Large-Scale Multi-omic Biosequence Transformers for Modeling Protein-Nucleic Acid Interactions

Sully F. Chen*†, Robert J. Steele*‡, Glen M. Hocky §, Beakal Lemeneh¶, Shivanand P. Lad‖, Eric K. Oermann**

Abstract

The transformer architecture has revolutionized bioinformatics and driven progress in the understanding and prediction of the properties of biomolecules. Almost all research on large-scale biosequence transformers has focused on one domain at a time (single-omic), usually DNA/RNA or proteins. These models have seen incredible success in downstream tasks in each domain, and have achieved particularly noteworthy breakthroughs in sequence modeling and structural modeling. However, these single-omic models are naturally incapable of efficiently modeling multi-omic tasks, one of the most biologically critical being protein-nucleic acid interactions. We present our work training the largest open-source multi-omic foundation model to date. We show that these multi-omic models (MOMs) can learn joint representations between various single-omic distributions that are *emergently* consistent with the Central Dogma of molecular biology despite only being trained on unlabeled biosequences. We further demonstrate that MOMs can be fine-tuned to achieve state-of-the-art results on protein-nucleic acid interaction tasks, namely predicting the change in Gibbs free energy (ΔG) of the binding interaction between a given nucleic acid and protein. Remarkably, we show that multi-omic biosequence transformers *emergently* learn useful structural information without any *a priori* structural training, allowing us to predict which protein residues are most involved in the protein-nucleic acid binding interaction. Lastly, we provide evidence that multi-omic biosequence models are in many cases superior to foundation models trained on single-omics distributions, both in performance-per-FLOP and absolute performance, suggesting a more generalized or foundational approach to building these models for biology.

1 Introduction

It has long been a goal in bioinformatics to develop models that can extract useful and accurate information from the primary sequences of nucleic acids and protein chains. This goal is particularly desirable in modern times where primary sequence information of proteins and nucleic acids can be obtained through high-throughput and scalable technologies. The advent of deep learning has led to an explosion of progress in the last decade on machine learning, with advances in natural language processing (NLP) yielding particularly fruitful results in the processing of sequential data. Of particular note is the transformer architecture [1], which has seen incredible success in NLP tasks in the form of large language models (LLMs) such as GPT-4 [2], BERT [3], and LLaMA [4]. These results in processing natural language have recently been extended into se-

*Authors contributed equally
†Duke University School of Medicine, Durham, NC 27710, USA
‡NYU Langone Health, New York, NY 10016, USA
§Department of Chemistry and Simons Center for Computational Physical Chemistry, New York University, New York, NY 10012, USA
¶NYU Langone Health, New York, NY 10016, USA
‖Duke University School of Medicine, Department of Neurological Surgery, Durham, NC 27710, USA
**NYU Langone Health, Department of Neurological Surgery, New York, NY 10016, USA

quences of imaging data, and notably into multimodal modeling of both vision and text in modern vision-language models such as LLaVA [5]. Similar successes have already been found in bioinformatics by using transformer models to model distributions over biosequences.

The majority of research applying transformers to biosequences has focused on applying the architecture to single-omics, typically nucleic acid distributions (genomics, transcriptomics, epigenetics, etc.) or proteomics. These efforts have yielded astonishing successes in several tasks, with the most notable being the prediction of the 3D structure of proteins from their primary sequences [6–13]. Other work has focused on developing models that produce useful representations of single-omics biosequences for various downstream tasks. There exist numerous protein foundation models [14–24], and we find the most variety of model architectures in this class. Notably, there are many generative models [25–27], encoder-decoder models [21, 22], and even a diffusion model [25].

Many models primarily rely on structural information [18,21,23], with many focused solely on structure prediction [6–10, 13, 28–31]. Of these, most are transformer variants [6–10, 13], though there are notable message-passing and other graph neural network approaches [28, 29, 31] as well as a diffusion-based approach [30].

Several genomics foundation models have been trained as well, primarily on human genomics data [32–35]. Other genomic foundation models have been trained on human and murine data [36], multi-species genomes [37], prokaryotic genomes [38], and even metagenomic scaffolds [39]. Notably, very few models integrate broad, multi-species training data, with the exception of DNABERT-2 [37], though this dataset notably lacks genomes from the domain Archaea and consists of only 32 billion base pairs. To date, the largest DNA foundation model to be trained consists of 40 billion parameters [40], and was trained multi-species genomes and found to be successful at multiple downstream tasks. Genomic models augmented with epigenetic data have also demonstrated great success in downstream tasks such as predicting epigenetic markers [41–44], detecting splice sites and promoter regions [35], modeling the histone code [45], and modeling the phosphorylation of protein kinases [46]. Other foundation models focus on transcriptomics, primarily focusing on single-cell RNA (scRNA) [47–51]. Other foundation models for mRNA [52] and general RNA [53] have also been trained. Transcriptomic foundation models have successfully predicted transcriptome-to-proteome translations [54], gene rankings [55], cell type annotation [56], and drug response [51, 56].

However, there is a notable lack of work exploring the ability for transformers to model *multiple biosequences simultaneously*. Only two existing models incorporate both nucleic acid and protein information: AlphaFold3 [8], a closed-source proprietary model, and RosettaFoldNA [10]. Furthermore, both of these models are focused on structure predictions rather than generally learning from multi-omic sequences. We hypothesized that jointly modeling multi-omic sequences with a single shared model would allow for the exploration of the rich interaction between both sequence distributions as proteins and nucleic acids biologically have a complex but natural relationship, with proteins regulating nucleic acid expression through transcription factors, epigenetic modifications, and many other mechanisms, while nucleic acids regulate proteins via transcription/translation, small interfering RNAs (siRNA), micro-RNA, long non-coding RNA (lncRNA), and more. We trained several multi-omic models (MOMs) on a large corpus of nucleic acid and protein sequences. We name our trained foundation models OmniBioTE (omni-biosequence transformer encoders). We show that these models *emergently* learn rich joint representations of different sequence types, and that these representations are broadly useful for downstream structural and functional predictive tasks. We show that MOMs can be easily fine-tuned to predict the binding interactions of protein-nucleic acid complexes with high accuracy, as well as predict the effects of various mutations to the nucleic acid sequence on binding affinity, a critical task for designing nucleic acid aptamers or studying the impact of mutations on protein-nucleic acid binding. Remarkably, we show that MOMs naturally learn useful structural information when trained to predict the properties of

nucleotide-protein interactions, allowing for the identification of critical protein residues involved in the binding interaction exclusively from the primary sequences. We find that this structural information arises *without any a priori structural training*. Furthermore, we train identical models with the same compute budget on *only* nucleic acid or protein sequence, and show that there is little performance degradation from expanding the distribution to include multiple sequence types, suggesting a strong overlap in both distributions and a potential benefit to modeling both. We evaluate all models on previously benchmarked single-omic nucleic acid and protein tasks, achieving state-of-the-art results on a subset of nucleic acid tasks. An overview of the pretraining, fine-tuning, and exploration of our model can be found in Figure 1.

2 Results

We hypothesized that simply pre-training on large amounts of multi-omic data would lead to a joint representation space between modalities. We found that OmniBioTE learns joint representations between nucleic acids and protein sequences despite *never explicitly being taught these relations*. Next, we explored pre-trained OmniBioTE models' performance on downstream multi-omic tasks. Owing to the native multi-modality of OmniBioTE, OmniBioTE can be readily fine-tuned to predict the ΔG of previously unseen protein-nucleic acid interactions with high correlation to the ground truth values. Remarkably, we found that models trained to predict these physical quantities from primary nucleic acid and protein sequences learn structural information *emergently*, and encode this information within the models' attention maps. We further tested the robustness of our ΔG prediction models by predicting the change in binding affinity as a DNA-binding protein's motif is mutated. We then tested OmniBioTE's performance on another downstream multi-modal task by explicitly fine-tuning the model in a supervised fashion to predict which protein residues make contact with nucleotides given the protein sequence and nucleic acid sequence, with a contact defined at several thresholds (either less than 8Å, 6Å, or 4Å inter-residue-nucleotide distance). Lastly, we compared OmniBioTE models' performance on single-omics downstream tasks to identical models trained on only a single sequence modality with an identical compute budget to determine the performance change from training on multi-omic data. We found that training on multi-omic data results in increased performance on almost all downstream tasks.

2.1 Joint Sequence Representations Arise Emergently

We found that it is trivial to learn a low-rank projection that extracts joint features from the OmniBioTE embeddings and readily generalizes in contrast to the single-omic models (Figure 2.a-b). Despite OmniBioTE never being explicitly (or even implicitly) taught a correspondence between genes and their corresponding translated protein sequences, the model naturally learns these associations from the underlying distributions consistent with the central dogma of molecular biology. We hypothesize that this is due to the efficient coding hypothesis [57], where, if N protein sequences and their corresponding genes occur in the same dataset, it is considerably more efficient to memorize one of the modalities and learn the mapping between them, rather than memorize both sequences separately. This property holds at all scales, indicating that multi-omic training robustly learns joint representations for genes and protein sequences, even at low parameter counts.

2.2 MOMs natively perform multi-omic tasks

We demonstrated OmniBioTE's potential as a foundation model for natively multi-omic tasks by fine-tuning each OmniBioTE model to predict the ΔG of protein-nucleic acid binding interactions between both wild-type and mutant nucleic acid sequences. In these tasks, we found superior performance of OmniBioTE compared to recent, purpose-built, deep learning-based methods [58], likely owing to the rich sequence information gleaned from the large-scale multi-omic pretraining (Figure 3a). We compared

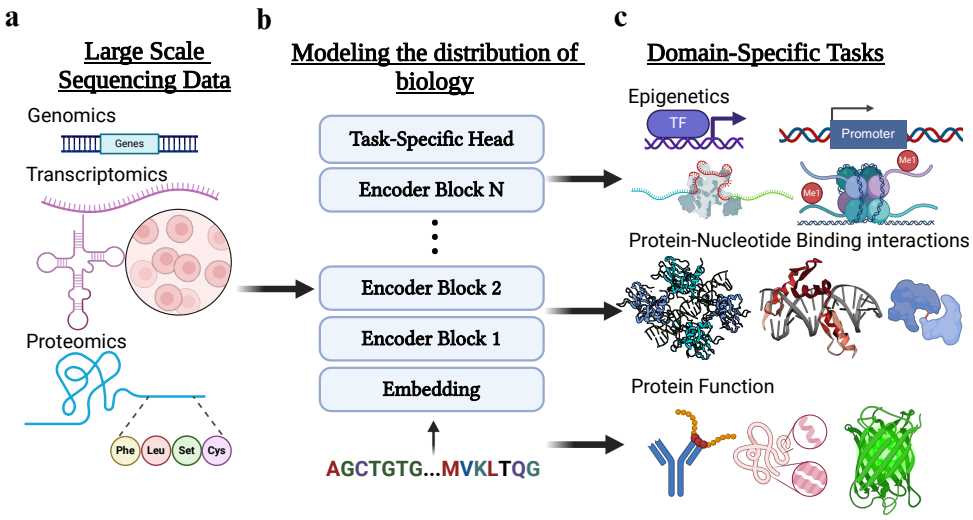

Figure 1: **The OmniBioTE Pipeline. a.** First, we gather large-scale datasets consisting of proteomic data, single-cell transcriptomics, and more. **b.** Next, we employ large-scale pertaining over these sequences via an encoder transformer and the masked language-modeling objective. **c.** Finally, we can finetune this foundation model with a task-specific head to tackle a wide variety of tasks.

our approach to using AlphaFold3 derived structures combined with molecular dynamics simulations. Notably, this task is notoriously difficult and computationally expensive to tackle via structural prediction followed by molecular simulation [59, 60]. We found that AlphaFold3 based simulationss were, as expected, notably more computationally intensive with worse results (Sec. S1, Extended Figure S1). We also note that there is an empirical maximum Pearson correlation and minimum mean absolute error achievable through gold-standard wet-lab measurement, as variations in techniques and experimental uncertainty produce different measured values across groups and labs. This empirical maximum Pearson correlation was estimated to be around 0.81, and the minimum absolute error was estimated to be around 0.6 kcal/mol [61].

We next confirmed that the multi-omic approach is considerably more performant and compute efficient than using two single-omic models (Figure 3a,b). We see a clear trend of increasing performance with model scale, as opposed to over-fitting with greater parameter count, indicating the robustness of the approach and potential for further performance gains with greater scale in both compute and data (Figure 3a,b). We found similar results on the contact prediction task, where performance, as measured in F1-score, improves with scale (Figure 3d).

On our held-out JASPAR binding motif evaluation, we found that all of our ΔG prediction models on average predict a decrease in binding affinity (increase in ΔG) as the strongest-binding motif is mutated. Additionally, we found that the average predicted decrease in binding affinity increases with scale (Figure 3c).

