Multi-Dataset Integration and Residual Connections Improve Proteome Prediction from Transcriptomics Using Deep Learning

This manuscript ([permalink](#)) was automatically generated from [xomicsdatascience/transcriptome-proteome-nas-manubot@5d752f7](#) on July 25, 2024.

Authors

- **Caleb W. Cranney**

  ⓘD [0000-0001-8482-758X](#) · ◯ [CCranney](#) · 🐦 [CalebCranney](#)

  Department of Computational Biomedicine, Cedars Sinai Medical Center, Los Angeles CA 90048; Smidt Heart Institute, Cedars Sinai Medical Center, Los Angeles CA 90048; Advanced Clinical Biosystems Research Institute, Cedars Sinai Medical Center, Los Angeles CA 90048 · Funded by Grant R35GM142502, National Institute of General Medical Sciences (NIGMS)

- **Jesse G. Meyer**

  ⓘD [0000-0003-2753-3926](#) · ◯ [jessegmeyerlab](#) · 🐦 [j_my_sci](#)

  Department of Computational Biomedicine, Cedars Sinai Medical Center, Los Angeles CA 90048; Smidt Heart Institute, Cedars Sinai Medical Center, Los Angeles CA 90048; Advanced Clinical Biosystems Research Institute, Cedars Sinai Medical Center, Los Angeles CA 90048 · Funded by Grant R35GM142502, National Institute of General Medical Sciences (NIGMS)

✉ — Correspondence possible via [GitHub Issues](#)

Abstract

Proteomes are well known to poorly correlate with transcriptomes measured from the same sample. While connected, the complex processes that impact the relationships between transcript and protein quantities remains an open research topic. Many studies have attempted to predict proteomes from transcriptomes with limited success. Here we use publicly available data from the Clinical Proteomics Tumor Analysis Consortium to show that deep learning models designed by neural architecture search (NAS) achieve improved prediction accuracy of proteome quantities from transcriptomics. We find that this benefit is largely due to including a residual connection in the architecture that allows input information to be remembered near the end of the network. Finally, we explore which groups of transcripts are functionally important for protein prediction using model interpretation with SHAP.

Introduction

The central dogma of biology posits a linear flow of information from genetic encoding to mRNA transcripts to functional proteins. However, this seemingly straightforward relationship belies a more intricate reality, where the interactions between omic layers are multifaceted and complex. Elucidating these relationships is crucial for understanding biological systems in both healthy and diseased states. By disentangling these interactions, researchers can identify markers and patterns specific to complex diseases, ultimately enabling the development of targeted treatments.

Among the multi-omic data type relationships, the connection between transcripts and their corresponding proteins is particularly enigmatic. Despite being quantifiable and having a direct derivative relationship, their relative quantities often exhibit only weak correlations [1], [2]. Research has identified several possible causes for this discrepancy, including alternate rates of protein generation and decay [3], [4], [5], varying reactions to environmental stimuli [6], [7], or simply systematic experimentation error bias [8], [9]. Recent works have even found that the most predictive transcripts of a protein include those that are involved in protein-protein interactions [10]. While predicting protein quantity through direct transcript-to-protein correlation remains elusive, using contextual transcript information may reveal more complex proteomic-transcriptomic relationships that could be leveraged to predict one from the other. Moreover, predicting proteins from transcripts could reduce the time and financial costs associated with future studies, as transcript quantification is generally easier to accomplish than protein quantification [11].

The National Cancer Institute's (NCI) Clinical Proteomic Tumor Analysis Consortium (CPTAC) provides a valuable resource for exploring this issue, offering multi-omic datasets that enable research into healthy and cancerous disease states [12]. Specifically, CPTAC datasets comprise transcriptomic and proteomic data (among others) from various cancer types and adjacent healthy tissue, which makes this data useful for detection of inter-omic relationships. Several research collaborations have occurred to use this data to study RNA-protein relationships. The 2017 NCI-CPTAC DREAM Proteogenomics challenge [13/], a sub challenge of which aimed to predict protein quantities from transcript abundance, is a notable example. Contestants used a variety of methods, including random forest regression, genetic models, spline regression, linear regression, and elastic net methods, as well as ensemble combinations [14]. Using a test set only from breast cancer and ovarian cancer, the winning model achieved a pearson correlation of 0.41 and 0.47 between the true and predicted protein quantities, respectively [15]. Notably, deep learning neural networks were not strongly represented in the challenge results.

Since the close of the challenge, research into deep learning has expanded significantly, with a focus on replicating human behaviors like computer vision and natural language processing (NLP). However, the underlying principles of deep learning are equally applicable to biological data [16], [17], [18]. The key challenge lies in developing a model architecture that best fits a given problem. One underutilized strategy for deep learning model design is the NAS [19]. NAS serves a function like hyperparameter tuning, in that a range of values for specific hyperparameters are evaluated to obtain the optimal configurations. In the case of NAS, the concept is expanded to include model architecture in addition to hyperparameters, designing optimal and unique model architectures in an automated fashion. While NAS has been applied to genomic data [20], its application to predicting proteomic data from transcriptomic data remains unexplored.

We previously showed how machine learning can accurately predict the metabolome from the proteome, and how model interpretation revealed important biological insights [21]. Here, we extend that work to transcript-to-protein deep learning prediction models and demonstrate that utilizing NAS improved the accuracy. Furthermore, we highlight the potential of model interpretation to identify

patterns in transcript-protein relations that underpin biological processes characteristic of a disease state.

Methods

Data Acquisition and Preprocessing

To facilitate readability and reproducibility, the code for downloading, processing, and splitting data was developed as a multi-class in-house package. The bridge design pattern specifically was used to allow for interchangeability of data source input to expand beyond CPTAC in the future, as well as for allowing custom processing and splitting depending on the experiment. This involved writing an abstract parent data processing class, and individual child classes would utilize compartmentalized class components specific to the experiment. This was done to enable external researchers to trace data processing workflows with ease.

CPTAC data was downloaded directly from zenodo using the cptac python package [22] separately for each cancer type. Analyses were only performed using transcripts and proteins common between all datasets.

Cancer-specific datasets were normalized independently of one another. For experiments where each dataset required an identical train-validation split, this normalization was calculated on the training partition then applied to the training and validation partitions both. For the five by two cross validation experiments, it was determined that dataset-specific analyses would be enhanced by the inclusion of all other datasets in the training dataset [Supplementary Figure 1]. In these instances, the train-validation split was applied only to the target dataset, with a split of 0.45, 0.45, and 0.1 for training, validation and testing, respectively. All non-target datasets were normalized on the entire dataset as the training partition, ignoring validation or testing partitioning entirely. In a standard data partition, each dataset had a split of 0.8, 0.1 and 0.1 for training, validation and testing, respectively.

The final dataset contained data for breast cancer (BRCA), kidney cancer (CCRCC), colon cancer (COAD), brain cancer (GBM), squamous cell cancer (HNSCC), lung cancer (LSCC and LUAD), ovarian cancer (OV), and pancreatic cancer (PDAC) as well as adjacent healthy tissue. 59286 transcripts and 7822 proteins were shared across all datasets with a combined number of 1256 samples.

NAS

The search space used in this study chose a model architecture consisting of three segments, or blocks, of sub-architectures. These blocks could vary in number of neurons, number of layers, activation functions between layers, intra-block residual connections, dropout rates, or be removed entirely to simplify the network. It was determined that while mRNA to encoded protein quantities are not directly correlated, the quantity of one likely has a strong impact on the quantity of the other. Thus, the search space also included a residual connection inserting mRNA input quantities for proteins being predicted right before the final output layer of the network.

