Structure-Based Function Prediction using Graph Convolutional Networks

Vladimir Gligorijevic[1], P. Douglas Renfrew[1], Tomasz Kosciolek[2,3], Julia Koehler Leman[1], Kyunghyun Cho[4,13,14], Tommi Vatanen[5,9], Daniel Berenberg[1], Bryn Taylor[2,11,12], Ian M. Fisk[10], Ramnik J. Xavier[5,6,7], Rob Knight[2,11,12] & Richard Bonneau[1,14]

[1]*Center for Computational Biology, Flatiron Institute, Simons Foundation, New York, NY, USA*
[2]*Department of Pediatrics, University of California San Diego, La Jolla, CA, USA*
[3]*Malopolska Centre of Biotechnology, Krakow, Poland*
[4]*Facebook AI Research*
[5]*Broad Institute of MIT and Harvard, Cambridge, MA, USA*
[6]*Center for Computational and Integrative Biology, Massachusetts General Hospital and Harvard Medical School, Boston, MA, USA*
[7]*Gastrointestinal Unit, and Center for the Study of Inflammatory Bowel Disease, Massachusetts General Hospital and Harvard Medical School, Boston, MA, USA.*
[8]*Center for Microbiome Informatics and Therapeutics, MIT, Cambridge, MA, USA*
[9]*The Liggins Institute, University of Auckland, Auckland, New Zealand*
[10]*Scientific Computing Core, Flatiron Institute, Simons Foundation, New York, NY, USA*
[11]*Center for Microbiome Innovation, University of California San Diego, La Jolla, CA, USA*
[12]*Department of Computer Science and Engineering, University of California San Diego, La Jolla, CA, USA*
[13]*CIFAR Azrieli Global Scholar*
[14]*Center for Data Science, New York University, New York, NY, USA*

Recent massive increases in the number of sequences available in public databases challenges current experimental approaches to determining protein function. These methods are limited by both the large scale of these sequences databases and the diversity of protein functions. We present a deep learning Graph Convolutional Network (GCN) trained on sequence and structural data and evaluate it on ~40k proteins with known structures and functions from the Protein Data Bank (PDB). Our GCN predicts functions more accurately than Convolutional Neural Networks trained on sequence data alone and competing methods. Feature extraction via a language model removes the need for constructing multiple sequence alignments or feature engineering. Our model learns general structure-function relationships by robustly predicting functions of proteins with $\leq 30\%$ sequence identity to the training set. Using class activation mapping, we can automatically identify structural regions at the residue-level that lead to each function prediction for every protein confidently predicted, advancing site-specific function prediction. De-noising inherent in the trained model allows an only minor drop in performance when structure predictions are used, including multiple *de novo* protocols. We use our method to annotate all proteins in the PDB, making several new confident function predictions spanning both fold and function trees.

Proteins are linear chains of amino acid residues that fold into 3-dimensional structures to carry out a wide variety of functions within the cell. Even though many (5–30%, depending on the organism) functional regions of proteins are disordered (lack a well defined ensemble average) the majority of protein domains in natural proteins fold into specific and ordered three-dimensional conformations as a result of the physical interactions within the chain[1–5]. The structural features of proteins, in turn, determine the wide range of functions: from binding specificity, forming structures within the cell, to catalysis of biochemical reactions, transport, or signal transduction. There are several widely used classification schemes that help to organize these myriad protein functions including: the Gene Ontology (GO) Consortium[6], the Comprehensive Enzyme Information System (BRENDA)[7], Enzyme Commission (EC) numbers[8], Kyoto Encyclopedia of Genes and Genomes (KEGG)[9], and others. GO, for example, classifies proteins into hierarchically related functional classes (also called GO terms) organized into 3 different ontologies: Molecular Function (MF), Biological Process (BP) and Cellular Component (CC) describing different aspects of protein functions.

The advent of low-cost and efficient protein sequencing technologies has resulted in the massive growth in the number of sequences available in key protein sequence databases, like the UniProt Knowledgebase (UniProtKB, `http://uniprot.org`). UniProt currently contains over 100 million sequences and only ~0.5 million sequences (0.5%) that are manually annotated (UniProtKB/Swiss-Prot). Most proteins with unknown function (i.e., hypothetical proteins) are unlikely to be experimentally characterized. Understanding the functional roles and studying the mechanisms of these newly discovered proteins, in both health and disease, is one of the most

important biological problems in the post-genomic era. In parallel to the growth of sequence data, the advent of experimental as well computational techniques in structure biology has made the three-dimensional structures of many proteins available[10–16].

The Protein Data Bank (PDB, `http://wwpdb.org`) remains the main repository of information about the three-dimensional structures of proteins, nucleic acids, and complex assemblies, and has also experienced significant growth in recent years, reaching over 150,000 entries.

To address the sequence-function gap many computational methods have been developed over the years. These methods typically aim to predict protein function for whole protein genes, but much work is also directed at the related problem of predicting function in a site- or domain-specific manner (that automatically generates functional hypothesis linked to residues, regions or domains)[17–20]. Traditional machine learning classifiers, such as support vector machines, random forests, and high-dimensional statistical methods like logistic regression have been used extensively for the protein function prediction problem, and have established that integrative prediction schemes can outperform homology-based function transfer[21,22]. Systematic benchmarking efforts, such as the Critical Assessment of Functional Annotation (CAFA1[23] & CAFA2[24]) and MouseFunc[25], have also played a key role in the development of these methods and have shown that integrative machine learning and statistical methods outperform traditional sequence alignment-based methods (e.g., BLAST)[22]. However, the performance of these methods is typically strongly affected by the quality of manually-engineered features constructed from either sequence or structure (features that rely heavily on heuristics that in turn require domain-expert knowledge, and in

some cases unstable assumptions, thresholds and preprocessing pipelines)[26]. Here, we focus on methods that can take as inputs sequence and features that are readily derived from sequence (such as predicted structure) and do not focus on, or compare to, the many methods that rely on protein networks like *GeneMANIA*[27], *Mashup*[28], *DeepNF*[29], and other integrative network prediction methods. We focus our study in this way to present a method that can be applied to very large volumes of sequence where many proteins are from unknown organisms lacking the required network data (and thus hope to address the critical need for these methods in metagenomic contexts).