2.3 Attention maps encode structural information about protein-nucleic acid binding interactions

Remarkably, we found that OmniBioTE models finetuned to predict ΔG from *purely primary sequence data* naturally express structural information that emerges in the model's attention maps. The perfor-

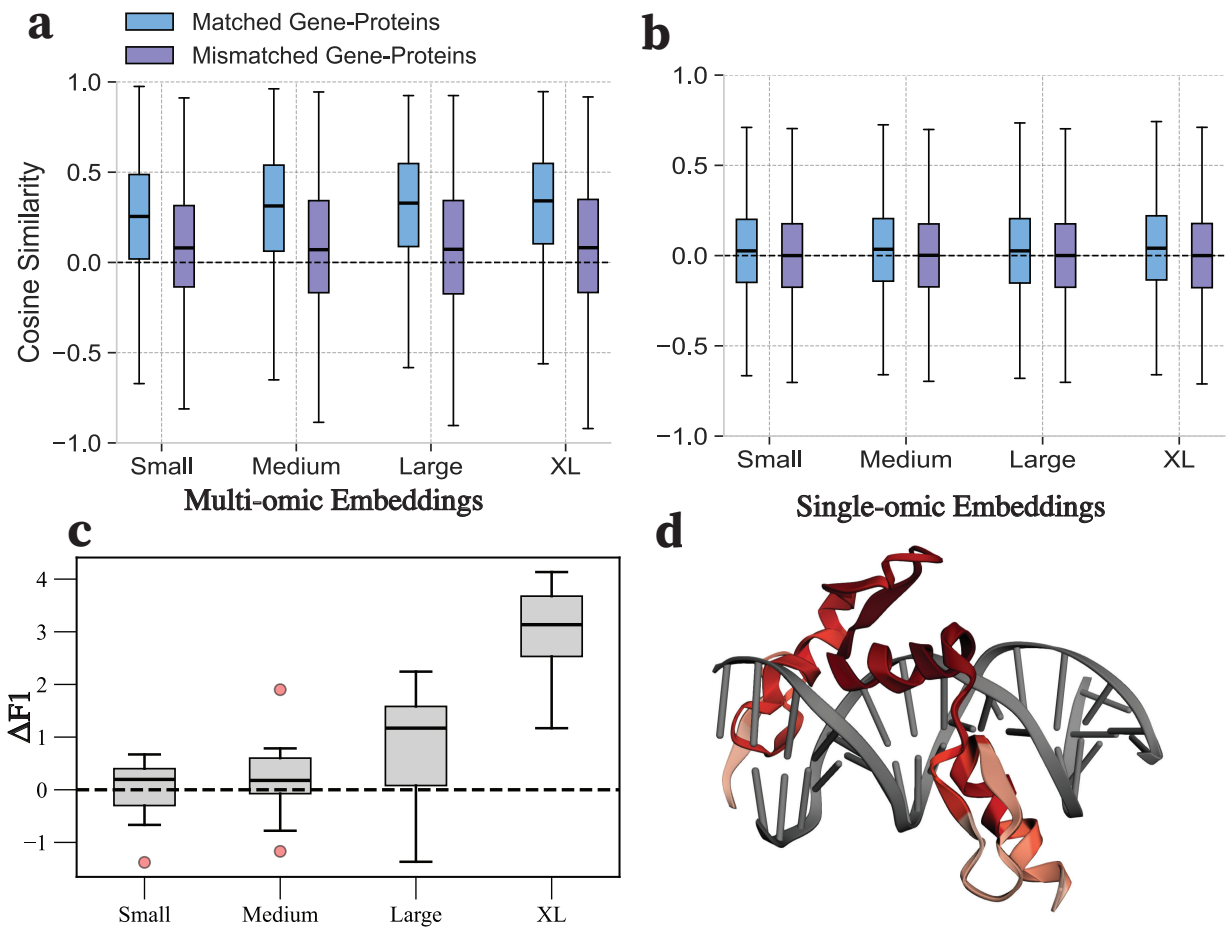

Figure 2: **a.** The distribution of cosine similarity between feature vectors produced by OmniBioTE via a low-rank feature extractor on the 98% held-out data. **b.** The analogous plot produced by NucBioTE and ProBioTE with two separate feature extractors. **c.** The increase in F1-score on the contact-prediction task using attention maps from OmniBioTE models fine-tuned to predict binding affinity compared to attention maps from the base models. **d.** An example of predicted contact probability for Zinc finger and BTB domain-containing protein 7A (ZBTB7A) bound to a DNA duplex computed from the attention maps produced by the fine-tuned OmniBioTE models.

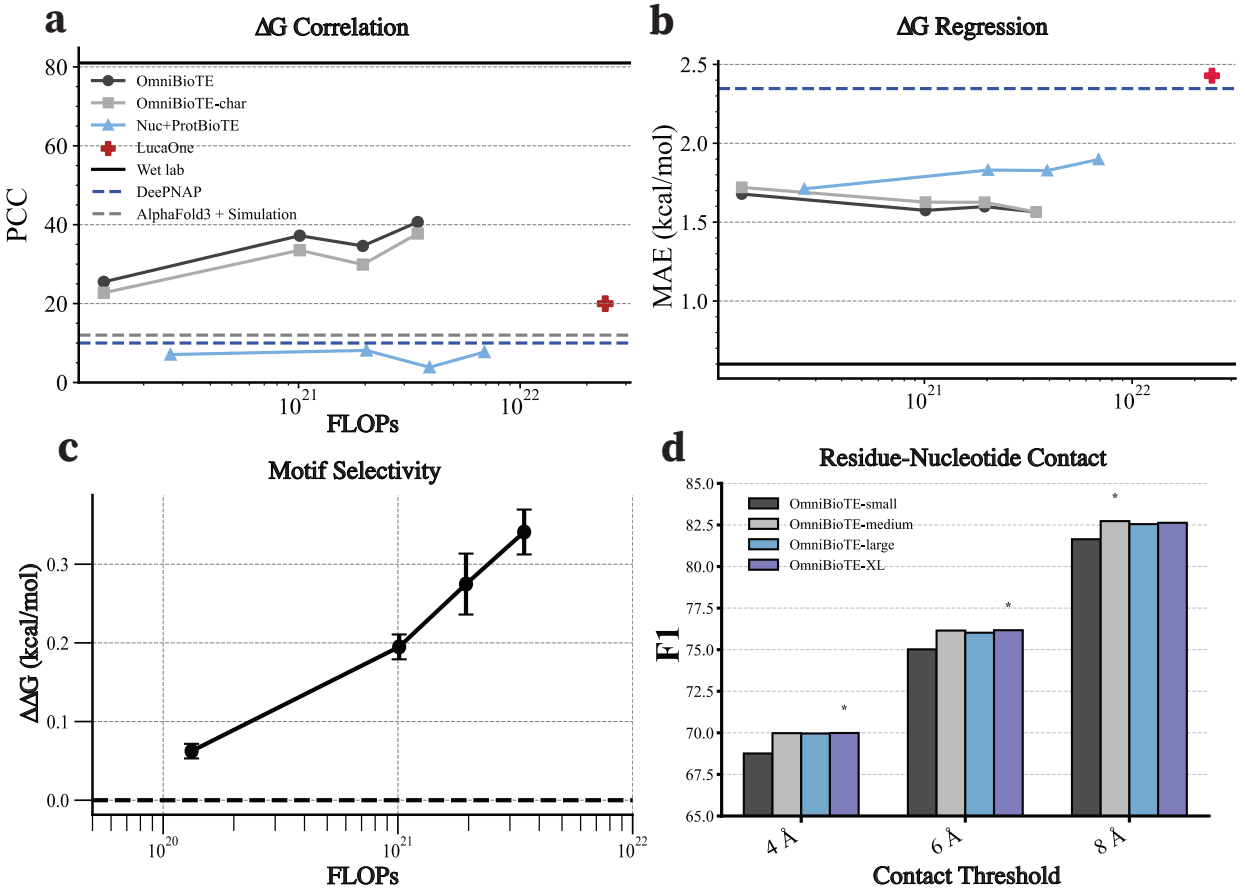

Figure 3: **a.** Performance on 10-fold cross-validation over the ProNAB dataset as a function of pre-training compute. The empirically maximum achievable correlation in a wet lab (due to inter-measurement error) is shown as the solid black line. **b.** Mean absolute error over the 10-fold cross-validation set. The empirically minimum achievable mean absolute error is shown as the solid black line. **c.** The predicted $\Delta\Delta G$ of mutated motifs as a function pre-training compute. **d.** Performance on the supervised PDB contact evaluation task trained at various contact thresholds across model scale. The positive-to-negative ratio of the dataset is 0.29, 0.16, 0.09, and the maximum F1-score achievable with random guessing is 37.0, 24.7, and 15.7, for 8Å, 6Å, and 4Å, respectively.

mance on the contact-prediction task, as measured in F1-score, increases when using attention maps produced by the fine-tuned OmniBioTE models compared to the base OmniBioTE models, and the disparity increases with scale (Figure 2c). This effect arises *without any a priori structural knowledge* during training. This finding further confirms that fine-tuned OmniBioTE models learn to compute binding affinity values through meaningful computational processes predicated on biological sequences as opposed to learning spurious correlations in the dataset.

2.4 Single-omics: Nucleic acid models

OmniBioTE exceeds performance on many downstream nucleic acid tasks compared to previous models trained exclusively on genomic data (Figure 4a,b). Notably, we find that multi-omic training achieves more favorable scaling-laws (performance-per-FLOP) than other training methodologies (Figure 4a,c,e). This demonstrates that models benefit from pre-training on multi-omic sequences.

OmniBioTE-XL also sets a new state-of-the-art for the prediction of epigenetic marks, core promoter detection, mouse transcription factor binding site prediction, promoter sequence detection, splice site detection, and SARS-CoV-2 variant classification. The full set of evaluation results can be found in Supplementary Tables S3, S4, S5, and S6.

2.5 Single-omics: Protein models

We find that OmniBioTE with the 2048 vocabulary tokenizer initially underperforms compared to its protein-only counterpart, though the per-residue model significantly outperforms this control (Figure 4c,e). Broadly, both OmniBioTE and the ProtBioTE control achieve performance comparable to previous machine-learning based methodologies when measuring performance-per-FLOP, though generally underperform in absolute performance compared to more recent methodologies that use considerably more training compute with large parameter counts, such as ESM-2 [13] which roughly used 12,000 A100 GPU days. We hypothesize this may be due to sub-optimal token budgets at larger parameter counts, causing the models to stray from the compute-performance Pareto-frontier. The full evaluation results for both ProteinGLUE and TAPE can be found in the Supplementary Tables S7, S8, S9, S10, S11).

3 Discussion

We developed OmniBioTE, a series of first-of-its-kind multi-omic models (MOMs), and analyzed their properties across a wide range of scales and tasks. We demonstrated the unique potential of MOMs for modeling protein-nucleic acid interactions by fine-tuning OmniBioTE to achieve state-of-the-art performance on the task of predicting the change in binding energy between a protein and nucleic acid. We also showed that as a result of this fine-tuning process, OmniBioTE learns meaningful structural information, allowing one to estimate how strongly a given protein residue or nucleotide participates in binding interactions.

We found that OmniBioTE emergently learned a joint representation space between nucleic acid and protein sequences despite never explicitly being trained on a joint objective, demonstrating that training biosequence transformer models on multi-omic data can learn non-trivial representations across sequences even with a simple masked language model objective. We attribute this emergence from self-supervised pre-training as being a consequence of the efficient coding hypothesis [57]. We hypothesize that considerably richer representations could be learned if auxiliary training objectives were introduced, such as structure/property prediction, cross-attention between different modalities, or the addition of multiple sequence alignment data. Beyond additional learning objectives, we note that there has been a considerable amount of research into multi-modal vision-language modeling using novel model architectural components including cross-attention and cross-modal projectors [5, 62–64], and that many of these approaches may be of interesting in multi-modal biosequence modeling as well.

We additionally found that an added benefit of multi-omic training is that MOMs are superior at scale to similar models trained on single-omics data

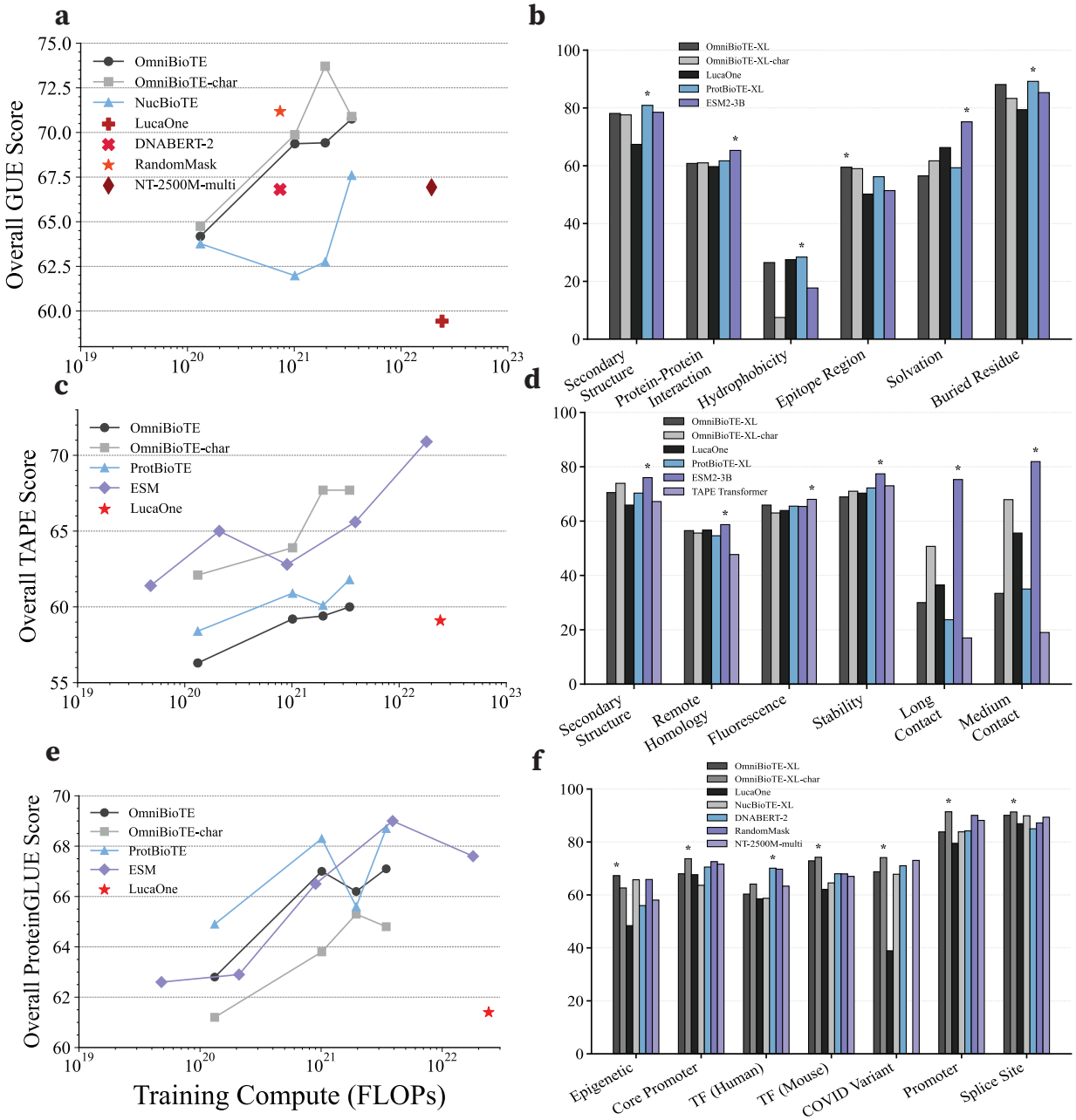

Figure 4: **Model performance and scaling across single-omic benchmarks. a,c,e.** Aggregate benchmark performance for each task plotted against pre-training FLOPs. **b,d,f.** Comparison of top-performing models on individual sub-task performance for each benchmark.