The NAS workflow was based on the "Multi-Objective NAS with Ax" workflow tutorial on the official pytorch website [23], utilizing Meta's Ax package to do so. The process includes designing a search space as a separate python script that accepts variables that dictate the model structure, setting up a torchx runner and scheduler for submitting model training scripts concurrently, and defining optimization requirement configurations. Ax uses Bayesian optimization to evaluate and compare model configurations and their predictive accuracy, highlighting the impact specific architecture decisions have on the final loss.

Model Evaluation and Comparison

Losses between predicted and true outputs were calculated using mean-squared error. The dummy regressor identified the mean of the true output data and used it as the prediction of all data points. The random forest regressor was run with log2 max feature and 50 node max depth limitations, as the number of inputs and outputs in creating a forest of full trees would otherwise require upwards of years to calculate. The manually designed model consisted of two hidden layers with output sizes of 12k, and 10k, respectively. Hidden layers employed batch normalization, a dropout rate of 0.6, and used the leaky ReLU activation function with a negative slope of 0.05 [Supplementary Figure 2]. The NAS-optimized model consisted of three blocks of layers, followed by a single output layer. The first block consisted of a single neural layer with 319 neurons, a sigmoid activation function, a dropout rate of 0.52. The second block consisted of three neural layers with 508 neurons, a sigmoid activation function, a dropout rate of 0.69, and a residual connection skipping the middle layer. The third block consisted of a single neural layer with 7822 neurons (the output size), a tanh activation function, and a dropout rate of 0.9. The optimal model also sported a batch size of 128 and a learning rate of 1e-4.

Model Interpretation Using SHAP Values

SHAP values [24] were calculated using all available samples from CPTAC used in training and validation of the optimal model, which included the direct transcript residual connection. SHAP values for the top 13 accurately predicted proteins across cancers using the NAS optimized model were extracted and graphed independently, specifically CAVIN1, FERMT2, FLNA, HCLS1, TK3, MCM3, MCM4, MCM6, P4HB, PTPN6, SMC2, STAT1, and VCL. MMP14 was included as a candidate because of its role in cancer regulation. SHAP values for targeted proteins were extracted and analyzed independently of the raw SHAP outputs for memory efficiency purposes.

Direct transcripts were determined by directly matching gene names to the list of predicted proteins. A cutoff of -15 mean absolute SHAP value was chosen to separate correlated and uncorrelated transcripts. Transcripts that had an absolute mean SHAP value above -15 in at least 70% of the chosen proteins were determined to belong to the correlated category. Transcripts that had a null or 0 SHAP value were excluded from categorization.

A SHAP analysis was also run on 3 variations of the optimal model, specifically models with no input residual connection, a correlated transcript residual connection, and an uncorrelated transcript residual connection. To accommodate residual connections of varying lengths, the output size of the third block and the input size of the final output layer were altered to match the size of the residual connection for these adjustments.

Code Availability

CPTAC data was downloaded using the cptac python package [22]. All models were developed using Pytorch [25] and Pytorch Lightning [26]. NAS was implemented with Meta's Adaptive Experimentation Platform (Ax) [27]. NAS evaluation metrics were tracked with tensorboardX [28]. Scikit-learn was used to perform train-validation splits and several non-neural net regression models [29]. Pandas was used to load, process and save CPTAC data [30]. Numpy was used to perform various calculations [31]. SHAP was performed with the shap python package [24].

Code for data processing and model training and evaluation can be found at [https://github.com/xomicsdatascience/RnaToProteinDataModule]. Classes for data processing and model generation can be found in the src directory, while scripts for running the different experiments can be found in the scripts directory.

Large Language Model Edit

Results

CPTAC Model Configuration

The CPTAC dataset provides a unique opportunity to investigate the relationships between multi-omic layers, particularly in predicting the proteome from transcriptome data [Figure 1A]. A range of approaches were employed to achieve this, including one-to-one mapping of mRNA transcripts to proteins compared with various regressors: dummy, linear, random forest, and neural networks [Figure 1B]. The latter proved to be the most variable, with different architectures yielding diverse predictive accuracies. To address this, a NAS was utilized to automate the selection of optimal neural network architectures [Figure 1C]. The NAS approach iteratively evaluated various model architectures from a search space to identify the most effective structure for predicting protein quantities from transcriptome data [Figures 1C, 1D]. The optimal NAS model was benchmarked against other models, including a dummy regressor, and its performance was evaluated using a 100x2 cross-validation metric across different cancer types. The results showed that the optimal NAS model consistently outperformed all other models, including the previously optimal random forest model [Figure 1E, Supplementary Figure 3].
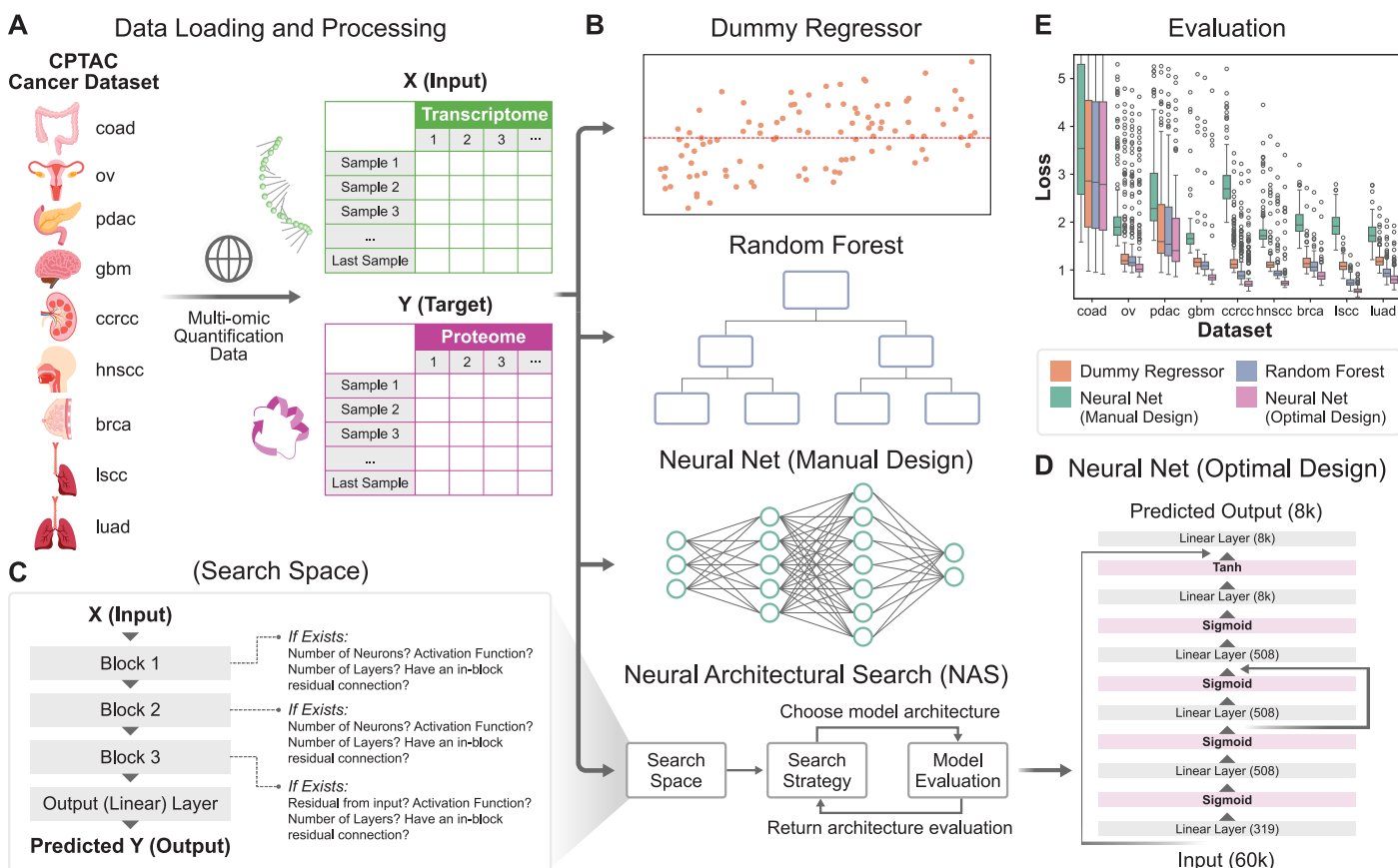
Figure 1: Comparison of methods for predicting the proteome from the transcriptome. (A) CPTAC cancer datasets were downloaded from the web and processed to match samples across omic layers. (B) Four primary methods were evaluated for predicting the proteome from the transcriptome. They are a dummy regressor to serve as a benchmark, a random forest regressor, a manually designed neural network, and an optimized neural network chosen via NAS. (C) The primary outline of the NAS search space used to identify the optimized neural network. (D) The model architecture of the optimized neural network. (E) A comparison between the four methods via 100x2 cross validation, indicating the optimized neural network outperforms all the others. While the Y-axis was shortened for visibility, a rendering was provided in Supplementary Figure 3 that captures the full boxplot whiskers.