In the last decade, deep learning approaches have achieved unprecedented increase in performance on a broad spectrum of problems ranging from learning protein sequence embeddings for contact map prediction[30] to predicting protein structure[31,32] and function[33]. In particular, Convolutional Neural Networks (CNN)[34], that have been state-of-the-art in computer vision, have also shown tremendous success in addressing problems in computational biology. They enabled task-specific feature extraction directly from protein sequence or its 3D structure overcoming the limitations of feature-based ML methods. The majority of sequence-based protein function prediction methods use 1D CNNs, or variations thereof, that search for recurring spatial patterns within a given sequence and converts them hierarchically into complex features using multiple convolutional layers.

Recent work has employed 3D CNNs to make predictions and extract features from protein structural data[35,36]. These methods take as input a 3D volumetric protein structure represented on a grid. Storing explicit 3D representations of protein structure at high resolution is not memory

efficient (most of the 3D space is unoccupied by protein structure); thus, in most cases, the 3D CNN would convolve over empty space which is somewhat inefficient. More recently, geometric deep learning methods[37] and more specifically Graph Convolutional Networks (GCNs)[38,39] have offered a way to overcome these limitations by generalizing convolutional operations on more natural graph-like molecular representations. Graph Convolutional Networks have shown tremendous success in various problems ranging from learning useful molecular fingerprints[40], to predicting biochemical activity of drugs[41], to protein interface prediction[42].

Here, we describe a method based on GCNs for functionally annotating protein sequences and structures that outperforms current methods and scales to the size of current repositories of sequence information. We model protein structures as graphs that are derived from protein contact maps (truncated residue-residue-pair distance maps). Residue-level sequence features together with contact maps, are fed into GCNs. The GCNs uses a deep architecture to further propagate residue-level features between residues at different proximity to each other in the protein contact graph to construct final protein-level feature representations that prove useful for protein function prediction. For learning sequence features we use Bidirectional Long Short-Term memory Language Model (LSTM-LM) pretrained on a corpus of around 2 million protein sequences. Our LM is trained to predict an amino-acid residue in the context of residues before and after it in a protein sequences. Using features from a pre-trained, task-agnostic LSTM LM as input to classification tasks has demonstrated tremendous success in many NLP[43] and biological problems[30]. We show that such features can significantly boost performance of GCN in function prediction task. Using LM features together with contact maps of experimental PDB structures we show that our method

outperforms sequences-only state-of-the-art methods. Moreover, by testing our method on *de novo* predicted structures we show that our method is robust to expected errors and can significantly de-noise predicted structures while still confidently predicting their functions.

In addition to improved accuracy of function predictions, our method also provides the ability to interpret predictions by analysing what the method is learning during the training. Designing transparent and explainable methods for interpreting classification results made by complex neural network classifiers has been a main focus of many recent studies[44–48]. For instance, a recent work in computer vision, uses Class Activation Maps (CAMs) on trained CNN-based architectures[44] to localize the most important regions in images relevant for making correct classification decisions[44]. Here, we propose a similar approach, adapted for GCNs, for detecting functional regions in proteins. For each PDB chain, CAM detects GO term-specific sites on its 3D structure by identifying residues relevant for making accurate GO term prediction. Here, we show that, for various GO terms, these functional sites often correspond to known binding regions, conserved regions or active sites. Interestingly, our model is not explicitly trained to predict functional sites, but instead such predictions stem solely from the CAM analysis of the graph convolution parameters of the trained model. Performing such analysis for identifying functional sites is also very efficient as it does not require any further training or modification of the model's architecture.

As we demonstrate below, analysing results from CAM approach and finding their biological meanings is challenging for some protein structures. However, in most cases, this approach can automatically navigate the hierarchy from small sites, to larger binding sites, to domains to

whole-protein localized functions. The site-specificity afforded by our function predictions is very valuable, especially in the case when predicting functions of poorly studied, unannotated proteins. Site-specific predictions provide first insights into the correctness of predictions and frames follow-up genetics or validation experiments (for example, highlighting the salient residues in the protein's 3D structure, detected by CAM, could serve as a potential validation technique to many biochemists and other domain experts studying predictions made by our model).

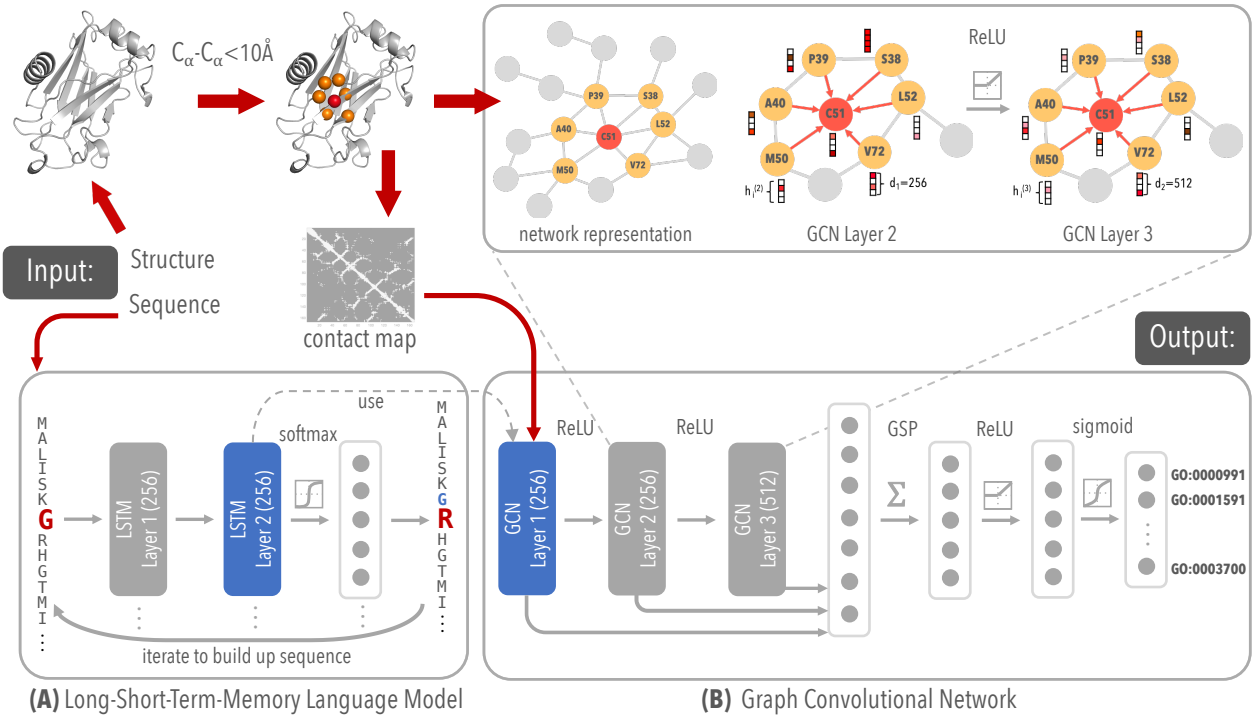

Figure 1: **Schematic overview of our method's pipeline.** (A) LSTM language model, pretrained on ~2 million Pfam protein sequences, used for extracting residue level features of PDB sequence. (B) our GCN with 3 graph convolutional layers for learning complex structure–to–function relationships.