with identical architectures and compute budgets. Despite the difference in datasets, we found no downsides to mixing in other modalities during pretraining for our biosequence foundation models in this project. In fact, our MOMs set new state-of-the-art performance numbers for several of the downstream nucleic acid tasks. Our MOMs also considerably outperformed a combination of single-omic models on the multi-omic task of binding affinity prediction, and significantly outperformed molecular dynamics methods in conjunction with structural predictions from AlphaFold3, despite being a considerably more computationally intensive baseline. Lastly, we showed that these results robustly transfer to completely unseen and unrelated datasets by testing our models on the JASPAR dataset.

There are several notable limitations to this work that deserve special mention. Most notably, we only scratched the surface on multi-omic biosequence modeling. As noted earlier, there are many popular ways of training multi-omic sequence models, and we elected for a simple approach using a masked language modeling task. We additionally only investigate our scaling over a rough two orders of magnitude of compute, and leave the training of larger models on larger datasets as future research directions that seem reasonably likely to yield performance benefits consistent with the scaling results we found in this work. Lastly, we only investigated a masked language modeling task for pre-training rather than the more popular autoregressive training framework, again leaving this approach open as a viable future research direction.

Many of biology's most significant interactions occur between proteins and DNA, and we demonstrate the first large-scale attempt at building and scaling foundation models to specifically learn these critical molecular interactions. Beyond their biological significance, modeling the interactions between nucleic acids and proteins is of great pharmaceutical and clinical importance; models that can assist with the development of nucleic acids that modify the function of naturally occurring proteins would greatly accelerate pharmaceutical development. Many notable pharmaceutical drugs and candidate drugs that function via nucleic acid-protein interaction have already shown great promise, such as pegaptanib [65], an RNA aptamer targeting vascular endothelial growth factor, as well as RNA sequences that target nucleolin [66], coagulation factors [67–70], CCL2 [71], CXCL12 [72], and hepticidin [73]. Foundational biosequence models have the promise of dramatically improving our ability to both understand and predict biology, and we hope that our work with OmniBioTE presents the first of many efforts to build multi-omic models that can capture the full richness of biomolecular interactions.

4 Methods

Broadly, we train dense, non-causal, encoder transformer models of varying sizes using the masked-language-modeling (MLM) objective [3] on 250 billion tokens of nucleic acid and protein sequences of varying types. We additionally train control models consisting of only nucleic acid or protein sequences with equal compute budgets to evaluate the effect of training on additional sequence types. We demonstrate that our MOMs *emergently* learn joint representations between nucleic acid and protein sequences by showing that there exist meaningful features roughly invariant to sequence modality, and that such features *do not* exist in single-omic models.

We evaluate our suite of models by fine-tuning on several single-omics datasets that assess performance on various downstream tasks relevant to molecular biology, structural biology, and biochemistry. Additionally, we design two novel multi-omic tasks that require inference on both protein and nucleotide sequences simultaneously. Lastly, we show that through inference upon the models' attention maps, structural information is learned without any *a priori* structural training.

4.1 Training Data

We source our nucleic acid data from GenBank [74], a collection compiled by the National Center for Biotechnology Information. We preprocessed the entire GenBank archive by first removing all metadata from each sequence, with the exception of sequence

type (DNA, mRNA, tRNA, etc.). This produced 242,855,368 sequences with a total of 312,190,748,151 base pairs, primarily composed of general DNA, general RNA, mRNA, cRNA, and single-stranded RNA. A full breakdown of nucleic acid sequence data can be seen in Table 1. We source our protein data from Uniref100 [75], a dataset maintained by UniProt. Similarly to the nucleic acid data, we remove all metadata from each sequence, yielding 369,597,671 sequences with a total of 1,739,747,047 residues.

We take a subset of 10^{11} base pairs and protein residues total to train a byte-pair encoding tokenizer [76] using the Sentencepiece library [77], with a vocabulary size of 2^{11} for protein sequences and nucleic acid sequences (2^{12} unique tokens total). Our choice of tokenizer and vocabulary size was chosen based on previous work [37]. Additionally, we train a multi-omic per-residue/nucleotide model at each size, where each token is simply a single base pair or residue.

4.2 Architecture and Training

OmniBioTE is based on the GPT-2 architecture [78] and the LLaMA-2 architecture [79]. We substitute learned positional embeddings [80] for rotary positional embeddings (RoPE) [81] and replace the causal self-attention mechanism [78, 80] with a full, non-causal attention operation [3]. We additionally scale the pre-softmax causal-attention at 1/width rather than 1/width^2 in accordance with maximal update parameterization (μP) [82]. We use an aspect ratio (the ratio of model width to depth) of 128. We modify Kaparthy's NanoGPT [83] for a lightweight and simple model implementation. We train four OmniBioTE variants, OmniBioTE-small (88 million non-embedding parameters), OmniBioTE-medium (675 million), OmniBioTE-large (1.3 billion) and OmniBioTE-XL (2.3 billion). Additionally, we train controls for each model on only nucleic acid data or only protein data (henceforth referred to as "NucBioTE-[size]" and "ProtBioTE-[size]"). For experiments requiring fine-grained, single-nucleotide/residue inference, we also train an OmniBioTE model of each size that uses a single-character tokenizer rather than a byte-pair encoding (BPE). In total, we train 16 models.

We train each model for 250 billion tokens with a context length of 1024 tokens for the BPE-tokenized models and a context length of 2048 characters for the single-character models (to accommodate for the decreased amount of data per token). We train at a batch size of 786432, 1032192, or 1048576 tokens (chosen based on available compute and memory and to maximize throughput) with the masked language modeling objective [3]. We use AdamW [84] ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$, weight decay $= 10^{-2}$), employing μP for stable hyperparameter transfer. For the parameters with fixed learning rate under μP (the embedding and unembedding parameters), we set the learning rate to 0.05, and scale learning rates of the rest of the parameters via 32/width. These hyperparameters were determined empirically with sweeps at the 10^6-parameter-scale. Finally, all learning rates are decayed with PyTorch's OneCycleLR [85], with a warmup period of 1 billion tokens, a starting and ending learning rate scale of 10^{-5}.

4.3 Evaluations

We design our own multi-omic benchmark to assess our model's ability to accurately characterize protein-nucleic acid interactions. We further design several novel benchmarks to assess the performance and interpretability of our models on protein-nucleic acid tasks. In addition to our main multi-omic tasks, we evaluate our approach on several popular benchmarks to evaluate single-omic performance on a variety of nucleic acid and protein-based tasks in an effort to assess the baseline single-omic capabilities of our model before multi-omic task-specific fine-tuning. All fine-tuning optimization is performed via AdamW [84] with identical hyperparameters as described in the pre-training step unless otherwise specified.

4.3.1 Protein-Nucleic Acid Binding Evaluation

To showcase the native multimodality of our generalist model, we designed a novel evaluation task using the ProNAB dataset [86]. ProNAB consists of 20,090 samples comprised of 14606 protein-DNA complexes,

5323 protein-RNA complexes, and 161 protein-DNA-RNA complexes. These samples are composed of 798 unique DNA-binding proteins and 340 unique RNA-binding proteins. We refer to the original work for a detailed description of the dataset composition [86]. The objective of our task is as follows: given the primary sequence of a nucleic acid-binding protein, a wild-type nucleic acid sequence, and a mutant nucleic acid sequence, predict the ΔG of both the protein-wild and protein-mutant nucleic acid complex. This task is of particular interest in the prediction of unknown DNA/RNA-binding protein interactions with the human genome.

We assemble our dataset by first filtering the ProNAB dataset, rejecting any nucleic acid or protein sequences with non-standard residues (we use only the standard 20 amino acids and the 5 standard nucleotide bases), leaving 850 unique proteins, and 15994 protein-nucleic acid complexes. We then split the data into 10 cross-validation sets. Ultimately, we end up with 752 unique proteins and 12282 total protein-nucleic acid interactions.

The ProNAB dataset often has multiple nucleic acid sequences per protein, thus the number of unique proteins is vastly outweighed by the number of unique nucleic acids. To avoid data leakage in the train and test sets, we group samples by protein sequence, then create folds by randomly grouping by protein sequence such that the folds do not have any proteins in common. Furthermore, we conduct sequence similarity analysis on the protein sequences in the train and test set via sequence alignment with the BLOSUM62 substitution matrix [87] to ensure minimal train/test leakage. We found that the average alignment score between identical protein sequences in our dataset was 5.20 ± 0.15 (identical sequences may have different scores due to the BLOSUM62 scores), while over 99.4% of pair-wise comparisons in our train/test set had an alignment score below 0.0, and 99.9% had a score below 1.0 suggesting that our results are not purely a result of sequence homology. As an extra precaution, we keep any proteins that have a sequence similarity score over 1.5 *with any other protein sequence in the dataset* strictly in the train set of all cross-validation sets to guarantee there is no significant sequence homology in any cross-validation fold. As a result, 13 unique proteins and 232 protein-nucleic acid interactions were always kept in the train set to avoid any significant sequence homology in the validation sets.

To compute a ΔG value, we first concatenate a primary protein sequence and nucleic acid sequence pair and run a forward pass through OmniBioTE. We then take the embedding produced by the first token and apply a linear projection which produces a single ΔG value. If a complex is composed of a protein and a double-stranded DNA or RNA molecule, we append the second nucleic acid sequence as well. We finetune our model to predict ΔG from the protein-nucleic acid pairs in the train set, with mean-squared error (MSE) as our loss target. As a single-omic control, we compute the embeddings of the protein and nucleic acid sequences separately with the corresponding ProBioTE and NucBioTE model. We then concatenate these embeddings and use a linear projection head to produce the ΔG value.

Our primary evaluation metrics are the Pearson correlation coefficient of ΔG prediction with the ground-truth measured value, as well as the mean absolute error of the predicted ΔG values. We begin with a pretrained OmniBioTE model, then further train our models for 64 epochs with a batch size of 256 on the wild-type ΔG prediction task. The projection head learning rate initialized to 10^{-2}, the embedding vector learning rate initialized to 10^{-3}, and the non-embedding parameters learning rate to $10^{-4} \cdot 1024/\text{width}$. All learning rates are decayed with PyTorch's OneCycleLR, an implementation of the learning rate schedule first described in [85].

As a baseline, we train a recent deep-learning-based architecture, DeePNAP [88] on the identical cross-validation dataset as our model. We train the DeePNAP architecture for 64 epochs with a batch size of 256. For the training, we use AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, weight decay $= 10^{-2}$, weight decay $= 10^{-2}$), starting at a learning rate of 10^{-3} and decaying linearly to 0.0. Additionally, we fine-tune a recently released DNA-Protein model, LucaOne [89] in a similar manner. Specifically, we set the embedding learning rate to 10^{-4}, the non-embedding parameter learning rates to $2.5 \cdot 10^{-5}$, and the projection head learning rate to 10^{-2}. We train

the LucaOne with identical AdamW hyperparameters, batch size, and epochs.

Lastly, we compare against a baseline that is more representative of current computational methods. First, we predict the structure of the protein-nucleic acid complex with AlphaFold3 [8] and use molecular dynamics simulations to predict the ΔG of the binding interaction.

4.3.2 Binding Motif Specificity

To further validate the robustness of the OmniBioTE models fine-tuned to predict binding affinity, we evaluate whether the models can correctly predict the specificity of various DNA-binding proteins (DBPs) to their motifs. First, we gather a set of 2,145 DBPs and their position-frequency matrices (PFMs) from JASPAR [90]. Using the same sequence similarity rejection technique described in the ProNAB experiment, we filter all DBPs from the JASPAR dataset that have any significant overlap with the ProNAB dataset used in the cross-validation evaluation. We then use our fine-tuned OmniBioTE model to compute the ΔG for each DBP-motif pair, where the motif is computed via the most frequent nucleotide in each position of the PFM. Next, we mutate each motif by randomly substituting each nucleotide with probability 5%. This produces a mutated motif that would have a reduced binding affinity to the DBP as empirically known by the PFM, but would still be "in distribution" of the plausible binding motifs. We generate 8 unique mutated motifs per DBP-nucleic acid pair. We predict the ΔG for these mutated interactions and compute the difference between the predicted ΔG of the highest-binding-affinity motif. If the model has learned to model the binding interaction correctly, we should expect the ΔG to increase after the motif is mutated.