Model Performance and Correlation Analysis

Given that the proteome and transcriptome do not reliably correlate with each other, a correlation between predicted protein quantities and actual protein quantities was calculated. An example of the RNA/protein correlation for BGN is shown in Figure 2A, while the true versus neural network-predicted value of the protein is shown in Figure 2B. R2 scores were calculated between transcripts and proteins, as well as predicted proteins to actual proteins, and compared to the benchmark transcript-to-protein correlation. These analyses revealed that the predictions from the model greatly improved the prediction for this example protein. Additionally, it was assessed whether a simple linear model could sufficiently correlate predicted proteins with actual proteins. The results revealed that the optimal NAS model outperformed both the direct and linear evaluations, highlighting its effectiveness in predicting protein quantities from transcriptome data. A global view of the R2 score distributions for each of the three relationships across all 7822 genes showed that our neural network from NAS performed best [Figure 2C].

Inspection of the optimal NAS model's architecture identified key components, including a single-layer block, a three-layer block with a residual connection, and a final single-layer block with a tanh activation function [Figure 1D]. Notably, the mRNA residual connection played a crucial role in the model's accuracy. To quantify the impact of each component, individual components were selectively removed or nullified to determine their impact on the output, revealing that the mRNA residual connection had the most dramatic effect on the model's performance [Figure 2D].
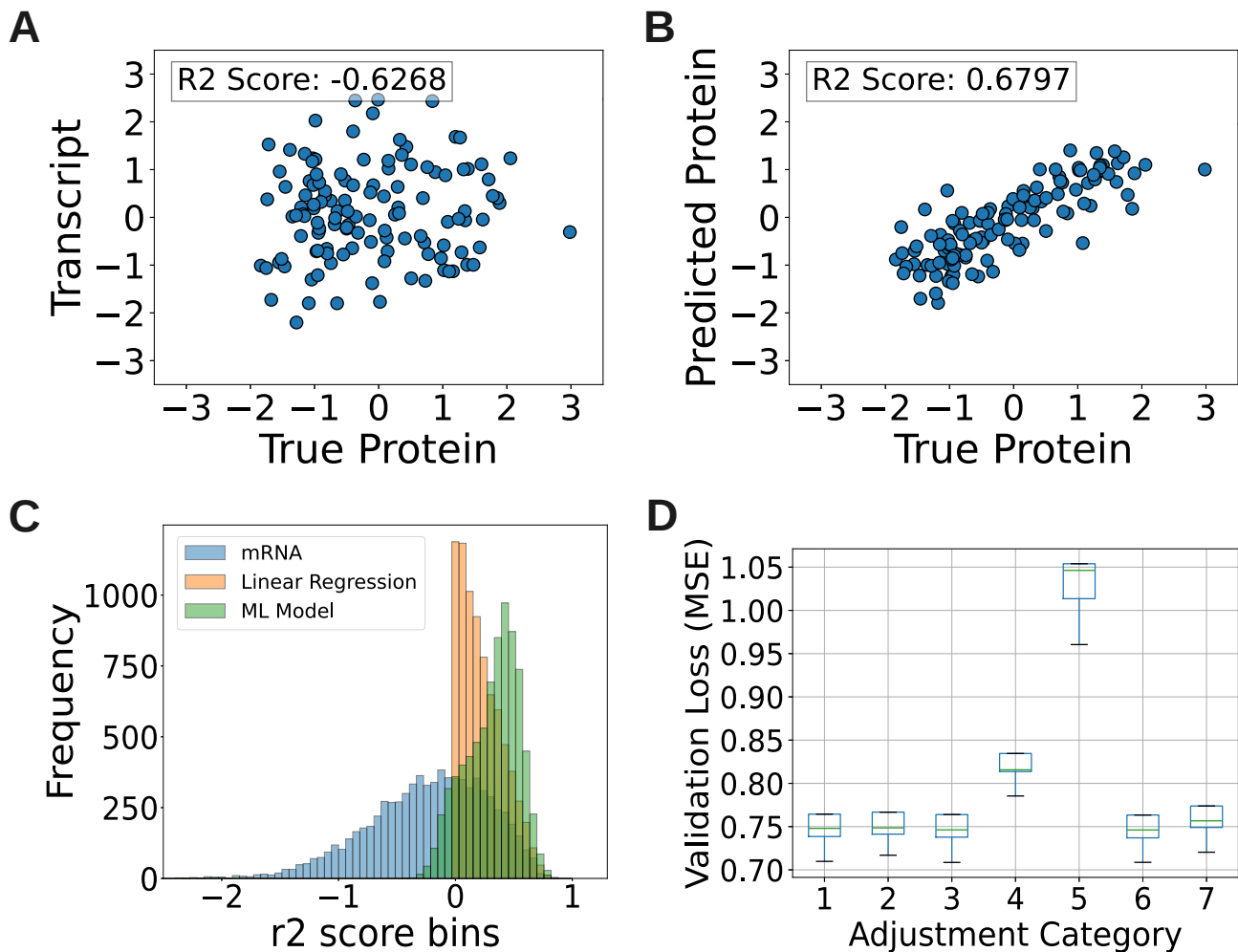
Figure 2: Evaluation of the impact and design of the NAS-optimized neural network. (A) Comparison of the normalized transcripts quantities against the matching normalized protein quantities of the validation dataset for the BGN gene. Note that the values have a negative R2 score, indicating little to no correlation. (B) Comparison of the normalized predicted output quantities against the matching true normalized protein quantities of the validation dataset for the BGN gene. Note the change in R2 score, indicating strong correlation. (C) Validation set R2 scores for every protein evaluated by the model. The different bar colors indicate which dataset the true protein quantities were

compared to, specifically matching mRNA quantities (blue), the output of a simple linear regression mapping (orange), and the output of the NAS-optimized neural network (green). (D) An evaluation of the different unique components of the NAS-optimized neural network. Corresponding to the search space summary, the categories map to (1) removal of the first block, (2) removal of the second block, (3) removal of the residual connection in the second block, (4) changing the dropout rate of the third block, 0.9, to match that of the first block, 0.52, (5) removing the residual connection from the input to the immediately before the output layer, (6) the unchanged optimized model, and (7) using a sigmoid activation function for the third block instead of the NAS-determined tanh activation function.