RESULTS

Method overview

Our model takes as input a protein sequence and structure (in the form of a contact map) and outputs GO term probabilities. The method consists of two main parts: a LSTM LM that is learned from a very large corpus of protein sequences (Figure 1A), and a GCN that uses protein structure (Figure 1B). The first stage is an LSTM-LM (pretrained on the full set of protein domain sequences in the protein families database (Pfam)[49], see Methods) is used for extracting residue-level features from PDB sequences. It is composed of two stacked forward direction LSTM layers and two stacked reverse direction LSTM layers[50]. The residue-level hidden states from both LSTM blocks are concatenated and passed to a *softmax* layer for predicting probabilities of amino acids at each position of each sequence given the previous and following amino acid residues. The concatenated residue-level features constructed for PDB sequences, together with PDB contact maps, are used as an input for the second stage of our method. The second stage is a GCN used for predicting GO terms that uses three graph convolutional layers, a global average pooling layer followed by a number of fully connected layers to learn a complex structure-to-function relationship. The number of fully connected layers and final model dimensions are chosen based on the method's performance on the validation set (see Method section). During the training of GCN the parameters of the LM are frozen; i.e., the LSTM LM stage is only used as a PDB sequence feature extractor. One main advantage of our method, in comparison to, standard CNNs, is that it convolves features over residues that are distant in the primary sequence, but close to each other in the 3D space. Such an

operation, implemented here using graph convolution, leads to better protein feature representations and ultimately to more accurate function predictions. The effect of long-range connections on predictive performance of our method is shown in [**Supplementary Figure 1**]. Another advantage of our method is that LM sequence features boost the predictive power of our method compared to simplified residue feature representation [**Supplementary Figure 1**].

1 Evaluating our method on PDB structures

The performance of our method, computed based on the Area Under the Precision-Recall (AUPR) curve, averaged over all EC numbers and over all GO terms in all three branches of GO is shown in Figure 2a (see panel 1 for EC and panels 2-4 for GO). The performance is compared to 1) BLAST baseline, in which every test sequence receives GO terms that are transferred from the sequence in the training set with the score being the pairwise sequence identity (as done in CAFA1[23] & CAFA2[24]), as well as to the 2) state-of-the-art CNN trained only on the sequence data (see Methods section for the architecture details). Our method substantially outperforms both CNN and BLAST on the EC numbers (only the most specific EC numbers are considered in the training, i.e., leaf nodes in the EC tree), MF-GO and CC-GO, but not on BP-GO (see also average AUPR curves for EC and MF-GO in Figure 2c and Figure 2d).

We explored the performance of our method on individual GO terms and EC classes [**Supplementary Figures 2-5**]. We observe that for the majority of MF-GO and CC-GO terms, our method outperforms the sequence-only CNN method, indicating the importance of structure fea-