4.3.3 Protein-Nucleotide Contact Prediction

We gather all structures from the Research Collaboratory for Structural Bioinformatics Protein Data Bank [91] that contain strictly one protein chain and either one or two nucleic acid chains. For each residue in the protein-nucleic acid complex, we compute the distance to the nearest nucleotide and label a residue as "contacting a nucleotide" if it is within 8Å of a nucleotide. Next, we group data by primary protein sequence and create 10 cross-validation splits by protein grouping to avoid data-leakage. To fine-tune OmniBioTE, we concatenate the protein and nucleic acid sequences together and compute a forward pass through the model as usual. Instead of un-embedding the hidden states of the final layers, we instead compute a linear projection to a single scalar, upon which a sigmoid function is applied to yield a contact prediction. Although the nucleic acid sequence is included in the forward pass, contact prediction is only computed for the protein residues. We train the model against the binary cross-entropy loss function for 32 epochs on each fold with a batch size of 256, with an identical training setup to the runs in the protein-nucleic acid binding evaluation. We additionally test other contact thresholds (4Å and 6Å) to evaluate the robustness of our approach.

4.3.4 Genome Understanding Evaluation

To evaluate OmniBioTE's generalizability to a variety of domain-specific nucleic acid tasks, we employ the Genome Understanding Evaluation (GUE) suite [37]. GUE consists of several genetic and epigenetic classification tasks over human, mouse, yeast, and coronaviridae genomes. Core promoter detection, transcription factor prediction, promoter detection, splice site detection, epigenetic mark prediction, and COVID variant classification were the target classes among these genomes. The promoter detection task is a binary classification task, where the goal is to determine whether a sequence of DNA is or is not a promoter. The promoter task is divided into several subcategories: proximal promoter detection, core promoter detection, and TATA/non-TATA motif promoter detection. The proximal promoter task contain the entire promoter sequence (including the core promoter) in the classification task, while the core promoter task only includes the sequence in close proximity to the transcription start site. The TATA class is composed of promoters that contain a TATA-motif, while the non-TATA does not have a TATA motif. Transcription factor detection is an-

other binary classification task, where the goal is to determine whether a DNA sequence is the binding site of a transcription factor. This task is divided into human and murine datasets. Splice site detection is a classification task where the goal is to determine if a DNA sequence contains a splice donor or acceptor site. The epigenetic tasks' goals are to determine whether a nucleic acid sequence taken from a yeast genome is likely to contain a given epigenetic modification Lastly, the COVID variant task is a multi-class classification task where the goal is to predict which variant-type (Alpha, Beta, Delta, Eta, Gamma, Iota, Kappa, Lambda and Zeta) a 1000 base pair snippet was sequenced from. We refer to the original work for a full characterization of the evaluation set. All tasks use Matthews correlation coefficient as the primary metric, with the exception of the COVID variant classification task, which uses F1-score.

For each classification task, we fine-tune a base OmniBioTE or NucBioTE model. A class prediction is generated by taking the first token's final embedding and applying a linear projection down to the number of classes in place of the original final projection head, followed by a SoftMax operation. We set the embedding parameter learning rate to 10^{-3}, the transformer weight matrices to $1024 \cdot (\text{model width})^{-1} \cdot 10^{-4}$, and lastly, set the learning rate of the projection head to 10^{-2} for all model sizes. Hyperparameters were determined with sweeps over the validation sets. All learning rates are decayed with PyTorch's OneCycleLR. The small and medium models are trained for 15000 steps with a batch size of 32 over the training data, while the large and XL models were trained for 30000 steps with a batch size of 32. We find that final validation performance is relatively robust to the number of epochs over each dataset, thus these training parameters were chosen to yield a reasonable training time. The model that performs best on the validation set is evaluated on the test set. We additionally fine-tune LucaOne as an additional multi-omic baseline. We train with the exact same optimizer hyperparameters described for LucaOne in the protein-nucleic acid binding evaluation above. We train with batch size 32 for 30,000 iterations on each task.

4.3.5 Tasks Assessing Protein Embeddings

We employ the Tasks Assessing Protein Embeddings (TAPE) suite [92] to evaluate OmniBioTE's ability to generalize to unseen protein-based tasks. TAPE consists of five challenges: secondary structure prediction, residue contact prediction, remote homology detection, fluorescence prediction, and stability prediction. Secondary structure prediction is a per-residue classification challenge, where the goal is to determine what type of secondary structure each residue composes. The secondary structures are split into one of either 3 or 8 classes, depending on the task. Residue contact prediction involves generating an $N \times N$ mask, where N is the length of the protein, with each element of the mask predicting the probability that a residue pair are within 8 Å of each other. Remote homology detection involves mapping a primary protein sequence to one of 1195 homologies, with the aim to learn to classify primary sequences into meaningful structural families. Fluorescence prediction is a regression task, where the goal is to predict the log fluorescence intensity of a protein from a given primary structure. Finally, stability prediction is a regression task that aims to predict the maximum concentration at which a protein is still structurally stable. All classification tasks are measured in accuracy, while all regression tasks are measured via Spearman's correlation coefficient. We train each task (excluding the contact evaluation which is discussed below) for 64 epochs over the dataset with a batch size of 32, with identical initial learning rate parameters and schedule as the GUE tasks [37], though we initialize the non-embedding model parameter learning rate to $1024 \cdot (\text{model width})^{-1} \cdot 10^{-4}$, the embedding learning rate to 10^{-4}, and the projection head learning rate to 10^{-2} for all model sizes.

The residue contact evaluation task involves predicting an $L \times L$ matrix of values between 0 and 1, with each element (i, j) representing the probability that residue i in the primary sequence is within 8 Å of residue j. To generate this prediction matrix, embeddings are generated from a transformer model [80], and a learned linear projection head transforms each embedding into 128-dimensional vectors. As inspired by previous work [93], a tensor of shape

$256 \times L \times L$ is constructed, where item $[:, i, j]$ corresponds to the i^{th} 128-dimensional vector concatenated with the j^{th} 128-dimensional vector. This tensor is transformed via an 8-layer ResNet [94] to yield a final $(1 \times L \times L)$ matrix, which after transformation by the sigmoid function, produces the desired probability matrix. Binary cross-entropy is used as the loss target, with masks applied computing the loss only on residue pairs that are separated by at least 12 total residues (excluding "short" contacts). Fine-tuning is performed for 128 epochs with a batch size of 128. The learning rate of non-embedding transformer parameters was set to $1024 \cdot (\text{model width})^{-1} \cdot 10^{-4}$, with the projection head and ResNet [94] using a learning rate of 10^{-3}. Learning rates were warmed up and decayed via the PyTorch OneCycleLR [85] learning rate scheduler as mentioned previously.

We fine-tune a series of ESM2 models [13] to compare both absolute performance and scaling performance against a state-of-the-art single-omic protein model. Specifically we finetune the 8 million, 35 million, 150 million, 650 million, and 3 billion parameter ESM2 models in an identical fashion as the OmniBioTE models above. For brevity, we hereafter refer to the ESM models as ESM2-XS (8 million), ESM2-S (35 million), ESM2-M (150 million), ESM2-L (650 million), and ESM2-XL (3 billion). We use the same embedding and head learning rate as the OmniBioTE finetuning runs, and set the non-embedding parameter learning rate to $640 \cdot (\text{model width})^{-1} \cdot 10^{-4}$. Additionally, we evaluate LucaOne via the same hyperparameters described in the protein-nucleic acid binding evaluation, with the same number of iterations and batch size for each task.We use AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, weight decay $= 0.01$) as the optimizer for all models.

4.3.6 Protein General Language of Life Evaluation

To explore per-residue tasks (i.e., tasks that require a prediction for every residue in the protein), we employ the Protein General Language of Life Evaluation (ProteinGLUE) [95]. We refer to the original work for a full description of ProteinGLUE, but briefly, ProteinGLUE consists of several tasks:

Secondary structure prediction: the task is identical to the TAPE secondary structure task discussed above [92]. Accuracy is the primary metric.

Solvent accessibility: the task is to either classify whether a residue has less than 7% solvent accessibility, as well as a regression task to predict the actual solvent accessibility value. For the binary classification task, accuracy is the primary metric, and Pearson correlation coefficient is used as the primary metric for the regression task.

Protein-protein interaction: the task is to predict which residues interact in either homodimer or heterodimers. Area under the receiver operating characteristic curve (AUCROC) is used as the primary metric.

Epitope region detection: the task is to predict which regions of a protein are antigenic epitopes. The performance of this task is measured in AUCROC.

Hydrophobic patch prediction: the goal of this task is to predict the largest rank of a hydrophobic patch that a residue belongs to. This task is measured via Pearson correlation coefficient.

Each task was trained with a batch size of 32 for 16 epochs on all tasks except for the protein-protein interaction, for which 64 epochs were used owing to a smaller dataset size. Identical initial learning rates and schedules used in the TAPE evaluation mentioned above were used. We compare against ESM models in a similar manner as the TAPE evaluations, namely with an embedding learning rate of 10^{-4}, a projection head learning rate of 10^{-2}, and a non-embedding parameter learning rate of $640 \cdot (\text{model width})^{-1} \cdot 10^{-4}$. We use the same optimizers and hyperparameters as described in the TAPE evaluations. We evaluate LucaOne on this task with identical hyperparameters as the TAPE evaluation.

4.4 Per-Residue Evaluations

Because the protein and nucleic acid datasets were tokenized with byte-pair encoding [76], most tokens contain several base pairs or residues. Evaluations that require a per-residue prediction, such as secondary structure, are not directly compatible with this tokenization scheme. To solve this issue, we ap-

ply two simple strategies at train and test time. At train time, we compute the target of a single token as the mode of all the residues it contains in the case of a classification task or the mean of the values of the residues it contains in the case of a regression task. This allows the input sequence length and the target sequence length to be the same size. At test time, we simply duplicate the value at the predicted token by the number of residues that token contains, allowing us to construct a prediction with the same length as the target ground truth. This method places an upper bound on the maximum achievable performance our model can achieve on any per-residue task, but in practice, this upper bound is higher than state-of-the-art results previously reported. This is likely due to the fact that nearby residues often share similar values in per-residue prediction tasks (e.g., if a residue is in a beta chain, its adjacent residues are likely to be in a beta chain as well). We note that our evaluation results are still directly comparable to previous per-residue methods, as we duplicate our predictions to match the ground truth dimensionality rather than decreasing the ground truth dimensionality to match the sequence length (as is done at train time).

For the contact evaluations, the non-uniform number of residues encoded by each token presented a significant challenge. We remedy this issue by transforming prediction targets from residue to token space for training and transforming predictions from token to residue space for evaluation. Transformation of prediction maps from residue space to token space was accomplished by assigning the (i, j)-token pair as a true contact if *any* of the residues contained within token i contact *any* of the residues within token j. Similarly, the (i, j)-token pair of the contact mask, used to ignore short-range contacts in the loss function, was assigned a positive value if any of the residues contained within token i are at least 12 residues apart from any of the residues contained in token j. Transforming from token space to residue space for evaluation is done in a simpler manner: residue (n, m) is assigned the value of the token pair (i, j), where i is the token containing residue n and j is the token containing residue j. For the per-residue/nucleotide models, the models were evaluated normally.

4.5 Interpretability

4.5.1 Protein-Nucleic Acid Interactions

To show that OmniBioTE learns semantically meaningful features, we demonstrate that when trained to predict the binding affinity between a nucleic acid and a protein sequence, OmniBioTE implicitly learns structural information despite exclusively being trained on primary sequence data. We fine-tune one OmniBioTE model of each size, in an identical fashion as described for the protein-nucleic acid binding evaluation, though we use all available data rather than cross-validation splits, as the goal is to fine-tune OmniBioTE models to be highly capable of predicting binding interactions, then investigate their mechanics.

Next, we gather all structures from the Research Collaboratory for Structural Bioinformatics Protein Data Bank [91] that contain strictly one protein chain and either one or two nucleic acid chains. For each residue in the protein-nucleic acid complex, we classify the residue as making contact with a nucleotide if it is within 8Å of any nucleotide (in the same manner as described in the Protein-nucleic acid Contact Prediction task). We then compute a forward pass through either the OmniBioTE model fine-tuned to predict ΔG or through the base OmniBioTE model (control) and collect the attention maps produced by each head in each layer (this results in N^2 attention maps, where N is the number of layers). Next, we concatenate these attention maps along the channel dimension to produce an $N^2 \times L \times L$ tensor, where L is the length of the input sequence. We then train a small convolutional network consisting of four layers. The first layer takes the N^2 channels and applies a 3×3 convolution to produce 64 channels, the next two layers apply a 3×3 convolution producing 64 channels, and the final layer again applies a 3×3 convolution but produces only one channel. The output of the convolutional net is an $L \times L$ tensor, and we average across the last dimension to produce L logits that, after a sigmoid operation, yield the predicted probability that a given residue makes contact with a nucleotide (this task is identical to the Protein-Nucleic acid Contact Prediction task described above). We

train this convolutional network via AdamW with a learning rate of 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay of 10^{-2}, and $\epsilon = 10^{-8}$ for 1000 steps with a batch size of 256, linearly decaying the learning rate to zero over the course of training. Critically, *the weights of the underlying OmniBioTE model remain frozen throughout training*, meaning that the convolutional network must extract this structural information strictly from the attention maps produced by the underlying model. We compare the F1-score on each of the 10 folds for the attention maps produced by the base OmniBioTE model and those produced by the OmniBioTE model fine-tuned to predict binding affinity. If the fine-tuned model has learned meaningful structural information from the fine-tuning process, we would expect the F1-score for convolutional networks trained on these attention maps to be higher than those of the base model.

