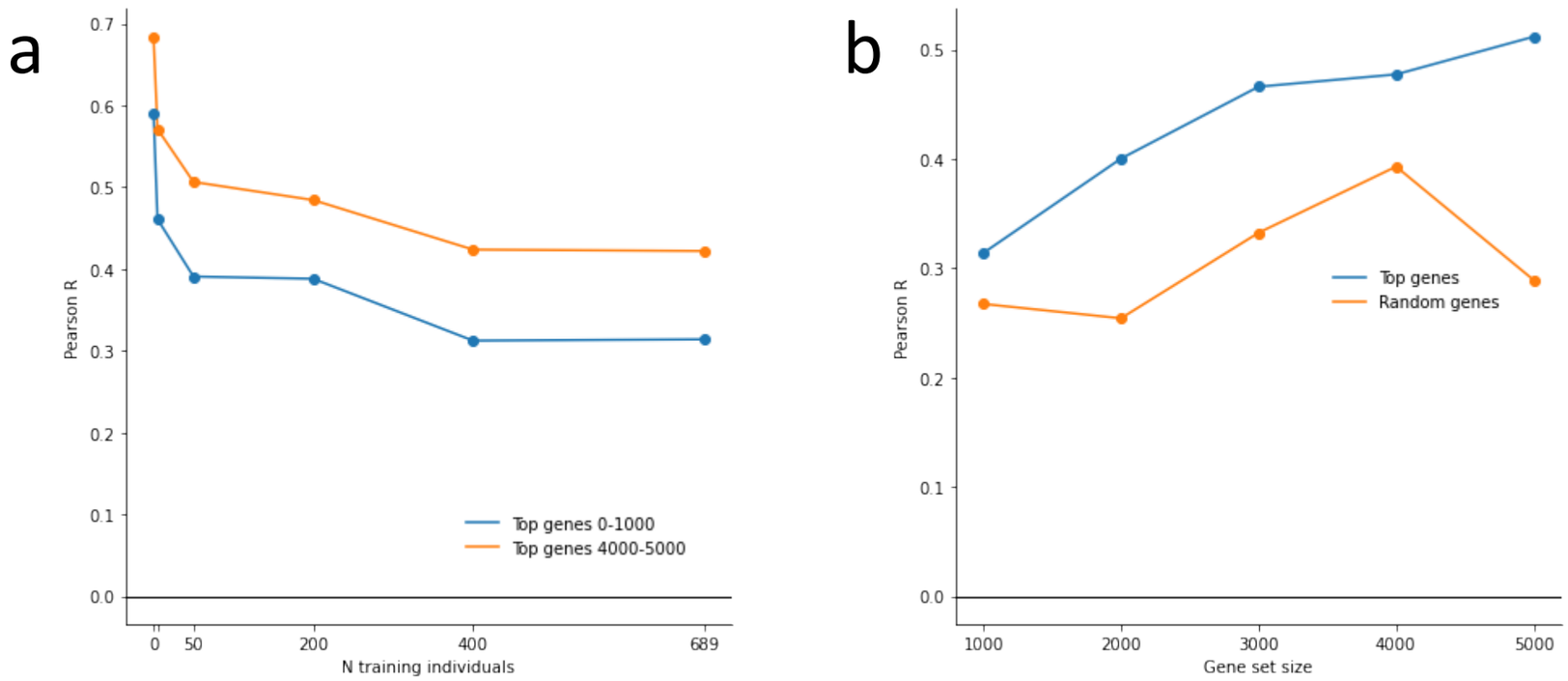


Figure S13: Unseen gene across-gene performance across number of training individuals, gene set size. (a) Same analysis as in Fig. 2c, but shown for unseen genes, across-gene correlation. (b) Same analysis as in Fig. 2d, but shown for unseen genes, across-gene correlation.