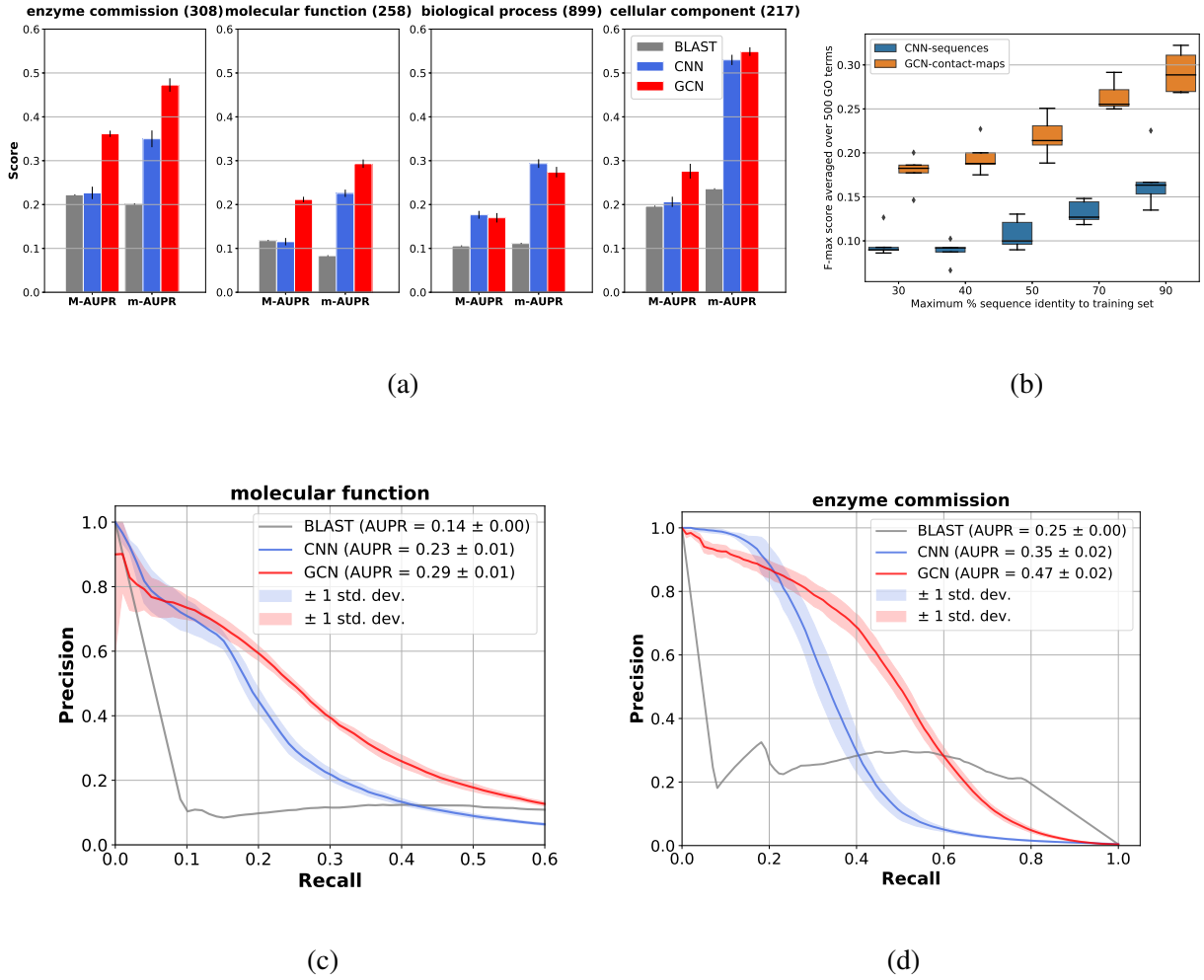

Figure 2: **Improved performance over GO terms in different ontologies and EC numbers.** (A) AUPR scores, summarized over all EC numbers/GO terms both under the micro-averaging (m-AUPR) and macro-averaging (M-AUPR), computed on the test set comprised of PDB chains chosen to have ≤ 30 % sequence identity to the PDB chains in the training set. The numbers in brackets indicates the number of EC classes and the number of GO terms in different ontologies used in the training of the model; (B) Distribution of F-max score averaged over all MF-GO terms grouped by maximum % sequence identity to the training set; Precision-recall curves for each method for MF-GO terms (C) and EC numbers (D); The curves are averaged over prediction results from 10 different separately trained GCN or CNN models.

tures in improving the classification performance. Our method performs better than sequence-only

CNN for more specific MF-GO and CC-GO terms with fewer training examples (see **Supplemen-**

tary Figures 6A,B). By looking at the individual CC-GO term performance, we demonstrate that our method outperforms CNN on almost all GO terms with average PDB chain length ≥ 200 (see **Supplementary Figure 6F**), illustrating the importance of encoding distant amino-acid contacts via the structure graph. This demonstrates the superiority of graph convolutions over sequence convolutions in constructing more accurate protein features. Specifically, in the case of long protein sequences, a CNN, with reasonable filter lengths, would most likely fail to convolve over residues at different ends of the long sequence, even after applying multiple consecutive CNN layers; wheres, GCN applied on contact maps would, in 3 layers, access feature information from the complete structure.

2 Evaluating our method on predicted structures

Here, we ask how well can our method tolerate the error in predicted structures. We demonstrate this for the Rosetta *de novo* prediction procedure and for another *de novo* deep-leaning-based, structure prediction method, [11]. We used Rosetta macromolecular modeling suite[51] and protein contact predictions from DeepMetaPSICOV contact predictor (DMPfold)[11] to fold sequences of ~500 experimentally annotated PDB chains and obtain the lowest energy decoy from folding. With respect to contact or distance map prediction (the relevant input feature here), *Rosetta* has the highest error of the three methods tested (Supplementary Figure showing TM scores). We construct two kinds of Cα–Cα contact maps for each PDB chain – one from its experimental (i.e., *NATIVE*) structure and one from the lowest-energy (i.e., *LE*) decoy.

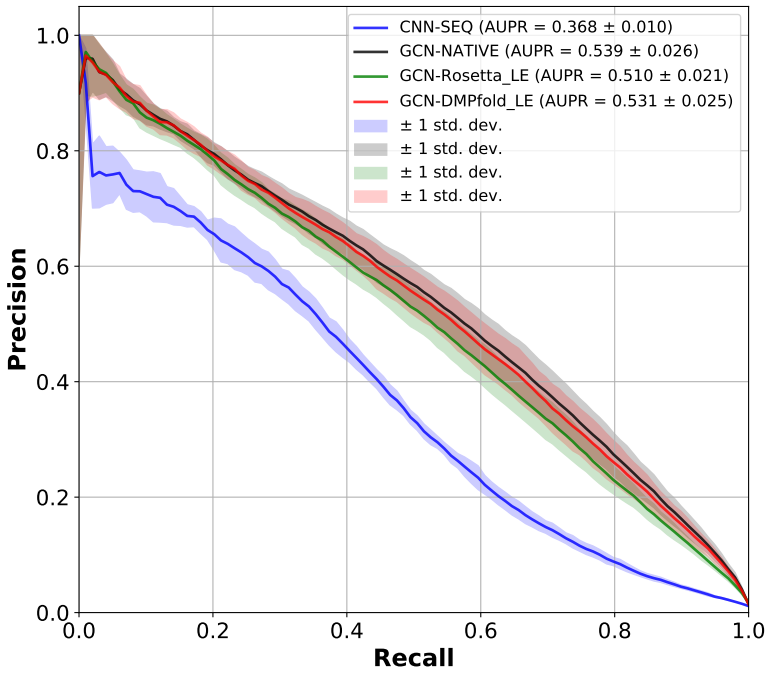

Figure 3: Precision-recall curves representing the performance on ~500 annotated test proteins obtained by using our CNN applied to sequences (blue), GCNN applied to NATIVE (black), Rosetta-predicted lowest energy (LE) (green) structures and DMPfold lowest energy (LE) structures (red). The curves are averaged over prediction results from 10 different separately trained GCN or CNN models.

We run our method on both predicted (Rosetta *de novo*) and native (derived from high quality experimental structures) contact maps and report the results together with results of the CNN applied only on sequences in Figure 3. We observe that our GCN model exhibits higher performance than that of the CNN even when accounting for error in predicted contact maps. Even though Rosetta-predicted structures often result in noisy contact maps, the fact that the performance of our method on the predicted LE structures is not drastically impaired can be attributed to the high denoising ability of the GCN implied by high correlation between GCN features extracted from NATIVE and LE contact maps (see **Supplementary Figure 7**).