4.5.2 Shared Representations Between Modalities

We aim to test whether OmniBioTE effectively learns a joint representation space between nucleic acid and protein sequences rather than simply learning to represent both modalities separately. In this case, we want to test whether OmniBioTE has learned representations of gene sequences (DNA) and their corresponding protein sequences that reflect shared functional or structural properties, independent of the sequence modality.

We first formalize the notion of invariance under transcription and translation. Let $x \in X$ be a gene (DNA) sequence, and let $y \in Y$ be the corresponding protein sequence produced by a mapping $G : X \to Y$, such as the standard transcription and translation process. Suppose that our pre-trained multimodal model outputs embeddings $\mathbf{z}_x$ for x and $\mathbf{z}_y$ for y, where $\mathbf{z}_x, \mathbf{z}_y \in \mathbb{R}^d$. We define a feature extractor $\phi : \mathbb{R}^d \to \mathbb{R}$ that maps an embedding to a scalar feature value. A feature is called *invariant* under the mapping G if

$$\phi(\mathbf{z}_x) = \phi(\mathbf{z}_y)$$

for all $x \in X$ and $y = G(x)$. In practical terms, such an invariant feature may correspond to the biological function or identity of a gene–protein pair, that is, a characteristic that remains constant regardless of the modality.

To test whether the model has indeed learned such invariant features, we conduct a contrastive learning experiment employing a strict linear transformation. In this experiment, we first obtain pairs of gene sequences (including both intronic and exonic regions) and their corresponding translated protein sequences. Using our pre-trained multimodal model, we compute the embeddings $\mathbf{z}_x$ and $\mathbf{z}_y$ for each gene and protein sequence, respectively. We then introduce a learnable linear function $W \in \mathbb{R}^{k \times d}$ with low rank $k \ll d$ to project the embeddings into a shared subspace, yielding $W\mathbf{z}_x$ and $W\mathbf{z}_y$. The function W is optimized via a contrastive objective that simultaneously maximizes the cosine similarity between corresponding pairs $W\mathbf{z}_x$ and $W\mathbf{z}_y$ while minimizing the similarity between non-corresponding pairs.

Specifically, we employ a contrastive loss function similar to the CLIP framework [96] to learn our feature extractor: let $X \in \mathbb{R}^{N \times d}$ and $Y \in \mathbb{R}^{N \times d}$ denote two batches of embeddings (with N samples and embedding dimension d), where each row x_i of X is a gene's feature vector,a and each row y_i of Y is the corresponding protein sequence. Any given pair x_i and y_j are unrelated if $i \neq j$. To compute the contrastive loss, each embedding in X and Y is normalized to unit length. The normalized embeddings are then used to compute a similarity matrix $S \in \mathbb{R}^{N \times N}$ whose entries are given by

$$S_{ij} = \frac{\langle \hat{x}_i, \hat{y}_j \rangle}{\tau},$$

where τ is a temperature parameter that controls the scaling of the cosine similarities.

In this setup, the diagonal elements S_{ii} represent the cosine similarity between corresponding pairs, while the off-diagonal elements S_{ij} for $i \neq j$ represent the similarities between non-corresponding pairs. Our final loss is composed of two terms: the first term considers each row of S as logits for a classification task in which the correct label for x_i is i. The second term is computed by treating each column as logits for the corresponding y_i. The two terms are simply averaged to compute the final scalar loss. This approach is identical to the original CLIP loss proposed

by Radford et al. [96]. For our experiments, we use $\tau = 0.07$, and $d = 16$.

We minimize this loss via the AdamW optimizer, with learning rate 0.01, linearly decayed to 0.0 over 10000 steps, $\beta = (0.9, 0.95)$, and $\epsilon = 10^{-8}$. *We optimize strictly over the projection matrix and leave the model parameters frozen*, as the goal is to test whether joint features are already learned, not whether they *can* be learned.

After learning ϕ, we apply this transformation to a held-out set of gene-protein pairs and compute the dot product between their feature representations. If ϕ is a generalizable feature extractor, we should see high dot product scores between corresponding held-out pairs and low dot product scores between non-corresponding held-out pairs.

Critically, we assess the generalization capability of the invariant features under very strict conditions; we train on only 2% of the available paired data and test on the remaining 98%. Strong performance in this setting indicates that the model's embeddings encode a shared subspace that captures the desired invariances.

For further validation, we perform a control experiment using two separately trained single-omic models—one trained solely on genes and the other solely on proteins. In this case, the embedding spaces of these models are learned independently, and there is no inherent guarantee of alignment between them. We attempt to learn two distinct feature extractors, ϕ_x and ϕ_y, for the gene and protein modalities, respectively, with the goal of minimizing the same contrastive loss.

5 Acknowledgements

The authors would like to thank Michael Retchin for his insightful comments and broad literature knowledge on protein-nucleic acid interaction. The authors would like to thank Douglas Kondziolka for his feedback on the manuscript. The authors would also like to thank Vincent D'Anniballe for his helpful discussion surrounding biosequence datasets. Lastly we would like to thank Michael Costantino and the NYU Langone High Performance Computing team for their assistance with maintaining state-of-the-art computing infrastructure necessary for this research.

GMH was supported by the National Institutes of Health through the award R35GM138312, and MD simulatoins were performed on NYU High Performance Computing resources, using GPUs purchased by the Simons Center for Computational Physical Chemistry (SCCPC) at NYU (SF Grant No. 839534).

6 Data and Code Availability

Datasets can be found from their respective open sources, specifically the National Center for Biotechnology Information (Genbank), and UniProt (for Uniref100). Additionally, we maintain code for the downloading and preprocessing of this data on our Github. We release all foundation models on HuggingFace https://huggingface.co/WeiHua/OmniBioTE to accelerate the development of novel downstream use cases built on top of our foundation model. Additionally, the code for training and evaluating our models is available on our Github repository (https://github.com/nyuolab/OmniBioTE). Code and data for predictions combining AlphaFold3 with MD simulations are available from Zenodo https://zenodo.org/records/15098577.

7 Inclusion and Ethics

SFC, RJS, GMH designed the experiments, wrote the code, and wrote the manuscript. BL assisted in data collection and analysis. SPL and EKO assisted in the design of the experiments, writing of the manuscript, and direction of the research.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information*

Processing Systems, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[2] OpenAI et al. Gpt-4 technical report, 2024.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[4] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[5] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[6] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

[7] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, aug 2021.

[8] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, jun 2024.

[9] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N. Kinch, R. Dustin Schaeffer, Claudia Millán, Hahnbeom Park, Carson Adams, Caleb R. Glassman, Andy DeGiovanni, Jose H. Pereira, Andria V. Rodrigues, Alberdina A. van Dijk, Ana C. Ebrecht, Diederik J. Opperman, Theo Sagmeister, Christoph Buhlheller, Tea Pavkov-Keller, Manoj K. Rathinaswamy, Udit Dalwadi, Calvin K. Yip, John E. Burke, K. Christopher Garcia, Nick V. Grishin, Paul D. Adams, Randy J. Read, and David Baker. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

[10] Minkyung Baek, Ryan McHugh, Ivan Anishchenko, Hanlun Jiang, David Baker, and Frank DiMaio. Accurate prediction of protein–nucleic acid complexes using rosettafoldna. *Nature Methods*, 21(1):117–121, jan 2024.

[11] Gustaf Ahdritz, Nazim Bouatta, Christina Floristean, Sachin Kadyan, Qinghui Xia, William Gerecke, Timothy J. O'Donnell, Daniel Berenberg, Ian Fisk, Niccolò Zanichelli, Bo Zhang, Arkadiusz Nowaczynski, Bei Wang, Marta M. Stepniewska-Dziubinska, Shang Zhang, Adegoke Ojewole, Murat Efe Guney, Stella Biderman, Andrew M. Watkins, Stephen Ra, Pablo Ribalta Lorenzo, Lucas Nivon, Brian Weitzner, Yih-En Andrew Ban, Shiyang Chen, Minjia Zhang, Conglong Li, Shuaiwen Leon Song, Yuxiong He, Peter K. Sorger, Emad Mostaque, Zhao Zhang, Richard Bonneau, and Mohammed AlQuraishi. Openfold: retraining alphafold2 yields new insights into its learning mechanisms and capacity for generalization. *Nature Methods*, may 2024.

[12] Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, Jianzhu Ma, and Jian Peng. High-resolution de novo structure prediction from primary sequence. *bioRxiv*, 2022.

[13] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[14] Ahmed Elnaggar, Hazem Essam, Wafaa Salah-Eldin, Walid Moustafa, Mohamed Elkerdawy, Charlotte Rochereau, and Burkhard Rost. Ankh: Optimized protein language model unlocks general-purpose modelling. *arXiv preprint arXiv:2301.06568*, 2023.

[15] Yaron Geffen, Yanay Ofran, and Ron Unger. Distilprotbert: a distilled protein language model used to distinguish between real proteins and their randomly shuffled counterparts. *Bioinformatics*, 38(Supplement_2):ii95–ii98, 2022.

[16] Ananthan Nambiar, Maeve Heflin, Simon Liu, Sergei Maslov, Mark Hopkins, and Anna Ritz. Transforming the language of life: transformer neural networks for protein prediction tasks. In *Proceedings of the 11th ACM international conference on bioinformatics, computational biology and health informatics*, pages 1–8, 2020.

[17] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

[18] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

[19] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in neural information processing systems*, 34:29287–29303, 2021.

[20] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In *International Conference on Machine Learning*, pages 8844–8856. PMLR, 2021.

[21] Michael Heinzinger, Konstantin Weissenow, Joaquin Gomez Sanchez, Adrian Henkel, Martin Steinegger, and Burkhard Rost. Prostt5: Bilingual language model for protein sequence and structure. *bioRxiv*, 2023.

[22] Bo Chen, Xingyi Cheng, Yangli-ao Geng, Shen Li, Xin Zeng, Boyan Wang, Jing Gong, Chiming Liu, Aohan Zeng, Yuxiao Dong, Jie Tang,

and Le Song. xtrimopglm: Unified 100b-scale pre-trained transformer for deciphering the language of protein. *bioRxiv*, 2023.

[23] Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. *bioRxiv*, pages 2023–10, 2023.

[24] Pascal Notin, Mafalda Dias, Jonathan Frazer, Javier Marchena-Hurtado, Aidan N Gomez, Debora Marks, and Yarin Gal. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. In *International Conference on Machine Learning*, pages 16990–17017. PMLR, 2022.

[25] Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Alex Xijie Lu, Nicolo Fusi, Ava Pardis Amini, and Kevin K Yang. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, pages 2023–09, 2023.

[26] Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41(8):1099–1106, 2023.

[27] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

[28] Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12(1):3168, 2021.

[29] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

[30] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. *BioRxiv*, pages 2022–12, 2022.

[31] Ratul Chowdhury, Nazim Bouatta, Surojit Biswas, Charlotte Rochereau, George M. Church, Peter K. Sorger, and Mohammed AlQuraishi. Single-sequence protein structure prediction using language models from deep learning. *bioRxiv*, 2021.

[32] Veniamin Fishman, Yuri Kuratov, Maxim Petrov, Aleksei Shmelev, Denis Shepelin, Nikolay Chekanov, Olga Kardymon, and Mikhail Burtsev. Gena-lm: A family of open-source foundational models for long dna sequences. *bioRxiv*, pages 2023–06, 2023.

[33] Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Michael Wornow, Callum Birch-Sykes, Stefano Massaroli, Aman Patel, Clayton Rabideau, Yoshua Bengio, et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *Advances in neural information processing systems*, 36, 2024.

[34] Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P de Almeida, Hassan Sirelkhatim, et al. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *BioRxiv*, pages 2023–01, 2023.

[35] Kseniia Dudnyk, Donghong Cai, Chenlai Shi, Jian Xu, and Jian Zhou. Sequence basis of

transcription initiation in the human genome. *Science*, 384(6694):eadj0116, 2024.

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

[37] Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. Dnabert-2: Efficient foundation model and benchmark for multi-species genome, 2024.

[38] Maxim Zvyagin, Alexander Brace, Kyle Hippe, Yuntian Deng, Bin Zhang, Cindy Orozco Bohorquez, Austin Clyde, Bharat Kale, Danilo Perez-Rivera, Heng Ma, et al. Genslms: Genome-scale language models reveal sars-cov-2 evolutionary dynamics. *The International Journal of High Performance Computing Applications*, 37(6):683–705, 2023.

[39] Yunha Hwang, Andre L Cornman, Elizabeth H Kellogg, Sergey Ovchinnikov, and Peter R Girguis. Genomic language model predicts protein co-regulation and function. *Nature communications*, 15(1):2880, 2024.

[40] Garyk Brixi, Matthew G Durrant, Jerome Ku, Michael Poli, Greg Brockman, Daniel Chang, Gabriel A Gonzalez, Samuel H King, David B Li, Aditi T Merchant, Mohsen Naghipourfar, Eric Nguyen, Chiara Ricci-Tam, David W Romero, Gwanggyu Sun, Ali Taghibakshi, Anton Vorontsov, Brandon Yang, Myra Deng, Liv Gorton, Nam Nguyen, Nicholas K Wang, Etowah Adams, Stephen A Baccus, Steven Dillmann, Stefano Ermon, Daniel Guo, Rajesh Ilango, Ken Janik, Amy X Lu, Reshma Mehta, Mohammad R.K. Mofrad, Madelena Y Ng, Jaspreet Pannu, Christopher Re, Jonathan C Schmok, John St. John, Jeremy Sullivan, Kevin Zhu, Greg Zynda, Daniel Balsam, Patrick Collison, Anthony B. Costa, Tina Hernandez-Boussard, Eric Ho, Ming-Yu Liu, Tom McGrath, Kimberly Powell, Dave P. Burke, Hani Goodarzi, Patrick D Hsu, and Brian Hie. Genome modeling and design across all domains of life with evo 2. *bioRxiv*, 2025.