Biological Feature Analysis and Model Improvement

To uncover biological features underlying the transcriptome-proteome relationship, a Shapley Additive Explanation (SHAP)[24] analysis was performed to identify key transcripts correlated with protein quantities. The results showed that mRNA transcripts emphasized by the mRNA residual connection had distinctly higher average absolute SHAP values, highlighting the importance of this connection. Furthermore, a clear distinction was observed between transcripts correlated with protein quantity and those that were not, with the former exhibiting higher SHAP values. These transcripts were categorized as direct, correlated, and uncorrelated, respectively [Figure 3, Supplementary Figure 4].
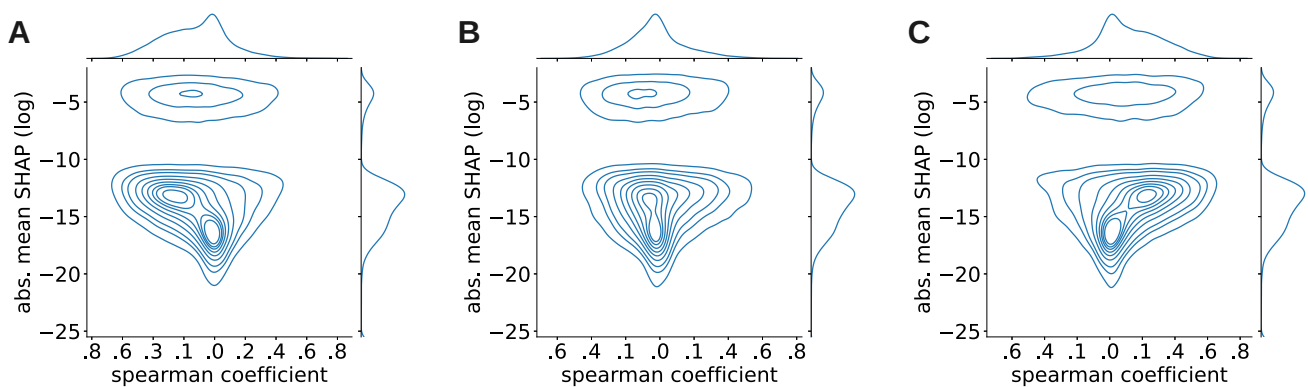
Figure 3: Model interpretation evaluating transcript impact on specific protein predictions. SHAP analysis was performed for several of the most well-predicted proteins using all samples in the dataset. As a correlative benchmark, the spearman coefficient between each transcript quantity and the specific protein quantity was also determined. A spearman coefficient of 0 indicates no correlation while negative or positive values indicate negative or positive correlation, respectively. A kernel density graph (KDE) was plotted for mean absolute SHAP (y axis) versus spearman coefficient (x axis) for each transcript's relation to (A) MCM6, (B) VCL, and (C) FERMT2 proteins specifically. Three general patterns of transcripts appeared, namely a high-impact group, a semi-correlated mid-impact group, and an uncorrelated low-impact group.

Direct and correlated transcripts might serve similar roles in predictive performance and the residual connection using direct transcripts may bias their importance according to SHAP. Instead, incorporating correlated transcripts into the residual connection could enhance the model or alter their importance. To test this, the input residual connection was manually configured to allow for differential insertion of each transcript category. The results showed that insertion of each group into the residual connection had a different effect on model performance and their SHAP values; while uncorrelated transcripts had a marginal impact on model accuracy, direct and correlated transcripts significantly improved predictive performance. Notably, direct transcripts had the largest impact, but correlated transcripts also demonstrated a substantial effect, suggesting a relationship between these transcripts and the predicted proteins [Figure 4]. It is also notable that groups promoted with a residual connection had a consistent corresponding boost in SHAP values, indicating a relationship between model architecture and interpretability.
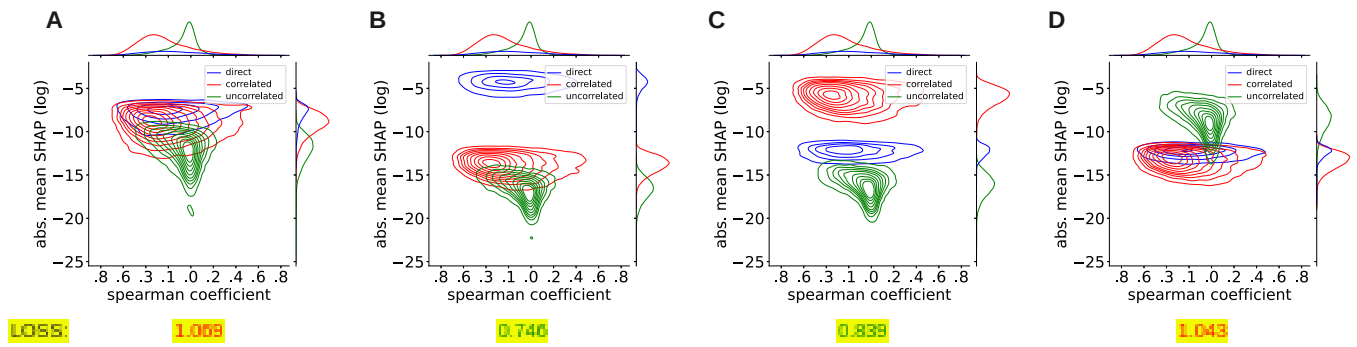
Figure 4: Evaluation of the input residual connection impact using several different categories of transcripts.
Kernel density estimation (KDE) for the MMP14 protein. Differing categories of transcripts were applied as the input residual connection, and the resulting impact on SHAP values between each category was plotted. The training loss for each model is included as a benchmark, with red text indicating high loss values and green text indicating low loss values. (A) Evaluation of the model where the input residual connection was removed entirely. (B) Evaluation of the model where only transcripts that directly matched to target proteins were used in the residual connection. (C) Evaluation of the model where only non-direct transcripts above a log absolute mean SHAP value of -15 were included in the residual connection. (D) Evaluation of the model where only non-direct transcripts below a log absolute mean SHAP value of -15 were included in the residual connection. Notably, each time a group is used as an input residual connection, the general impact of group members exceeds transcripts outside the group. The direct transcript model ultimately outperforms all others by mean-squared error (MSE) loss metrics.