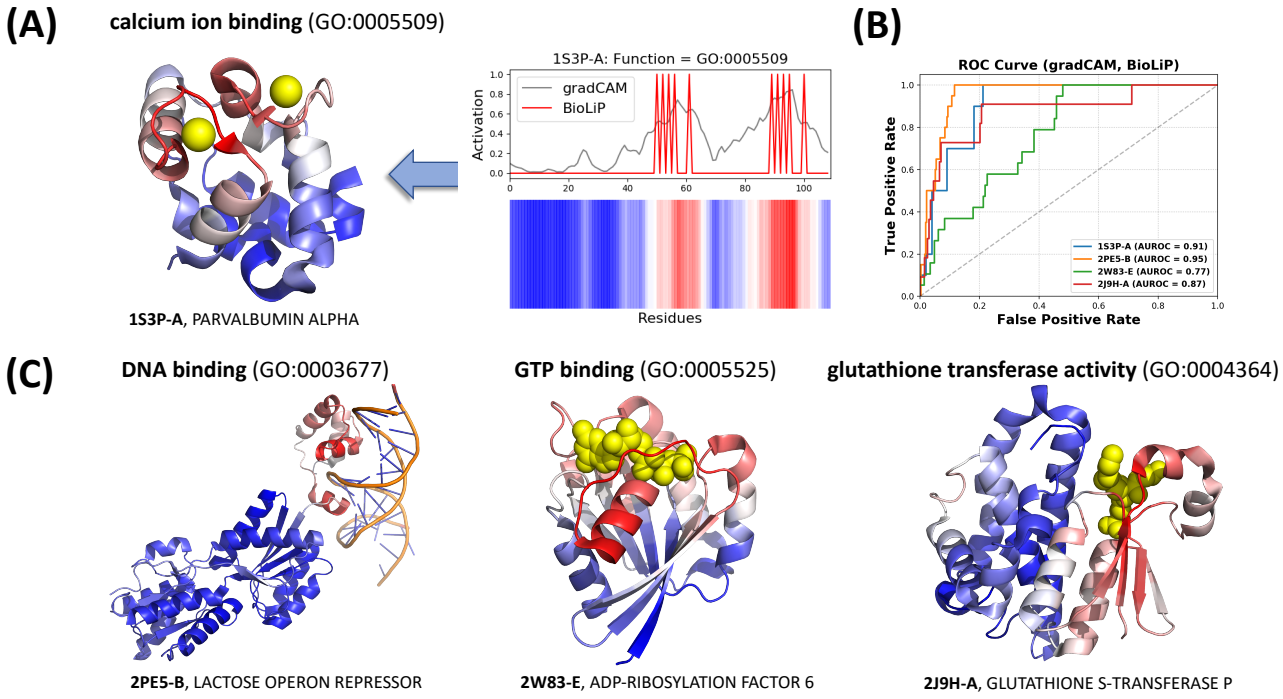

Figure 4: (A) An example of the gradient-weighted class activation map for 'Ca Ion Binding' (right) mapped onto the 3D structure of rat alpha-parvalbumin (PDB Id. = 1S3P), chain A (left), annotated with *calcium ion binding*. The two highest peaks in grad-CAM activation profile correspond to calcium binding regions. (B) ROC curves showing the overlap between grad-CAM activation profiles and binding sites, retrieved from the *BioLiP* database, computed for the PDB chains shown in panel C. (C) Examples of other PDB chains annotated with *DNA binding*, *GTP binding* and *glutathione transferase activity*. All residue are colored using gradient color scheme to match the grad-CAM activity profile, with more salient residues highlighted in red and less salient residues highlighted in blue.

3 From protein-level to residue-level prediction via class activation maps

Many proteins carry out their functions through a proxy of a few functionally important residues (e.g., active sites on an enzyme, or ligand-binding sites on a protein, and protein-protein interactions); this is especially the case for site-specific functions in MF branch of the GO. Designing ML methods for identifying such functional residues have been a subject of many recent studies[17–20].

Much recent work in ML has provided several new approaches for localizing signal to regions of the input feature space that lead to a positive predictions, giving a means of interpreting decisions made by neural networks[52,53]. In computer vision, these methods determine the regions of images that lead to positive object classifications; in NLP these methods lead to identification of sub-regions of documents[54]. Here, we use grad-CAMs[44], adapted for post-training analysis of GCNs, to determine a region of a protein that leads to the correct prediction of its GO function. For each GO term, the grad-CAM technique generates an activation map over the input data, in our case a sequence of residues and the contact or distance map, indicating the importance of each residue to the GO term classification decision (see example of CAM and its corresponding heatmap over sequence in Fig. 4A, **right**). It does so by first computing the contribution of each graph convolutional feature map of the model (trained on the MF-GO dataset) to the GO term prediction, and then by summing the feature maps with positive contributions to obtain a final residue-level activation map (see Method Section). For site-specific functions, this method identifies correct function regions and we provide several examples where we automatically and correctly identify functional sites for several functions where binding sites are known (see Fig. 4). Fig. 4A shows a grad-CAM for a *calcium ion binding (GO:0005509)* of alpha-parvalbumin protein (PDB id: 1S3P). The two highest peaks in the grad-CAM correspond to the binding regions in the 3D structure of the protein (Fig. 4A, **left**). Indices of the calcium binding residues of the 1S3P protein were retrieved from the BioLiP database[55] and compared to the residues identified by our method. ROC curve computed between the binary profile representing binding sites from BioLiP (shown in red) and the gradCAM profile (shown in green) in Fig. 4A, **right** are depicted in Fig. 4B. High area under the ROC curve

indicates high correspondence between binding sites and our predictions. Similar correspondence with BioLiP is observed for several other functions including *DNA binding (GO:0003677), GTP binding (GO:0005525)* and *glutathione transferase activity (GO:0004364)* (see Fig. 4C and their corresponding ROC curves in Fig. 4 B).

Our systematic analysis of grad-CAMs against BioLiP database reveal that the highest performing group of GO terms are related to functions with known site-specific mechanisms or site specific underpinnings, like metal binding. Therefore, we provide a systematic analysis of grad-CAMs for GO terms related to *metal binding* using also information from MetalPDB[56] database. We depict examples (with high AUROC scores) where CAMs correctly identify binding regions for *calcium ion binding (GO:0005509)*, *zinc ion binding (GO:0008270)* and *copper ion binding (GO:0005507)* (see **Supplementary Figure**).

There are also a large number of high quality protein structures in the PDB that lack functional annotation, or that have only high-level or incomplete annotation. This partial lack of annotation results from unbiased structural genomics projects, proteins having associations with processes but no function, and the fact that proteins can have multiple functions. Here we apply our method to the full PDB to 1) annotate unannotated chains, 2) complete partial annotations, and 3) look for new functions hiding in annotated proteins (find more moonlighting proteins). We present Supplementary Table 2 which shows the number and types of new annotations produced for the full PDB, and Supplemental file 1 which holds predictions and salience maps for all predictions produced for the PDB.

4 Discussion

In this work, we proposed a novel deep learning-based method for predicting protein function from both protein sequences and contact map representations of protein structures. Our method, trained on protein structures from PDB, is very efficient, and it can predict both GO terms and EC numbers of proteins and improves over state-of-the-art sequence-based methods on majority of GO terms especially. Features learned from protein sequences by the LSTM Language Model and from contact maps by the GCN lead to substantial improvements in protein function prediction accuracy, which could enable novel protein function discoveries. One important advantage of our method is that it makes function predictions that go beyond homology-based transfer by extracting local sequence and global structure features that would most likely be neglected by homology-based methods like BLAST (reflected in the substantial difference in the function prediction accuracy between our method and BLAST)[23].