[41] Sho Tsukiyama, Md Mehedi Hasan, Hong-Wen Deng, and Hiroyuki Kurata. BERT6mA: prediction of DNA N6-methyladenine site using deep learning-based approaches. *Briefings in Bioinformatics*, 23(2):bbac053, 02 2022.

[42] Gaetan De Waele, Jim Clauwaert, Gerben Menschaert, and Willem Waegeman. CpG Transformer for imputation of single-cell methylomes. *Bioinformatics*, 38(3):597–603, 10 2021.

[43] Junru Jin, Yingying Yu, Ruheng Wang, Xin Zeng, Chao Pang, Yi Jiang, Zhongshen Li, Yutong Dai, Ran Su, Quan Zou, et al. idna-abf: multi-scale deep biological language learning model for the interpretable prediction of dna methylations. *Genome biology*, 23(1):219, 2022.

[44] Jiyun Zhou, Qiang Chen, Patricia R Braun, Kira A Perzel Mandell, Andrew E Jaffe, Hao Yang Tan, Thomas M Hyde, Joel E Kleinman, James B Potash, Gen Shinozaki, et al. Deep learning predicts dna methylation regulatory variants in the human brain and elucidates the genetics of psychiatric disorders. *Proceedings of the National Academy of Sciences*, 119(34):e2206069119, 2022.

[45] Dohoon Lee, Jeewon Yang, and Sun Kim. Learning the histone codes with large genomic windows and three-dimensional chromatin interactions using transformer. *Nature Communications*, 13(1):6678, 2022.

[46] Zhongliang Zhou, Wayland Yeung, Nathan Gravel, Mariah Salcedo, Saber Soleymani, Sheng Li, and Natarajan Kannan. Phosformer: an explainable transformer model for

protein kinase-specific phosphorylation predictions. *Bioinformatics*, 39(2):btad046, 01 2023.

[47] Xi Fu, Shentong Mo, Alejandro Buendia, Anouchka Laurent, Anqi Shao, Maria del Mar Alvarez-Torres, Tianji Yu, Jimin Tan, Jiayu Su, Romella Sagatelian, et al. Get: a foundation model of transcription across human cell types. *bioRxiv*, pages 2023–09, 2023.

[48] Fan Yang, Wenchuan Wang, Fang Wang, Yuan Fang, Duyu Tang, Junzhou Huang, Hui Lu, and Jianhua Yao. scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. *Nature Machine Intelligence*, 4(10):852–866, 2022.

[49] Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Xuegong Zhang, and Le Song. Large-scale foundation model on single-cell transcriptomics. *Nature Methods*, pages 1–11, 2024.

[50] Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature Methods*, pages 1–11, 2024.

[51] Christina V Theodoris, Ling Xiao, Anant Chopra, Mark D Chaffin, Zeina R Al Sayed, Matthew C Hill, Helene Mantineo, Elizabeth M Brydon, Zexian Zeng, X Shirley Liu, et al. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

[52] Sizhen Li, Saeed Moayedpour, Ruijiang Li, Michael Bailey, Saleh Riahi, Lorenzo Kogler-Anele, Milad Miladi, Jacob Miner, Dinghai Zheng, Jun Wang, et al. Codonbert: Large language models for mrna design and optimization. *bioRxiv*, pages 2023–09, 2023.

[53] Albi Celaj, Alice Jiexin Gao, Tammy TY Lau, Erle M Holgersen, Alston Lo, Varun Lodaya, Christopher B Cole, Robert E Denroche, Carl Spickett, Omar Wagih, et al. An rna foundation model enables discovery of disease mechanisms and candidate therapeutics. *bioRxiv*, pages 2023–09, 2023.

[54] Linjing Liu, Wei Li, Ka-Chun Wong, Fan Yang, and Jianhua Yao. A pre-trained large generative model for translating single-cell transcriptome to proteome. *bioRxiv*, pages 2023–07, 2023.

[55] Hongru Shen, Jilei Liu, Jiani Hu, Xilin Shen, Chao Zhang, Dan Wu, Mengyao Feng, Meng Yang, Yang Li, Yichen Yang, et al. Generative pretraining from large-scale transcriptomes for single-cell deciphering. *Iscience*, 26(5), 2023.

[56] Jing Gong, Minsheng Hao, Xingyi Cheng, Xin Zeng, Chiming Liu, Jianzhu Ma, Xuegong Zhang, Taifeng Wang, and Le Song. xtrimogene: an efficient and scalable representation learner for single-cell rna-seq data. *Advances in Neural Information Processing Systems*, 36, 2024.

[57] Lay Kuan Loh and Mihovil Bartulovic. Efficient coding hypothesis and an introduction to information theory. *Retrieved from users. ece. cmu. edu/˜ pgrover/teaching/files/InfoTheoryEfficientCo dingHypothesis. pdf. Homayoun Shahri*, 2014.

[58] Uddeshya Pandey, Sasi M. Behara, Siddhant Sharma, Rachit S. Patil, Souparnika Nambiar, Debasish Koner, and Hussain Bhukya. Deepnap: A deep learning method to predict protein–nucleic acid binding affinity from their sequences. *Journal of Chemical Information and Modeling*, 64(6):1806–1815, 2024. PMID: 38458968.

[59] Fu Chen, Huiyong Sun, Junmei Wang, Feng Zhu, Hui Liu, Zhe Wang, Tailong Lei, Youyong Li, and Tingjun Hou. Assessing the performance of mm/pbsa and mm/gbsa methods. 8. predicting binding free energies and poses of protein–rna complexes. *RNA*, 24(9):1183–1194, 2018.

[60] Ercheng Wang, Huiyong Sun, Junmei Wang, Zhe Wang, Hui Liu, John ZH Zhang, and Tingjun Hou. End-point binding free energy calculation with mm/pbsa and mm/gbsa: strategies and applications in drug design. *Chem. Rev.*, 119(16):9478–9508, 2019.

[61] Christian Kramer, Tuomo Kalliokoski, Peter Gedeck, and Anna Vulpetti. The experimental uncertainty of heterogeneous public k i data. *Journal of medicinal chemistry*, 55(11):5165–5173, 2012.

[62] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021.

[63] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[64] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[65] Evangelos S. Gragoudas, Anthony P. Adamis, Emmett T. Cunningham, Matthew Feinsod, and David R. Guyer. Pegaptanib for neovascular age-related macular degeneration. *New England Journal of Medicine*, 351(27):2805–2816, 2004.

[66] Josué Carvalho, Artur Paiva, Maria Paula Cabral Campello, António Paulo, Jean-Louis Mergny, Gilmar F. Salgado, João A. Queiroz, and Carla Cruz. Aptamer-based targeted delivery of a g-quadruplex ligand in cervical cancer cells. *Scientific Reports*, 9(1):7945, 2019.

[67] Kenichi A. Tanaka, Fania Szlam, Christopher P. Rusconi, and Jerrold H. Levy. In-vitro evaluation of anti-factor ixa aptamer on thrombin generation, clotting time, and viscoelastometry. *Thrombosis and Haemostasis*, 101(5):827–833, May 2009.

[68] M. Y. Chan, C. P. Rusconi, J. H. Alexander, R. M. Tonkens, R. A. Harrington, and R. C. Becker. A randomized, repeat-dose, pharmacodynamic and safety study of an antidote-controlled factor ixa inhibitor. *Journal of Thrombosis and Haemostasis*, 6(5):789–796, May 2008.

[69] Claudia Riccardi, Albert Meyer, Jean-Jacques Vasseur, Domenico Cavasso, Irene Russo Krauss, Luigi Paduano, François Morvan, and Daniela Montesarchio. Design, synthesis and characterization of cyclic nu172 analogues: A biophysical and biological insight. *International Journal of Molecular Sciences*, 21(11):3860, May 2020.

[70] Petra Jilma-Stohlawetz, Paul Knöbl, James C. Gilbert, and Bernd Jilma. The anti-von willebrand factor aptamer arc1779 increases von willebrand factor levels and platelet counts in patients with type 2b von willebrand disease. *Thrombosis and Haemostasis*, 108(2):284–290, August 2012.

[71] Jan Menne, Dirk Eulberg, Diana Beyer, Matthias Baumann, Frantisek Saudek, Zsuzsanna Valkusz, Andrzej Więcek, and Hermann Haller. C-c motif-ligand 2 inhibition with emapticap pegol (nox-e36) in type 2 diabetic patients with albuminuria. *Nephrology, Dialysis, Transplantation*, 32(2):307–315, 2017.

[72] Frank A. Giordano, Julian P. Layer, Sonia Leonardelli, Lea L. Friker, Roberta Turiello, Dillon Corvino, Thomas Zeyen, Christina Schaub, Wolf Müller, Elena Sperk, Leonard Christopher Schmeel, Katharina

Sahm, Christoph Oster, Sied Kebir, Peter Hambsch, Torsten Pietsch, Sotirios Bisdas, Michael Platten, Martin Glas, Clemens Seidel, Ulrich Herrlinger, and Michael Hölzel. L-rna aptamer-based cxcl12 inhibition combined with radiotherapy in newly-diagnosed glioblastoma: dose escalation of the phase i/ii gloria trial. *Nature Communications*, 15(1):4210, 2024.

[73] Frank Schwoebel, Lucas T. van Eijk, Dirk Zboralski, Simone Sell, Klaus Buchner, Christian Maasch, Werner G. Purschke, Martin Humphrey, Stefan Zöllner, Dirk Eulberg, Frank Morich, Peter Pickkers, and Sven Klussmann. The effects of the anti-hepcidin spiegelmer nox-h94 on inflammation-induced anemia in cynomolgus monkeys. *Blood*, 121(12):2311–2315, 2013.

[74] Eric W Sayers, Mark Cavanaugh, Karen Clark, James Ostell, Kim D Pruitt, and Ilene Karsch-Mizrachi. GenBank. *Nucleic Acids Research*, 47(D1):D94–D99, 10 2018.

[75] Baris E. Suzek, Hongzhan Huang, Peter McGarvey, Raja Mazumder, and Cathy H. Wu. UniRef: comprehensive and non-redundant UniProt reference clusters. *Bioinformatics*, 23(10):1282–1288, 03 2007.

[76] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units, 2016.

[77] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, 2018.

[78] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[79] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[80] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[81] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023.

[82] Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer, 2022.

[83] Andrej Karpathy. NanoGPT. https://github.com/karpathy/nanoGPT, 2022.

[84] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[85] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates, 2018.

[86] Kannan Harini, Ambuj Srivastava, Arulsamy Kulandaisamy, and M Michael Gromiha. ProNAB: database for binding affinities of protein–nucleic acid complexes and their mutants. *Nucleic Acids Research*, 50(D1):D1528–D1534, 10 2021.

[87] Steven Henikoff and Jorja G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992.

[88] Uddeshya Pandey, Sasi M Behara, Siddhant Sharma, Rachit S Patil, Souparnika Nambiar, Debasish Koner, and Hussain Bhukya. Deepnap: A deep learning method to predict protein–nucleic acid binding affinity from their sequences. *Journal of Chemical Information and Modeling*, 64(6):1806–1815, 2024.

[89] Yong He, Pan Fang, Yongtao Shan, Yuanfei Pan, Yanhong Wei, Yichang Chen, Yihao Chen, Yi Liu, Zhenyu Zeng, Zhan Zhou, et al. Lucaone: generalized biological foundation model with unified nucleic acid and protein language. *bioRxiv*, pages 2024–05, 2024.

[90] Ieva Rauluseviciute, Rafael Riudavets-Puig, Romain Blanc-Mathieu, Jaime A Castro-Mondragon, Katalin Ferenc, Vipin Kumar, Roza Berhanu Lemma, Jérémy Lucas, Jeanne Chèneby, Damir Baranasic, et al. Jaspar 2024: 20th anniversary of the open-access database of transcription factor binding profiles. *Nucleic acids research*, 52(D1):D174–D182, 2024.

[91] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 01 2000.

[92] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019.

[93] Mingxue Xu, Yao Lei Xu, and Danilo P. Mandic. Tensorgpt: Efficient compression of the embedding layer in llms based on the tensor-train decomposition, 2023.

[94] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[95] Henriette Capel, Robin Weiler, Maurits Dijkstra, Reinier Vleugels, Peter Bloem, and K. Anton Feenstra. Proteinglue: A multi-task benchmark suite for self-supervised protein modeling. *bioRxiv*, 2021.

[96] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[97] Peter Eastman, Raimondas Galvelis, Raúl P Peláez, Charlles RA Abreu, Stephen E Farr, Emilio Gallicchio, Anton Gorenko, Michael M Henry, Frank Hu, Jing Huang, et al. Openmm 8: molecular dynamics simulation with machine learning potentials. *J. Phys. Chem. B*, 128(1):109–116, 2023.

[98] Benoît Roux and Christophe Chipot. Editorial guidelines for computational studies of ligand binding using mm/pbsa and mm/gbsa approximations wisely. *J. Phys. Chem. B*, 128(49):12027–12029, 2024.

[99] James A Maier, Carmenza Martinez, Koushik Kasavajhala, Lauren Wickstrom, Kevin E Hauser, and Carlos Simmerling. ff14sb: improving the accuracy of protein side chain and backbone parameters from ff99sb. *J. Chem. Theor. Comput.*, 11(8):3696–3713, 2015.