The types of transcripts in each category differed in proportion. While the direct transcripts consisted entirely of coding proteins, the correlated and uncorrelated transcripts had different mixtures of various RNA types. The correlated transcripts emphasized long non-coding RNAs and indirect protein coding mRNAs, while uncorrelated transcripts emphasized processed pseudogenes, though all three types were found in both groups [Figure 5, Supplementary Table 1]. Overall, the findings highlight the importance of optimizing neural network architectures and incorporating biologically relevant features, such as mRNA residual connections and correlated transcripts, to improve the prediction of protein quantities from transcriptome data.
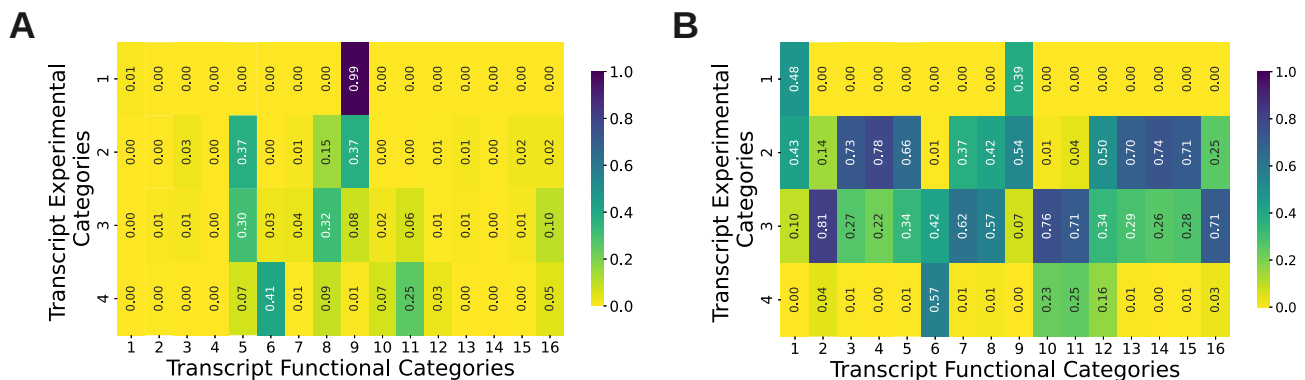
Figure 5: Evaluation of the functional composition of experimentally-derived categories of transcripts.
Transcript experimental categories are (1) direct transcripts, (2) correlated transcripts, (3) uncorrelated transcripts, and (4) transcripts in the dataset with zero quantity in the analyzed samples. Transcript functional categories are (1) 'IG_V_gene,' or immunoglobulin variable chain genes, (2) 'IG_V_pseudogene,' or inactivated immunoglobulin genes, (3) 'TEC,' or transcripts to be experimentally confirmed, (4) 'TR_V_gene,' or T-cell receptor genes, (5) 'lncRNA,' or long non-coding RNAs, (6) 'miRNA,' or microRNAs, (7) 'miscRNA,' or miscellaneous other RNAs, (8) 'processed_pseudogene,' or pseudogenes that lack introns, (9) 'protein_coding,' or transcripts that contain an open reading frame, (10) 'rRNA_pseudogene,' or non-coding ribosomal RNAs predicted to be a pseudogenes by the Ensembl pipeline, (11) 'snRNA,' or small nuclear RNAs, (12) 'snoRNA,' or small nucleolar RNA, (13) 'transcribed_processed_pseudogene,' or pseudogenes that lack introns and have expression indicated by locus-specific transcripts, (14) 'transcribed_unitary_pseudogene,' or pseudogenes where the presence of locus-specific transcripts indicates expression and there is an active orthologue in another species, (15) 'transcribed_unprocessed_pseudogene,' or pseudogenes with introns and have expression indicated by locus-specific transcripts, and (16) 'unprocessed_pseudogene,' or pseudogenes with introns. (A) Heatmap of

proportions of transcripts across experimental categories. (B) Heatmap of proportions of transcripts across functional categories.

Discussion

The results of this study demonstrate the potential of NAS in optimizing deep learning models for predicting protein quantities from transcriptome data. By automating the selection of optimal neural network architectures, we were able to identify a model that consistently outperformed other approaches, including random forest and linear regression. The importance of incorporating biologically relevant features, such as mRNA residual connections and correlated transcripts, was also highlighted.

The NAS-optimized model's ability to predict protein quantities from transcriptome data with improved accuracy has significant implications for the field of proteogenomics. By leveraging the strengths of deep learning and multi-omic data, we can gain a better understanding of the complex relationships between different omic layers. This, in turn, can lead to the identification of novel biomarkers and therapeutic targets for diseases. We expect that as larger and larger datasets become available for training; the approach we outlined here will improve this prediction further.

The SHAP analysis revealed several distinct categories of transcripts that had varying degrees of impact on protein quantity prediction. These findings suggest that the model can capture biologically meaningful patterns in the data, and that the incorporation of correlated transcripts can enhance predictive performance in future models. Although the residual connection clearly biased the group of direct transcripts, Figure 5 suggests that different classes of transcripts have different effects on protein quantities. For example, snRNA, which functions in the spliceosome to process pre-RNA, have almost no specific impact on protein quantity prediction and are found in the unimpactful, uncorrelated transcript category. Messenger RNAs generally have a high impact on predictive function. Long non-coding RNAs seem broadly more impactful for a gene's quantity, whereas the importance of pseudogenes appears to depend on their state of processing. Only a small subset of microRNAs (1%) were important for predicting this set of proteins, supporting their role in specific biological regulation.

Conclusion

This study demonstrates the potential of NAS and deep learning in optimizing protein quantity predictions from transcriptome data. The results highlight the importance of incorporating biologically relevant features and the need for further research into the development of more accurate and efficient models. By leveraging the strengths of these approaches, we can gain a better understanding of the complex relationships between different omic layers and make significant progress in the field of proteogenomics.

Acknowledgements

Supplementary Materials
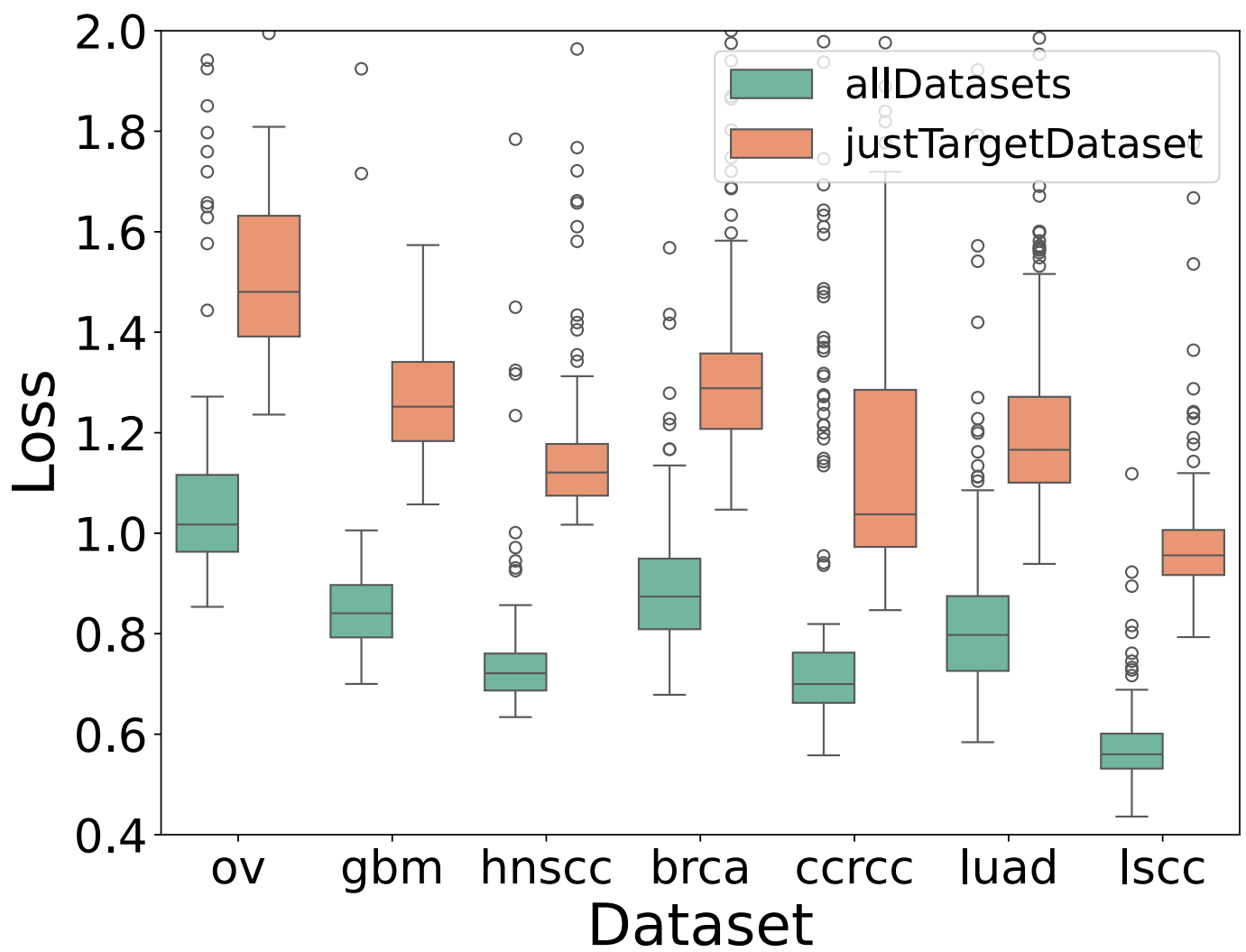
Figure 6 - figure supplement 1: Comparing all dataset use in training vs. only target dataset use in training.
Target dataset had an 80:20 trainining:validation split for all replicate calculations. In the justTargetDataset runs, no addiional changes were made. In the allDatasets runs, the training data was supplemented with all full non-target datasets.