Comparable performance of our method between Rosetta-predicted and their corresponding experimentally determined structures, which can be attributed to high denoising power of our method, indicates that our method can also be reliably used in predicting functions of proteins with computationally inferred structures. This opens a door for characterizing many proteins lacking experimentally determined 3D structures and the contents of many databases with available predicted structures (e.g., homology-based Swiss-Model[12], and ModBase[57] could be used for expanding the train set and improving predictive power of the model). The more extensive use of homology models allowed by the denoising properties of our network architecture will be a subject of future

study.

While this paper mainly focuses on introducing efficient and accurate function prediction model, it also provides means of interpreting prediction results. We demonstrate, on multiple different GO terms, that CAMs identify structurally-meaningful protein regions encompassing functionally relevant residues (e.g., ligand-binding residues). For some PDB chains, the accuracy at which the CAM identifies binding residues is quite remarkable, especially given the fact the model is not principally designed to predict this, and that the ligand-binding information was not given to the model *a priori*. However, the main disadvantage of considering this to be a *site-specific function prediction method* is in the multiple different meanings of CAMs. Specifically, for some GO terms related to "binding", CAMs do not necessarily identify binding residues/regions; instead, they identify regions of residues that are conserved among the sequences annotated with the same function. The most interesting example demonstrating this property is *maltose binding (GO:1901982)* (see **Supplementary Figure 8**). In this example, the salient residues are far from the residues binding maltose in the 3D structure; but, by looking at a few non-redundant PDB sequences annotated with *maltose binding*, we find that the CAM always identifies the same residues that are conserved across the sequences. These can be explained with the fact that any neural network, including ours, would always tend to learn the most trivial features that lead to the highest accuracy (AUPR=1 for *maltose binding* for both CNN and GCN)[58,59]. Despite these limitations, with the appropriate balancing and control of the bias in the training set (see **Supplementary Figure 9** showing the distribution of PDB chains belonging to different folds and how this correlates with grad-CAM performance), this approach has a huge potential in advancing site-specific

function prediction.

After the culmination of much effort two key problems in computational biology, *protein structure prediction* and *protein function protein*, are linked together by the described methods. Deep learning together with increasing amount of available sequence and structural data being generated each day has a potential to meet the annotation challenges posed by ever increasing volumes of genomic sequence, offering several new methods for interpreting protein biodiversity.

Methods

Construction of contact maps. We collect 3D atomic coordinates of proteins from the Protein Data Bank (PDB)[60]. As the PDB contains extensive redundancy in terms of both sequence and structure, we remove identical and similar sequences from our set of annotated PDB chains. We create a non-redundant set by selecting PDB chains that are not identical to any other PDB chain in the set. To do so, we first cluster all PDB chains (for which we were able to retrieve contact maps) by `blastclust` at 100% sequence identity (i.e., number of identical residues out of the total number of residues in the sequence alignment). Then, from each cluster we select a "representative" PDB chain as a PDB chain which is annotated (i.e., has at least one GO term in at least one of the 3 ontologies) and which is of high quality (has a high resolution structure). Each protein in the set is described by an ordered list of amino acid residues represented by thier X, Y and Z coordinates in angstrom (Å). To construct contact maps we use the α-carbon (Cα) atom type and consider two resides to be in contact if the distance between their corresponding Cα atoms is less than 10Å. We refer to this type of contact maps as Cα–Cα. We have also considered two other

criteria for contact map construction. Two residues are in contact: 1) if the distance between any of their atoms is less than 6.5Å (we refer to this type of contact maps as *ANY-ANY*) and 2) if the distance between their Rosetta neighbor atoms is less than sum of the neighbor radii of the amino acid pair (we refer to this type of contact maps as *NBR-NBR*). Rosetta neighbor atoms are defined as the β-carbon (Cβ) for all amino acids except glycine where the α-carbon is used. An amino acids neighbor-radius describes a potential interaction sphere that would be swept out by the side amino acid side-chain as it samples all possible conformations. Neighbor-neighbor contact maps are therefore more indicative of side-chain–side-chain interactions than Cα–Cα maps. We have also experimented with different cut-off thresholds for Cα–Cα and $ANY - ANY$ contact maps. We found that our method produced the best results with Cα–Cα and 10Å cut-off.

Function annotations of PDB chains. In the training of our models we use two sets of function labels: 1) Gene Ontology (GO)[6] terms and 2) enzyme commission (EC) numbers [7]. GO terms are hierarchically organized into 3 different onotologies – molecular function (MF), biological process (BP) and cellular component (CC). We train our models to predict GO terms separately for each ontology. The summary of GO identifiers as well as EC numbers for each PDB chain were retrieved from SIFTS[61] (Structure integration with function, taxonomy and sequence) database. SIFTS transfers annotation to PDB chain level via residue-level mapping between UniProt Knowledgebase (UniProtKB) and PDB entries. All the annotation files were retrieved from SIFTS database with PDB release 08.19 and UniPort release 2019.02. We consider annotations that are: 1) not electronically inferred (non-IEA), specifically, we consider GO terms with the following evidence codes: EXP, IDA, IPI, IMP, IGI, IEP, TAS and IC and 2) electronically inferred (IEA). Further-

more, we focus only on specific MF-, BP- and CC-GO terms that have enough training examples from the non-redundant training set (see the section above). That is, we select only GO terms that annotate ≥ 30 (for MF and CC) and ≥ 50 (for BP) non-redundant PDB chains. We retrieved enzyme classes for sequences and PDB structures from the lowest level (most specific level) of EC tree. The number of GO terms and EC classes in each ontology is represented in Table [**Supplementary Table 1**].

Top 500 Rosetta-predicted structures. The initial set of benchmark structures used here was Jane and Dave Richardson's "Top 500" dataset[62]. It is a set of hand curated, high quality (the top 500 best), protein structures that were chosen for their fit to their completeness, how well they fit the experimental data, and lack of high energy structural outliers (bond angle and bond length deviations). This set has been used in the past for fitting Rosetta energy/score terms and numerous other structural-bioinformatics validation tasks. Unfortunately, the structures in this set lacked sufficient annotations (many of these structures were the results of structural genomics efforts and had no, or only high level, annotations in GO and Brenda) Accordingly, we choose an additional 350 sequences from the PDB. These additional high quality benchmark structures were chosen by taking 119K chains with function annotations and filtering them with the PISCES Protein Sequence Culling Server[63] with the criteria below. That left us with 1606 SIFTS annotated chains from which we randomly selected 350. These proteins were then excluded from all phases of model training.