[100] Hai Nguyen, Daniel R Roe, and Carlos Simmerling. Improved generalized born solvent model parameters for protein simulations. *J. Chem. Theor. Comput.*, 9(4):2020–2034, 2013.

[101] Benedict Leimkuhler and Charles Matthews. Robust and efficient configurational molecular sampling via langevin dynamics. *J Chem Phys*, 138(17), 2013.

[102] Zhijun Zhang, Xinzijian Liu, Kangyu Yan, Mark E Tuckerman, and Jian Liu. Unified efficient thermostat scheme for the canonical ensemble with holonomic or isokinetic constraints via molecular dynamics. *J Phys Chem A*, 123(28):6056–6079, 2019.

[103] Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza Revilla, Nicolas Lopez Carranza, Adam Henryk Grywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Hassan Sirelkhatim, Guillaume Richard, et al. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *bioRxiv*, pages 2023–01, 2023.

[104] Chaoqi Liang, Lifeng Qiao, Peng Ye, Nanqing Dong, Jianle Sun, Weiqiang Bai, Yuchen Ren, Xinzhu Ma, Hongliang Yan, Chunfeng Song, et al. Toward understanding bert-like pre-training for dna foundation models. *arXiv preprint arXiv:2310.07644*, 2023.

Supplementary Information

S1 Predicting binding interactions between proteins and nucleic acids

We developed and assessed a pipeline for predicting the interaction energy between proteins and nucleic acids by combining AlphaFold3 (AF3) [8] with molecular dynamics (MD) simulations performed in OpenMM version 8.1.2 [97]. Our pipeline is available at the link https://github.com/hockyg/af3_protein_nuclic_md_pipeline. We applied this pipeline to as many targets as possible from the ProNAB database studied in Fig. 3. Ultimately, we were able to generate structure predictions and perform MD simulations on 599 protein/nucleic acid pairs.

Given that we wanted to compare our ability to predict binding energies from sequence directly, this required us to generate bound conformations via a machine learning approach that treats both proteins and nucleic acids, and so for that we selected one of the only available options, AF3 [8]. Due to the large size of the systems and the need to rapidly evaluate interactions, we were forced to use an *implicit solvent* approach. This need was exacerbated by the fact that AF3 predicted structures often have large regions with low confidence scores that are non-compact, resulting in simulation boxes that would be intractable if filled with water (i.e. millions of atoms including solvent and ions). Using implicit solvent, the MD simulations executed for our targets ranged in size from 1195 atoms to 60832 atoms, with an average size of approximately 7653.

To approximately compute the binding energy between a protein and nucleic acid in tractable computational time, we adopted a protocol similar to the so-called MM/GBSA approach [60]. To compute the binding free energy of a complex, we need to compute

$$\Delta G = G_{\mathrm{AB}} - G_{\mathrm{A}} - G_{\mathrm{B}}, \tag{1}$$

where A and B are the separate components and the free energies on each side are averaged over a conformational ensemble. ΔG has contributions that come from the (1) direct interaction energy between the molecules, (2) the change in solvation free energy due to the difference in buried surface area, (3) and the change in configurational entropy of both parts upon binding. When using simulations with implicit solvent, effects 1 and 2 are taken into effect if we simply calculate the MD energy. The third effect due to overall changes in the conformations of the bound and unbound A and B molecules is not possible to calculate in a single simulation and requires extensive calculations beyond the scope of this work. However, conveniently, we expect that for calculations of $\Delta\Delta G$ of mutation, this term cancels out. Below, we will therefore run short MD simulations and compute the energy of the complex as well as for separate components in order to see whether $\Delta G^{\mathrm{experiment}}$ can be predicted. *We emphasize that we do not expect this to work in general [98], and we are performing these calculations to set a baseline for our ML predictions given in the main text.*

To go from sequence to energy prediction, we start by converting entries in the ProNAB database [86] into YAML files suitable for AF3 predictions. This consists of specifying a protein chain and a nucleic acid chain (or chains in the case of a double stranded sequence). We also added 1 Mg^{2+} ion per nucleotide in case explicit divalent cations were needed for solvated MD simulations in the future. These divalent ions were removed for implicit solvent simulations performed next.

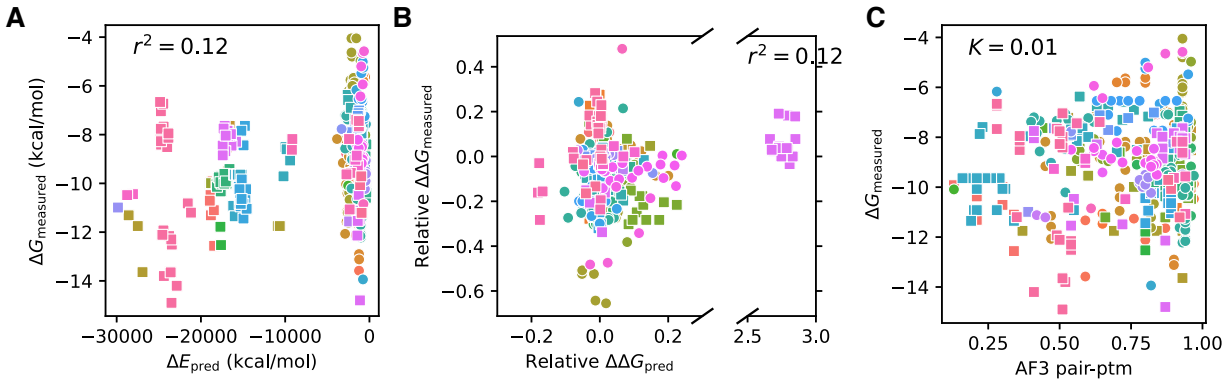

Figure S1: (A) Comparison of energy differences between bound and separated states using MD as described in the text with experimentally measured values. Squares denote proteins bound to DNA while circles are proteins bound to RNA. Color indicates a different protein. (B) Relative predicted free energy difference between binding a native and mutant nucleic acid sequence. (C) Comparison of experimental binding free energies with AF3 PTM scores (0-1 with 1 being a high confidence in binding prediction).

The topology of the system was built from the AF3 output CIF file using PDBFixer in OpenMM [97]. The forcefield used was Amber14 [99] with the GB2 implicit solvent model [100]. After minimizing the energy, the velocities were randomies and the system was equilibrated at $T = 300$K by running 100 ps of MD, before running 10 ns of MD using a 4 fs timestep and the `LangevinMiddle` integrator [101,102] with hydrogen mass repartitioning to a mass of 3 amu and a drag coefficient of 1 ps^{-1}. The energies of the full complex, and the protein and nucleic acid separately were averaged over the final 5 ns of MD simulation to produce the $\Delta E_{\text{pred}} = E_{AB} - E_A - E_B$ values in Fig. S1.

Fig. S1A shows a scatter plot comparing ΔE_{pred} with measured binding free energies for those complexes. Multiple values are given for the same protein when different nucleic acid sequences were given in the database for mutational studies. As can be seen, both the order of magnitude is greatly different, and also there is little to no correlation as measured by the square of the Pearson correlation, r^2.

In Fig. S1B we show the results of computing a predicted $\Delta\Delta G_{\text{pred}} \approx \Delta E_{\text{pred}}^{\text{mutant}} - \Delta E_{\text{pred}}^{\text{native}}$ to those from experiment. The absolute values of $\Delta\Delta G_{\text{pred}}$ were much larger than for experiment again, so here we show the relative difference (i.e. scaled by the native binding free energy or energy value). Again here there is no correlation. There is also one target (p04390) which is an outlier, but there is nothing obvious suggesting why this protein shows such a large relative error compared to the others. There is also little to no correlation when removing this outlier.

We check whether the PTM confidence scores reported by AF3 [8] are correlated with the binding affinity for these complexes. This metric also has no correlation, as measured by the Spearman rank order correlation coefficient, K (Fig. S1C).

Finally, we can also consider the time required to compute these results. Simulations that we attempted ranged from 5 to 912 ns/day on a single GPU. For those simulations that completed the full 10 ns of MD, times ranged from approximately 0.01 to 0.2 hours, for a total of approximately 13.4 GPU-hours. While this already far exceeds the inference time from our ML model, this was not the time consuming part of the calculation. AF3 calculations can be split into two parts, one which involves a multiple sequence alignment (MSA), and then the actual inference [8]. While the inference step is relatively fast, the MSA step is slow

and CPU bound, and was the longest part of the calculation. For these calculations, the average inference took 0.9±0.4 minutes on an A100 GPU, while the average MSA computation time took 47.5 minutes and ranged from 14 to 283 minutes on 16 CPUs, for a total of over 6700 CPU-hours.

Dataset Statistics

Sequence Type	**Average Length** (bp/residues)	**Minimum Length** (bp/residues)	**Maximum Length** (bp/residues)
DNA	16941.82 ± 1421192.40	6	363684565
mRNA	624.75 ± 539.25	6	84308
RNA	10027.20 ± 39070.07	6	167463040
cRNA	1769.64 ± 1945.04	35	157276
rRNA	482.19 ± 266.62	24	7097
ss-RNA	3087.72 ± 4797.85	14	35911
ss-DNA	1637.48 ± 1253.90	17	34395
ds-RNA	2075.30 ± 2197.43	48	31081
tRNA	249.23 ± 349.17	20	1208
ds-cRNA	657.16 ± 970.66	127	15341
ms-DNA	15492.76 ± 19181.55	84	45513
ds-mRNA	1695.75 ± 547.50	1114	2414
ms-RNA	606.50 ± 28.50	578	635
ds-rRNA	580.00 ± 0.00	580	580
peptide	388.19 ± 379.80	5	45359

Table S1: Training data statistics across all sequence types.

Evaluation Results

Model	ΔG PCC	ΔG MAE
OmniBioTE-small	25.50 ± 9.48	1.68 ± 0.22
OmniBioTE-medium	37.21 ± 7.34	1.57 ± 0.20
OmniBioTE-large	34.61 ± 7.86	1.60 ± 0.21
OmniBioTE-XL	**40.67 ± 9.77**	**1.56 ± 0.23**
OmniBioTE-small (per-nucleotide/residue)	22.73 ± 12.23	1.72 ± 0.20
OmniBioTE-medium (per-nucleotide/residue)	33.53 ± 6.51	1.63 ± 0.22
OmniBioTE-large (per-nucleotide/residue)	29.89 ± 12.65	1.63 ± 0.29
OmniBioTE-XL (per-nucleotide/residue)	37.75 ± 8.78	1.56 ± 0.25
Nuc+ProtBioTE-small	7.08 ± 11.15	1.71 ± 0.24
Nuc+ProtBioTE-medium	8.13 ± 11.62	1.83 ± 0.32
Nuc+ProtBioTE-large	3.85 ± 13.35	1.83 ± 0.29
Nuc+ProtBioTE-XL	7.71 ± 14.02	1.90 ± 0.29
LucaOne	19.98 ± 0.16	2.43 ± 0.29
DeePNAP	10.00 ± 11.13	2.35 ± 0.41
AlphaFold3 + simulation	11.00	–

Table S2: OmniBioTE performance across all 10-folds of the Pronab mutation benchmark as measured in Pearson correlation coefficient (PCC) and mean absolute error (MAE).

OmniBioTE						
Model	H3	H3K14ac	H3K36me3	H3K4me1	H3K4me2	H3K4me3
OmniBioTE-small	77.01	58.23	59.42	51.83	33.62	37.89
OmniBioTE-medium	79.75	66.40	68.01	60.03	49.56	55.02
OmniBioTE-large	80.64	67.31	69.48	59.04	46.64	55.33
OmniBioTE-XL	82.11	67.34	70.22	58.14	52.45	57.43
OmniBioTE (per-nucleotide)						
OmniBioTE-small (per-nucleotide)	77.85	53.96	60.93	54.67	30.32	30.32
OmniBioTE-medium (per-nucleotide)	80.76	56.94	63.58	54.69	32.52	45.27
OmniBioTE-large (per-nucleotide)	82.64	71.52	69.89	62.42	57.56	59.33
OmniBioTE-XL (per-nucleotide)	80.47	59.27	66.20	54.59	45.71	47.16
NucBioTE						
NucBioTE-small	76.93	53.83	56.46	46.81	36.11	40.34
NucBioTE-medium	75.44	49.76	59.04	38.98	27.55	35.10
NucBioTE-large	76.51	53.51	55.45	47.05	32.68	40.71
NucBioTE-XL	80.81	66.91	66.44	55.26	47.20	57.04
Baselines						
HyenaDNA [33]	67.17	31.98	48.27	35.83	25.81	23.15
NT-2500M-multi [103]	78.77	56.20	61.99	55.30	36.49	40.34
DNABERT-2 [37]	78.27	52.57	56.88	50.52	31.13	36.27
RandomMask [104]	77.62	65.07	63.68	54.47	53.88	62.19
LucaOne	72.28	44.61	46.72	42.33	28.79	25.96

Table S3: GUE Results (Epigenetics): Histone Modification Benchmarks (Part 1). Values represent the Matthews correlation coefficient of the predictions.