Predicted Output (8k)

Linear Layer (output size)

Leaky ReLU

Linear Layer (10k)

Leaky ReLU

Linear Layer (12k)

Input (60k)
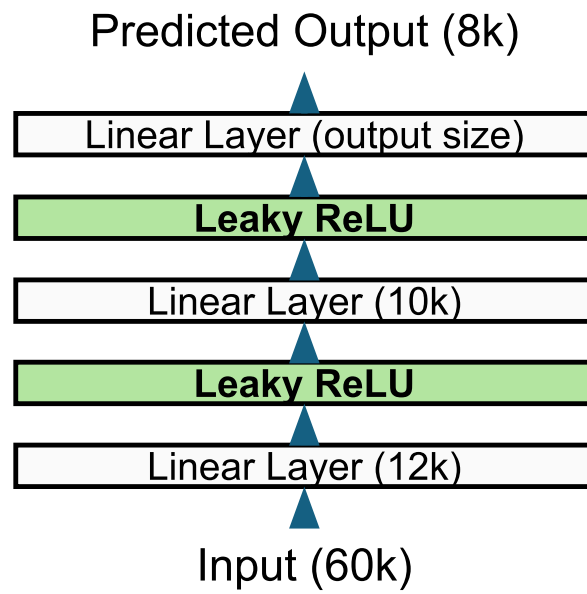
Figure 7 - figure supplement 2: Manually designed neural network architecture. Prior to Neural Architectural search optimization, a simple three-layer network with leaky relu activation functions was trained on the CPTAC datasets.
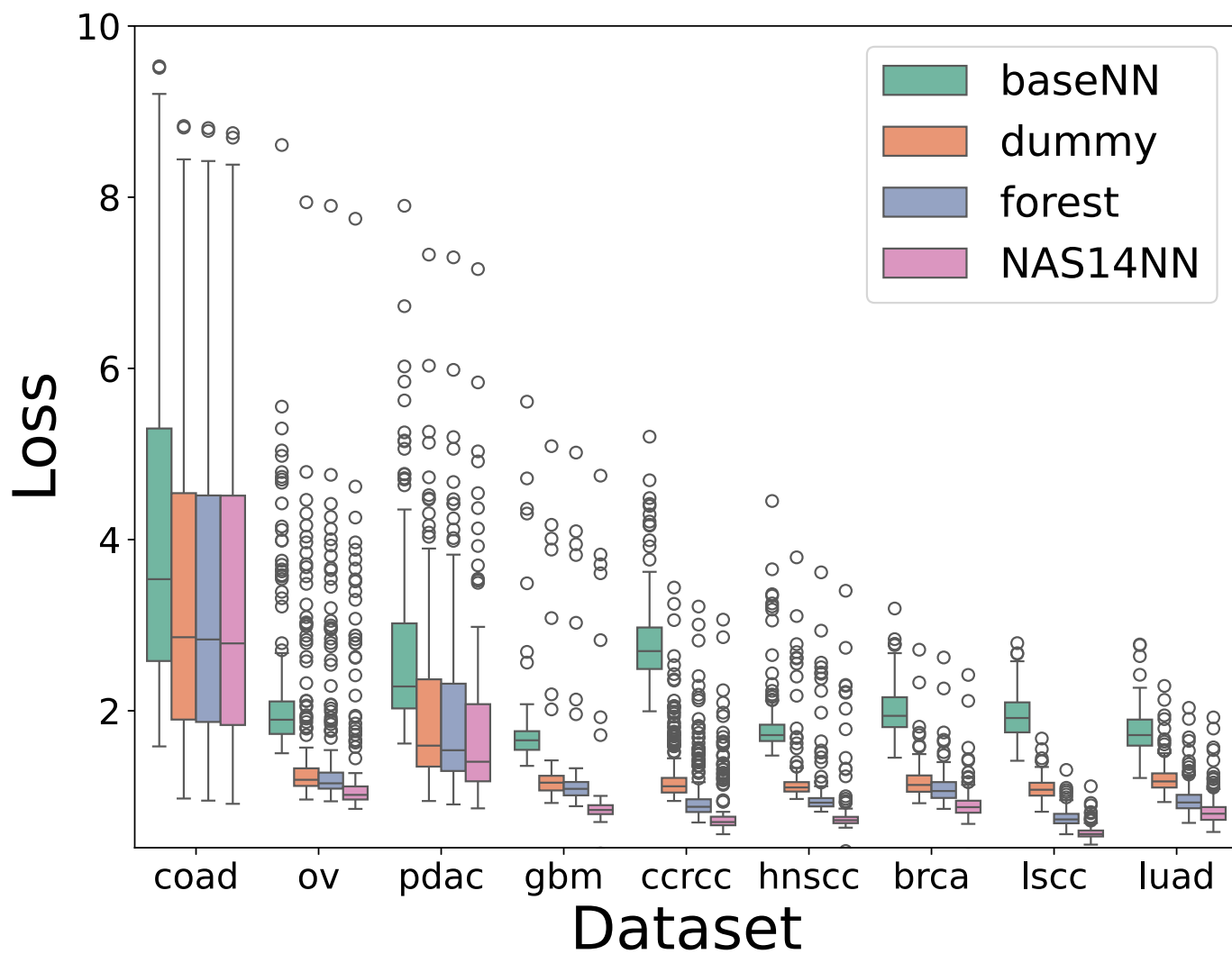
Figure 8 - figure supplement 3: Comparison of four model types via 100x2 cross validation with full whisker coverage. A rendition of figure 1E, but with full whisker coverage.
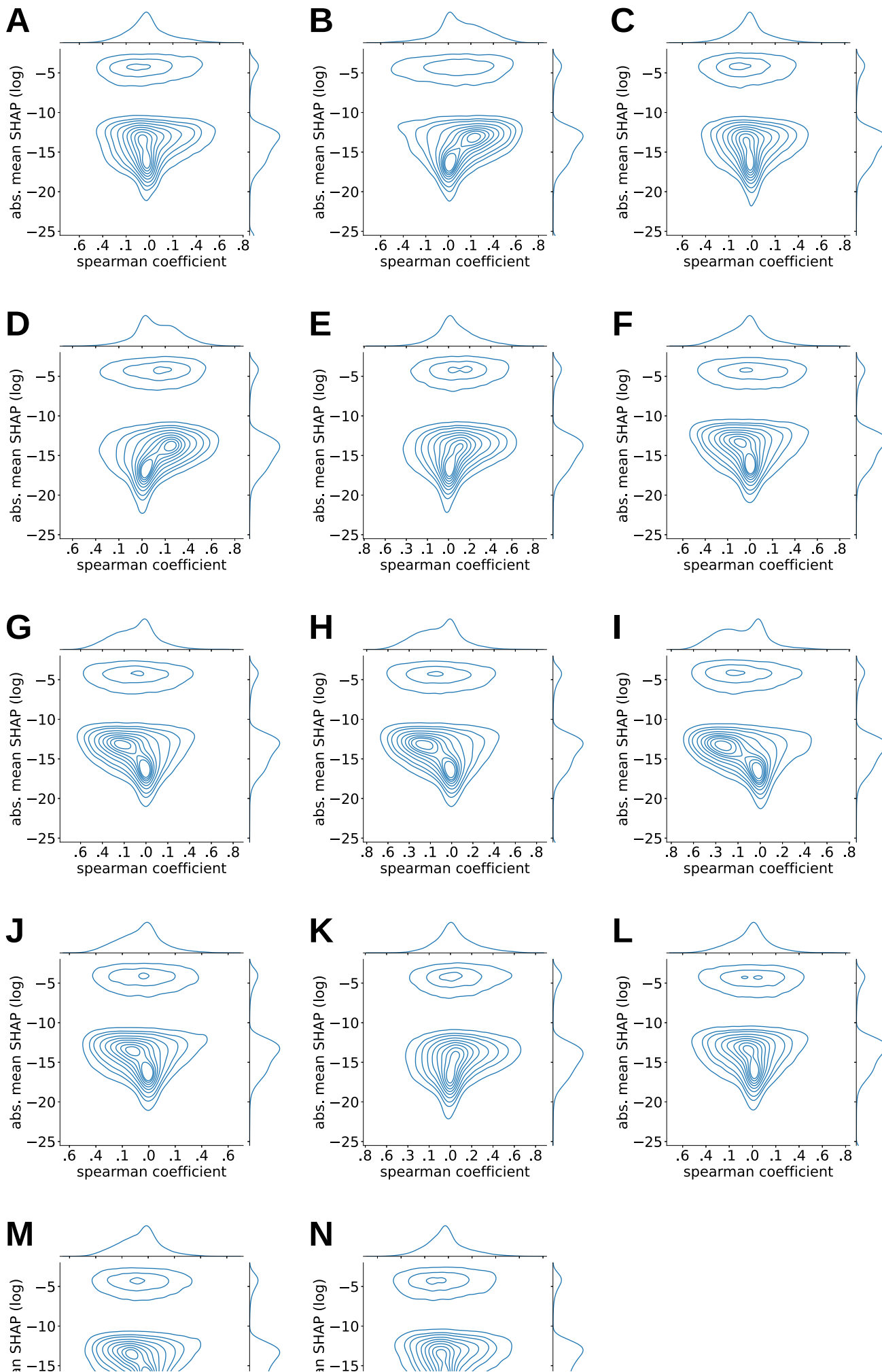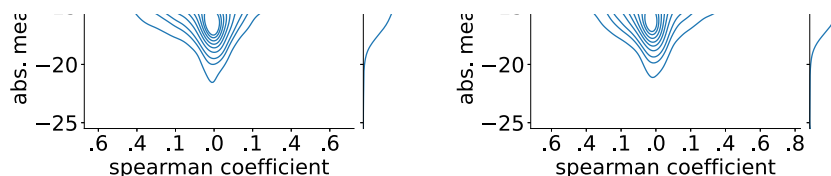
Figure 9 - figure supplement 4: Model interpretation evaluating transcript impact on specific protein predictions. SHAP analysis was performed for several of the most well-predicted proteins using all samples in the dataset. A kernel density graph (KDE) was plotted for mean absolute SHAP (y axis) versus spearman coefficient (x axis) for each transcript's relation to (A) CAVIN1, (B) FERMT2, (C) FLNA, (D) HCLS1, (E) HK3, (F) MCM3, (G) MCM4, (H) MCM6, (I) MMP14, (J) P4HB, (K) PTPN6, (L) SMC2, (M) STAT1, and (N) VCL proteins specifically.