Convolutional neural network. Convolutional neural networks (CNNs) have shown tremendous success in extracting information from sequence data and making highly accurate predictive models. Their success can be attributed to convolutional layers with highly reduced number of learnable

parameters which allow multi-level and hierarchical feature extraction. In the last couple of years a large body of work has been published covering various applications of CNNs, such as prediction of protein functions[33] and subcellular localization[64], prediction of effects of noncoding-variants[65] and protein fold recognition[66]. Here we describe in detail the architecture of the convolutional neural network used in our comparison study. We represent a protein sequence with L amino acid residues as a features matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_L] \in \{0, 1\}^{L \times c}$, where $c = 26$ dimensions (25 residues plus the gap symbol) are used as a one-hot indicator, $\mathbf{x}_i \in \{0, 1\}^c$, of the amino acid residue at position i in the sequence. This representation is fed into a convolution layer which applies a one-dimensional convolution operation with a specified number of kernels (weight matrices or filters), f_n, of certain length, f_l, and all outputs are then transformed by the rectified linear activation function ($ReLU$), which sets values below 0 to 0, i.e., $ReLU(x) = max(x, 0)$. It consists of two convolutional layers followed by a global max pooling layer and a dense layer with *sigmoid* activation function for predicting probabilities of GO terms or *softmax* activation functions for predicting EC enzyme classes. In the first convolution layer, we use $f_n = 360$ filters of different lengths: 120 filters of length $f_l = 5$, 100 filters of length $f_l = 10$, 80 filters of length $f_l = 15$ and 60 filters of length $f_l = 20$. After concatenating the results of the first CNN layer we obtain $L \times 360$ dimensional feature map for each sequence. Using filters of variable lengths ensures extraction of complementary information from protein sequences. The second convolutional layer has $f_n = |GO|$ number of filters for GO terms (or $f_n = |EC|$ for EC) classification. The length of the filters is set to $f_l = 3$ for all filters in the second convolutional layer. The architecture of the CNN used in our study is shown in [**Supplementary Note 1**].

LSTM language model for learning residue-level features. We use an approach similar to *Bepler & Berger*[30]. We train a bidirectional LSTM language model on ~2,000,000 Pfam sequences. The sequences are represented using 1-hot encoding (see above). The LM architecture is comprised of two stacked forward and two stacked backward LSTM layers with 512 units each (see Fig. 1A for forward direction LSTM). The LSTM LM model is trained for 5 epochs using ADAM optimizer with learning rate $lr = 0.001$ and batch size of 128.

The residue-level features, extracted from the final LSTM layers' hidden states, $\mathbf{H}^{LM} = [\overrightarrow{\mathbf{H}}, \overleftarrow{\mathbf{H}}]$, are combined together with 1-hot representation of sequences, $\mathbf{X}$, through learnable non-linear mapping:

$$\mathbf{X}^{input} = ReLU(\mathbf{H}^{LM}\mathbf{W}^{LM} + \mathbf{X}\mathbf{W}^{X} + \mathbf{b}) \tag{1}$$

where $\mathbf{X}^{input}$ is the final residue-level feature representation passed to the fist GCN layer, $\mathbf{H}^{(0)} = \mathbf{X}^{input}$ (see equation 4). The parameters, $\mathbf{W}^{LM}$, $\mathbf{W}^{X}$ and $\mathbf{b}$ are trained together with the parameters of the GCN. All the parameters of the LSTM LM are frozen during the training. See [**Supplementary Note 2**] summarizing LSTM-LM architecture used in our study.

Graph convolutional network. Graph Convolutional Networks (GCNs) have recently been shown to be powerful methods for extracting features from data that is naturally represented as one or more graphs[37]. This makes GCN a suitable candidate method for extracting features from a protein by taking into account their graph-based structure of amino acids represented by contact maps. In particular, they have achieved a remarkable performance in classifying documents in citation

networks[39], modeling and predicting chemical properties of molecules[40,41,67] and protein interface prediction with applications in drug discovery and design[42]. Here, we propose our model based on the work of *Kipf & Welling*[39]. A protein graph can be represented by an adjacency matrix (also termed contact map), $\mathbf{A} \in \mathbb{R}^{L \times L}$, encoding connections between its L residues, and a residue-level feature matrix, $\mathbf{X} \in \mathbb{R}^{L \times c}$.

We explore different residue-level feature representations including the one-hot encoding representation of residues as in the CNN ($c = 26$), LSTM language model ($c = 512$, i.e., the concatenated output forward and reverse LSTM layers), and no sequence features.[1] We refer to the last case as function prediction from protein fold only; see [**Supplementary Figure 1**].

The graph convolution takes both adjacency matrix, $\mathbf{A}$ and residue-level embeddings from the previous layer, $\mathbf{H}^{(l)} \in \mathbb{R}^{L \times c_l}$ and outputs the residue-level embeddings in the next layer, $\mathbf{H}^{(l+1)} \in \mathbb{R}^{n \times c_{l+1}}$:

$$\mathbf{H}^{(l+1)} = GC(\mathbf{A}, \mathbf{H}^{(l)}), \tag{2}$$

where $\mathbf{H}^{(0)} = \mathbf{X}$, and c_l and c_{l+1} are residue embedding dimensions for layers l and $l+1$, respectively. Concretely, we use the formulation of *Kipf & Welling*[39]:

$$GC(\mathbf{A}, \mathbf{H}^{(l)}) = ReLU(\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}), \tag{3}$$

[1]To be able to run GCN, in this case, feature matrix is substituted with an identity matrix, i.e., $\mathbf{X} = \mathbf{I}^L$.

where $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_L$ is the adjacency matrix with added self-connections represented by the identity matrix $\mathbf{I}_L \in \mathbb{R}^{L \times L}$; $\widetilde{\mathbf{D}}$ is the diagonal degree matrix with entries $\widetilde{\mathbf{D}_{ii}} = \sum_{j=1}^{L} \widetilde{\mathbf{A}}_{ij}$, and $\mathbf{W}^{(l)} \in \mathbb{R}^{c_l \times c_{l+1}}$ is a trainable weight matrix for layer $l+1$. To keep the residues' features on the same scale after every convolutional layer the adjacency matrix is first symmetrically normalized, hence the term $\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}$. Equation 3 updates features of each residue by a weighted sum of features of the residue in its one-hop neighborhood (adding self-connections ensures that the residue's own features are also included in the sum) (see also inset in Fig. 1).