Model	H3K79me3	H3K9ac	H4	H4ac
OmniBioTE				
OmniBioTE-small	63.48	61.94	79.70	47.12
OmniBioTE-medium	72.99	68.79	82.50	62.57
OmniBioTE-large	72.57	67.99	82.50	63.62
OmniBioTE-XL	73.35	66.75	81.55	63.71
OmniBioTE (per-nucleotide)				
OmniBioTE-small (per-nucleotide)	63.14	57.13	81.94	46.86
OmniBioTE-medium (per-nucleotide)	67.62	59.17	82.54	52.22
OmniBioTE-large (per-nucleotide)	73.69	67.91	83.48	65.86
OmniBioTE-XL (per-nucleotide)	73.07	60.89	80.90	58.18
NucBioTE				
NucBioTE-small	63.17	54.31	78.69	52.12
NucBioTE-medium	62.50	51.78	79.86	38.15
NucBioTE-large	66.78	58.41	80.84	50.19
NucBioTE-XL	72.73	65.95	82.65	62.41
Baselines				
HyenaDNA [33]	54.09	50.84	73.69	38.44
NT-2500M-multi [103]	64.70	56.01	81.67	49.13
DNABERT-2 [37]	67.39	55.63	80.71	50.43
RandomMask [104]	72.67	65.02	79.44	64.22
LucaOne	59.69	50.82	76.24	36.70

Table S4: GUE Results (Epigenetics): Histone Modification Benchmarks (Part 2). Values represent the Matthews correlation coefficient of the predictions.

Model	Human Transcription Factors					Covid
	0	1	2	3	4	
OmniBioTE						
OmniBioTE-small	65.67	70.07	56.43	46.36	65.81	67.93
OmniBioTE-medium	62.37	72.04	59.63	47.22	76.02	69.38
OmniBioTE-large	62.53	72.08	60.40	51.94	75.76	69.26
OmniBioTE-XL	64.82	69.95	63.75	55.44	75.65	68.77
OmniBioTE (per-nucleotide)						
OmniBioTE-small (per-nucleotide)	64.80	70.83	53.22	45.29	73.00	57.30
OmniBioTE-medium (per-nucleotide)	66.86	69.08	69.12	51.34	77.69	73.50
OmniBioTE-large (per-nucleotide)	65.77	70.46	67.49	51.62	77.74	76.55
OmniBioTE-XL (per-nucleotide)	66.50	67.82	62.95	53.32	76.02	74.11
NucBioTE						
NucBioTE-small	65.50	69.92	53.82	38.98	74.00	66.02
NucBioTE-medium	64.19	66.98	53.50	50.28	73.03	59.66
NucBioTE-large	63.50	65.24	56.67	41.90	69.28	67.01
NucBioTE-XL	64.78	68.50	59.15	43.18	76.83	67.82
Baselines						
HyenaDNA [33]	62.30	67.86	46.85	41.78	61.23	23.27
NT-2500M-multi [103]	66.64	70.28	58.72	51.65	69.34	73.04
DNABERT-2 [37]	71.99	76.06	66.52	58.54	77.43	71.02
RandomMask [104]	67.13	72.55	71.64	60.14	77.20	–
LucaOne	66.84	69.00	57.23	41.25	67.83	38.92

Table S5: GUE Results: Human Transcription Factors and COVID. Values represent the Matthews correlation coefficient of the predictions, with the exception of the COVID variant prediction task which uses F1-score.

Model	Mouse Transcription Factors				
	0	1	2	3	4
OmniBioTE					
OmniBioTE-small	46.67	82.67	81.71	68.29	43.07
OmniBioTE-medium	56.42	84.94	79.88	70.78	47.96
OmniBioTE-large	57.38	84.60	76.33	78.01	49.70
OmniBioTE-XL	60.50	85.01	83.61	83.26	52.01
OmniBioTE (per-nucleotide)					
OmniBioTE-small (per-nucleotide)	37.79	82.00	75.62	71.90	39.93
OmniBioTE-medium (per-nucleotide)	64.07	85.47	85.39	80.82	52.33
OmniBioTE-large (per-nucleotide)	63.83	84.86	83.55	84.24	51.43
OmniBioTE-XL (per-nucleotide)	63.95	85.60	81.10	87.52	53.05
NucBioTE					
NucBioTE-small	48.92	82.95	73.22	70.83	41.58
NucBioTE-medium	52.62	82.63	77.76	69.22	40.76
NucBioTE-large	48.34	81.23	72.00	69.91	37.15
NucBioTE-XL	53.11	83.38	73.85	63.73	48.65
Baselines					
HyenaDNA [33]	35.62	80.50	65.34	54.20	19.17
NT-2500M-multi [103]	63.31	83.76	71.52	69.44	47.07
DNABERT-2 [37]	56.76	84.77	79.32	66.47	52.66
RandomMask [104]	55.61	82.72	77.61	74.06	49.81
LucaOne	52.33	82.57	73.44	57.11	45.17

Table S6: GUE Results: Mouse Transcription Factors. Values represent the Matthews correlation coefficient of the predictions.

Model	SS3	SS8	SS3 CB513	SS8 CB513
OmniBioTE				
OmniBioTE-small	77.1	64.9	77.6	63.0
OmniBioTE-medium	81.3	69.0	82.9	68.5
OmniBioTE-large	82.0	69.8	83.4	69.7
OmniBioTE-XL	82.7	70.7	87.0	72.0
OmniBioTE (per-residue)				
OmniBioTE-small (per-residue)	76.7	64.5	76.4	62.6
OmniBioTE-medium (per-residue)	81.8	69.4	82.9	69.9
OmniBioTE-large (per-residue)	82.5	70.6	83.0	69.5
OmniBioTE-XL (per-residue)	82.8	71.1	83.5	73.0
ProtBioTE				
ProtBioTE-small	79.8	67.3	81.3	67.0
ProtBioTE-medium	84.1	72.3	87.8	72.8
ProtBioTE-large	84.9	73.0	86.5	73.8
ProtBioTE-XL	85.4	74.3	88.8	75.0
ESM2-t6-8M	76.0	63.8	73.4	58.7
ESM2-t12-35M	79.9	67.8	77.3	63.0
ESM2-t30-150M	83.0	71.7	81.0	67.6
ESM2-t33-650M	85.3	83.0	83.0	70.4
ESM2-t36-3B	85.6	75.1	82.8	70.5
LucaOne	75.5	62.8	73.2	58.2

Table S7: Performance on the structural prediction tasks in the ProteinGLUE dataset. Values represent the accuracy of the predictions.

Model	Protein-protein Interaction (AUCROC)	Hydrophobic patch rank (PCC)	Epitope detection (AUCROC)	Solvent accessibility (PCC)	Buried-residue prediction (Accuracy)
OmniBioTE					
OmniBioTE-small-1k	58.0	26.4	50.2	61.5	86.6
OmniBioTE-medium-1k	61.8	23.9	61.6	65.7	88.0
OmniBioTE-large-1k	59.1	23.7	55.9	64.2	87.9
OmniBioTE-XL-1k	60.8	26.5	59.5	56.5	88.1
OmniBioTE (per-residue)					
OmniBioTE-small (per-residue)	0.532	0.145	0.628	0.598	0.800
OmniBioTE-medium (per-residue)	0.568	0.114	0.573	0.623	0.828
OmniBioTE-large (per-residue)	0.641	0.133	0.554	0.659	0.833
OmniBioTE-XL (per-residue)	0.610	0.075	0.590	0.617	0.833
ProtBioTE					
ProtBioTE-small-1k	58.5	27.2	53.6	62.4	87.2
ProtBioTE-medium-1k	61.9	30.4	51.5	65.4	88.6
ProtBioTE-large-1k	59.9	7.4	61.0	55.4	88.8
ProtBioTE-XL-1k	61.7	28.4	56.2	59.3	89.2
ESM2-t6-8M	61.7	30.3	51.7	67.6	79.7
ESM2-t12-35M	53.6	21.1	50.3	71.1	82.0
ESM2-t30-150M	65.9	20.0	50.4	74.9	83.6
ESM2-t33-650M	65.3	20.2	52.3	76.9	84.8
ESM2-t36-3B	65.3	17.7	51.4	75.2	85.3
LucaOne	59.7	27.5	50.2	66.3	79.4

Table S8: Performance on the remaining tasks in the ProteinGLUE dataset.

Model	Secondary Structure (3-way)			Secondary Structure (8-way)		
	CASP12	CB513	TS115	CASP12	CB513	TS115
OmniBiota (OmniBioTE)						
OmniBioTE-small	0.695	0.733	0.762	0.568	0.598	0.640
OmniBioTE-medium	0.717	0.784	0.794	0.600	0.642	0.680
OmniBioTE-large	0.722	0.786	0.801	0.591	0.646	0.674
OmniBioTE-XL	0.708	0.798	0.805	0.582	0.656	0.681
OmniBiota (per-residue)						
OmniBioTE-small (per-residue)	0.721	0.757	0.787	0.585	0.616	0.669
OmniBioTE-medium (per-residue)	0.746	0.813	0.820	0.619	0.678	0.707
OmniBioTE-large (per-residue)	0.749	0.819	0.825	0.630	0.685	0.705
OmniBioTE-XL (per-residue)	0.751	0.822	0.828	0.615	0.688	0.716
ProtBioTE						
ProtBioTE-small	0.707	0.769	0.782	0.568	0.626	0.667
ProtBioTE-medium	0.717	0.784	0.794	0.600	0.642	0.680
ProtBioTE-large	0.767	0.822	0.828	0.591	0.646	0.674
ProtBioTE-XL	0.764	0.827	0.831	0.642	0.691	0.717
Baselines						
ESM2-t6-8M	0.702	0.731	0.658	0.590	0.586	0.658
ESM2-t12-35M	0.730	0.773	0.805	0.607	0.631	0.690
ESM2-t30-150M	0.753	0.802	0.716	0.634	0.668	0.716
ESM2-t33-650M	0.780	0.831	0.843	0.667	0.700	0.733
ESM2-t36-3B	0.781	0.826	0.842	0.668	0.701	0.740
LucaOne	0.700	0.720	0.755	0.578	0.569	0.630
TAPE-Transformer	0.710	0.730	0.770	0.590	0.590	0.640
TAPE-ResNet	0.700	0.750	0.780	0.570	0.590	0.660
TAPE-LSTM	0.720	0.750	0.780	0.580	0.590	0.640
Supervised [11]	0.700	0.730	0.760	0.570	0.580	0.650
UniRep [12]	0.720	0.730	0.770	0.590	0.570	0.630

Table S9: Secondary structure performance. In the 3-way columns, CASP12, CB513, and TS115 scores are reported; in the 8-way columns, the corresponding scores are reported. All values are measured in accuracy.

Model	Fold	Superfamily	Family	Fluorescence	Stability
OmniBiota-mixed (OmniBioTE)					
OmniBioTE-small	0.208	0.906	0.362	0.666	0.686
OmniBioTE-medium	0.219	0.965	0.454	0.655	0.722
OmniBioTE-large	0.226	0.971	0.455	0.660	0.671
OmniBioTE-XL	0.242	0.970	0.482	0.659	0.689
OmniBiota-char (per-residue)					
OmniBioTE-small (per-residue)	0.201	0.914	0.342	0.659	0.700
OmniBioTE-medium (per-residue)	0.231	0.966	0.475	0.587	0.689
OmniBioTE-large (per-residue)	0.240	0.972	0.512	0.662	0.711
OmniBioTE-XL (per-residue)	0.223	0.973	0.470	0.539	0.699
OmniBiota-protein (ProtBioTE)					
ProtBioTE-small	0.194	0.951	0.406	0.666	0.702
ProtBioTE-medium	0.219	0.965	0.454	0.655	0.722
ProtBioTE-large	0.226	0.971	0.455	0.666	0.683
ProtBioTE-XL	0.241	0.972	0.463	0.663	0.654
Baselines					
ESM2-t6-8M	0.240	0.911	0.439	0.663	0.660
ESM2-t12-35M	0.288	0.961	0.574	0.673	0.723
ESM2-t30-150M	0.272	0.978	0.601	0.672	0.761
ESM2-t33-650M	0.231	0.965	0.530	0.665	0.720
ESM2-t36-3B	0.249	0.970	0.542	0.654	0.774
LucaOne	0.266	0.949	0.487	0.639	0.703
TAPE-Transformer	0.21	0.88	0.34	0.68	0.73
TAPE-ResNet	0.26	0.92	0.43	0.67	0.69
TAPE-LSTM	0.17	0.77	0.31	0.21	0.73
Supervised [11]	0.17	0.79	0.20	0.33	0.64
UniRep [12]	0.23	0.87	0.38	0.67	0.73

Table S10: Remote homology (Fold, Superfamily, Family) classification performance measured in accuracy and regression performance (Fluorescence, Stability) measured in Spearman's correlation coefficient.

Model	Contacts P@L (long)	Contacts P@L (medium)
OmniBiota-mixed (OmniBioTE)		
OmniBioTE-small	0.286	0.339
OmniBioTE-medium	0.237	0.350
OmniBioTE-large	0.280	0.371
OmniBioTE-XL	0.300	0.334
OmniBiota-char (per-residue)		
OmniBioTE-small (per-residue)	0.544	0.682
OmniBioTE-medium (per-residue)	0.467	0.614
OmniBioTE-large (per-residue)	0.636	0.725
OmniBioTE-XL (per-residue)	0.755	0.789
OmniBiota-protein (ProtBioTE)		
ProtBioTE-small	0.307	0.373
ProtBioTE-medium	0.386	0.406
ProtBioTE-large	0.302	0.347
ProtBioTE-XL	0.318	0.394
Baselines		
ESM2-t6-8M	0.521	0.609
ESM2-t12-35M	0.506	0.676
ESM2-t30-150M	0.515	0.654
ESM2-t33-650M	0.765	0.822
ESM2-t36-3B	0.753	0.819
LucaOne	0.365	0.556
TAPE-Transformer	0.17	0.19
TAPE-ResNet	0.20	0.20
TAPE-LSTM	0.10	0.18
Supervised [11]	0.18	0.22
UniRep [12]	0.17	0.17

Table S11: Contact evaluation performance, reporting Contacts P@L for long- and medium-range contacts. All values are the computed precision of the predictions.