category	direct	correlated	uncorrelated	noCount
IG_V_gene	69	62	14	0
IG_V_pseudogene	0	26	149	8
TEC	0	739	271	6
TR_V_gene	0	83	23	0
lncRNA	0	10851	5539	112
miRNA	0	8	497	668
misc_RNA	0	386	641	11
processed_pseudogene	0	4275	5760	144
protein_coding	7743	10700	1438	18
rRNA_pseudogene	0	6	370	113
snRNA	0	68	1186	409
snoRNA	0	164	110	54
transcribed_processed_pseudogene	0	352	145	4
transcribed_unitary_pseudogene	0	114	41	0
transcribed_unprocessed_pseudogene	0	654	254	7
unprocessed_pseudogene	0	649	1820	78

Supplement Table 1: Biological function categorization of transcripts for each experimental predictive category. {#tbl:transcript-function}

References

1. **Correlation between Protein and mRNA Abundance in Yeast**
   Steven P Gygi, Yvan Rochon, BRobert Franza, Ruedi Aebersold
   Molecular and Cellular Biology (1999-03-01) https://doi.org/grtw7n
   DOI: 10.1128/mcb.19.3.1720 · PMID: 10022859 · PMCID: PMC83965

2. **The utility of protein and mRNA correlation**
   Samuel H Payne
   Trends in Biochemical Sciences (2015-01) https://doi.org/gf2w7q
   DOI: 10.1016/j.tibs.2014.10.010 · PMID: 25467744 · PMCID: PMC4776753

3. **Insights into the regulation of protein abundance from proteomic and transcriptomic analyses**
   Christine Vogel, Edward M Marcotte
   Nature Reviews Genetics (2012-03-13) https://doi.org/bg9g
   DOI: 10.1038/nrg3185 · PMID: 22411467 · PMCID: PMC3654667

4. **Database for mRNA Half-Life of 19 977 Genes Obtained by DNA Microarray Analysis of Pluripotent and Differentiating Mouse Embryonic Stem Cells**
   LV Sharova, AA Sharov, T Nedorezov, Y Piao, N Shaik, MSH Ko
   DNA Research (2009-01-08) https://doi.org/cvg6wg
   DOI: 10.1093/dnares/dsn030 · PMID: 19001483 · PMCID: PMC2644350

5. **Global Protein Stability Profiling in Mammalian Cells**
   Hsueh-Chi Sherry Yen, Qikai Xu, Danny M Chou, Zhenming Zhao, Stephen J Elledge
   Science (2008-11-07) https://doi.org/dzv5xx
   DOI: 10.1126/science.1160489 · PMID: 18988847

6. **On the Dependency of Cellular Protein Levels on mRNA Abundance**
   Yansheng Liu, Andreas Beyer, Ruedi Aebersold
   Cell (2016-04) https://doi.org/f8kc6z
   DOI: 10.1016/j.cell.2016.03.014 · PMID: 27104977

7. **Gcn4p, a Master Regulator of Gene Expression, Is Controlled at Multiple Levels by Diverse Signals of Starvation and Stress**
   Alan G Hinnebusch, Krishnamurthy Natarajan
   Eukaryotic Cell (2002-02) https://doi.org/fq6gww
   DOI: 10.1128/ec.01.1.22-32.2002 · PMID: 12455968 · PMCID: PMC118051

8. **Experimental reproducibility limits the correlation between mRNA and protein abundances in tumor proteomic profiles**
   Swathi Ramachandra Upadhya, Colm J Ryan
   Cell Reports Methods (2022-09) https://doi.org/gqtvfv
   DOI: 10.1016/j.crmeth.2022.100288 · PMID: 36160043 · PMCID: PMC9499981

9. **Antibody reliability influences observed mRNA–protein correlations in tumour samples**
   Swathi Ramachandra Upadhya, Colm J Ryan
   Life Science Alliance (2023-05-11) https://doi.org/gt3ttb
   DOI: 10.26508/lsa.202201885 · PMID: 37169592 · PMCID: PMC10176110

10. **Protein prediction models support widespread post-transcriptional regulation of protein abundance by interacting partners**

Himangi Srivastava, Michael J Lippincott, Jordan Currie, Robert Canfield, Maggie PY Lam, Edward Lau

PLOS Computational Biology (2022-11-10) https://doi.org/grkjgm

DOI: 10.1371/journal.pcbi.1010702 · PMID: 36356032 · PMCID: PMC9681107

11. **Harmonization of quality metrics and power calculation in multi-omic studies**

Sonia Tarazona, Leandro Balzano-Nogueira, David Gómez-Cabrero, Andreas Schmidt, Axel Imhof, Thomas Hankemeier, Jesper Tegnér, Johan A Westerhuis, Ana Conesa

Nature Communications (2020-06-18) https://doi.org/gg2pmg

DOI: 10.1038/s41467-020-16937-8 · PMID: 32555183 · PMCID: PMC7303201

12. **CPTAC | Office of Cancer Clinical Proteomics Research**

https://proteomics.cancer.gov/programs/cptac

13. **NCI-CPTAC DREAM Proteogenomics**

admin

DREAM Challenges https://dreamchallenges.org/nci-cptac-dream-proteogenomics/

14. **Challenges in proteogenomics: a comparison of analysis methods with the case study of the DREAM proteogenomics sub-challenge**

Tara Eicher, Andrew Patt, Esko Kautto, Raghu Machiraju, Ewy Mathé, Yan Zhang

BMC Bioinformatics (2019-12) https://doi.org/gpfz94

DOI: 10.1186/s12859-019-3253-z · PMID: 31861998 · PMCID: PMC6923881

15. **Joint learning improves protein abundance prediction in cancers**

Hongyang Li, Omer Siddiqui, Hongjiu Zhang, Yuanfang Guan

BMC Biology (2019-12) https://doi.org/gr23js

DOI: 10.1186/s12915-019-0730-9 · PMID: 31870366 · PMCID: PMC6929375

16. **Ensemble deep learning in bioinformatics**

Yue Cao, Thomas Andrew Geddes, Jean Yee Hwa Yang, Pengyi Yang

Nature Machine Intelligence (2020-08-17) https://doi.org/ghhnj2

DOI: 10.1038/s42256-020-0217-y

17. **Deep learning in bioinformatics**

Seonwoo Min, Byunghan Lee, Sungroh Yoon

Briefings in Bioinformatics (2016-07-29) https://doi.org/gcgk8v

DOI: 10.1093/bib/bbw068 · PMID: 27473064

18. **Review of the Applications of Deep Learning in Bioinformatics**

Yongqing Zhang, Jianrong Yan, Siyu Chen, Meiqin Gong, Dongrui Gao, Min Zhu, Wei Gan

Current Bioinformatics (2021-01-01) https://doi.org/gt3ts9

DOI: 10.2174/1574893615999200711165743

19. **A Comprehensive Survey of Neural Architecture Search**

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-yao Huang, Zhihui Li, Xiaojiang Chen, Xin Wang

ACM Computing Surveys (2021-05-24) https://doi.org/gk76qj

DOI: 10.1145/3447582

20. **An automated framework for efficiently designing deep convolutional neural networks in genomics**

Zijun Zhang, Christopher Y Park, Chandra L Theesfeld, Olga G Troyanskaya

Nature Machine Intelligence (2021-03-15) https://doi.org/gnfth7

DOI: 10.1038/s42256-021-00316-z

21. **Multi-omic integration by machine learning (MIMaL)**

Quinn Dickinson, Andreas Kohler, Martin Ott, Jesse G Meyer
Bioinformatics (2022-09-15) https://doi.org/gq93d7
DOI: 10.1093/bioinformatics/btac631 · PMID: 36106996 · PMCID: PMC9801967

22. **Simplified and Unified Access to Cancer Proteogenomic Data**
Caleb M Lindgren, David W Adams, Benjamin Kimball, Hannah Boekweg, Sadie Tayler, Samuel L
Pugh, Samuel H Payne
Journal of Proteome Research (2021-02-09) https://doi.org/gt3tvb
DOI: 10.1021/acs.jproteome.0c00919 · PMID: 33560848 · PMCID: PMC8022323

23. **Multi-Objective NAS with Ax — PyTorch Tutorials 2.4.0+cu124 documentation**
https://pytorch.org/tutorials/intermediate/ax_multiobjective_nas_tutorial.html

24. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

25. **PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation
and Graph Compilation**
Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky,
Bin Bao, Peter Bell, David Berard, Evgeni Burovski, … Soumith Chintala
*Proceedings of the 29th ACM International Conference on Architectural Support for
Programming Languages and Operating Systems, Volume 2* (2024-04-27) https://doi.org/gt2mnc
DOI: 10.1145/3620665.3640366

26. **PyTorch Lightning**
William Falcon, The PyTorch Lightning team
(2019-03) https://github.com/Lightning-AI/lightning

27. **facebook/Ax**
Meta
(2024-07-25) https://github.com/facebook/Ax

28. **lanpa/tensorboardX**
Tzu-Wei Huang
(2024-07-25) https://github.com/lanpa/tensorboardX

29. **Scikit-learn: Machine Learning in Python**
Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion,
Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, … Édouard
Duchesnay
Journal of Machine Learning Research (2011) http://jmlr.org/papers/v12/pedregosa11a.html

30. **pandas-dev/pandas: Pandas**
The pandas development team
Zenodo (2024-04-10) https://doi.org/ggt8bh
DOI: 10.5281/zenodo.3509134

31. **Array programming with NumPy**
Charles R Harris, KJarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David
Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, … Travis E Oliphant
Nature (2020-09-16) https://doi.org/ghbzf2
DOI: 10.1038/s41586-020-2649-2 · PMID: 32939066 · PMCID: PMC7759461

32. https://ai.meta.com/blog/meta-llama-3