Given that we are classifying individual protein graphs with different number of residues, we use several layers, $N_l = 3$, of graph convolutions. The final protein representation is obtain by first concatenating features from all layers into a single feature matrix, i.e., $\mathbf{H} = [\mathbf{H}^{(1)}, \ldots, \mathbf{H}^{(N_l)}] \in \mathbb{R}^{L \times \sum_{l=1}^{L} c_l}$, and then by performing a global pooling layer after which we obtain a fixed vector representation of a protein structure, $\mathbf{h}^{(pool)} \in \mathbb{R}^{\sum_{l=1}^{L} c_l}$. The global pooling is obtained by a sum operator over L residues:

$$\mathbf{h}^{(pool)} = \sum_{i=1}^{n} \mathbf{H}_{i,:} \tag{4}$$

We then use two dense layers to learn complex protein-to-function relations with $ReLU$ activation function in the first layer and *sigmoid* (for predicting GO terms) and *softmax* (for predicting EC) activation function in the second layer. The second layer outputs probability vector $\hat{\mathbf{y}}$ of dimension $|GO|$ for predicting probabilities of GO terms, and $|EC|$ for predicting probabilities of EC classes.

Model training and hyperparameters. To account for imbalanced label problem, both CNN and GCN are trained to minimize *weighted binary cross-entropy* cost function that gives higher weights to GO term with fewer training examples:

$$\mathcal{L}(\mathbf{\Theta}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{|GO|} w_j y_{ij}\log(\hat{y}_{ij}) + (1-y_{ij})\log(1-\hat{y}_{ij}), \tag{5}$$

where $\mathbf{\Theta}$ is the set of all parameters in all layers to be learned; $w_j = \frac{N}{N_j^+}$ is class weight for function j, with N_j^+ being the number of positive examples associated with function j; N is the total number of samples and $|GO|$ is the total number of functions (i.e., GO terms); y_{ij} is the true binary indicator for sample i and function j (i.e., $y_{ij} = 1$ if sample i is annotated with function j and $y_{ij} = 0$ otherwise) and $\hat{y}_{ij}$ is the predicted (sigmoid) probability that sample i is annotated with function j.

All hyperparameters are determined through grid search based on the model's performance on the validation set. The validation set is comprised of 10% randomly chosen samples from the training set. To avoid overfitting, we use early stopping with $patience = 5$ (i.e., we stop training if validation loss does not improve in 5 epochs). We use ADAM optimizer[68] with learning rate $lr = 0.0001, \beta_1 = 0.95$ and $\beta_2 = 0.99$ and batch size of 64. The number of epochs is 200. Both GCN and CNN are implemented to deal with variable length sequences, by performing sequence/contact map padding only on the batch level. See [**Supplementary Note 3**] for all the optimal hyperparameters used in our study.

We partition the non-redundant set of PDB chains into train, validation and test such that

for each function we have at least 30 training examples and at least 3 test examples. The test set is chosen to be no more than 30% sequence identical to the training set (and typically much less, or unalignable). We perform experiments with different thresholds. See Figure 2b showing performance of our method for different sequence identity thresholds. In all our experiments we trained on both non-IEA and IEA PDB chains (see **Supplementary Figure 1**), but the the test set, composed of only experimentally annotated PDB chains (non-IEA), is always kept fixed. See Table [**Supplementary Table 1**].

In all our experiments we train different models and the final results are averaged over predictions made by the 10 different models.

Residue-level annotations. We use a method based on Gradient-weighted Class Activation Map (grad-CAM)[44] to localize function predictions on a protein structure (i.e., to find residues with highest contribution to a specific function). grad-CAM is a class-discriminative localization technique that provides visual explanations for predictions made by CNN-based models. Motivated by its huge success in image analysis, we use grad-CAM to identify important, function-specific residues in a protein structure. In a grad-CAM approach, we first compute the contribution of each filter, k, in the last convolutional layer to the prediction of function label l by taking derivative of the output of the model for function l, y^l, with respect to feature map $\mathbf{F}_k \in \mathbb{R}^L$ over the whole sequence of length L:

$$w_k^l = \sum_{i=1}^{L} \frac{\partial y^l}{\partial F_{k,i}} \tag{6}$$

where w_k^l represent the importance of feature map k for predicting function l, obtained by summing the contributions from each individual residue. Finally, we obtain the function-specific heat-map in a residue space by making the weighted sum over all feature maps in the last convolutional layer:

$$CAM^l[i] = ReLU(\sum_k w_k^l F_{k,i}) \tag{7}$$

where $ReLU$ function ensures that only features with positive influence on the functional label are preserved; $CAM^l[i]$ - indicates the relative importance of residue i to function l.

To account for variations in grad-CAM between different initializations of the same model architecture, we report the grad-CAMs averaged over an ensemble of 10 different separately trained models.

The advantage of grad-CAM is that it does not require re-training or changes in the architecture of the model which makes is computationally efficient and directly applicable to our models.

References

1. Mitchell, A. L. *et al.* InterPro in 2019: improving coverage, classification and access to protein sequence annotations. *Nucleic Acids Research* **47**, D351–D360 (2018).

2. Jones, D. T. & Cozzetto, D. DISOPRED3: precise disordered region predictions with annotated protein-binding activity. *Bioinformatics* **31**, 857–863 (2014).

3. Dawson, N. L. *et al.* CATH: an expanded resource to predict protein function through structure and sequence. *Nucleic Acids Research* **45**, D289–D295 (2016).

4. Gerstein, M. How representative are the known structures of the proteins in a complete genome? A comprehensive structural census. *Folding and Design* **3**, 497 –512 (1998).

5. Vogel, C., Berzuini, C., Bashton, M., Gough, J. & Teichmann, S. A. Supra-domains: Evolutionary units larger than single protein domains. *Journal of Molecular Biology* **336**, 809 – 823 (2004).

6. Ashburner, M. *et al.* Gene Ontology: tool for the unification of biology. *Nature News* (2000).

7. Chang, A., Schomburg, I., Jeske, L., Placzek, S. & Schomburg, D. BRENDA in 2019: a European ELIXIR core data resource. *Nucleic Acids Research* **47**, D542–D549 (2018).

8. Bairoch, A. The ENZYME database in 2000. *Nucleic Acids Research* **28**, 304–305 (2000).

9. Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y. & Morishima, K. KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Research* **45**, D353–D361 (2016).

10. Ovchinnikov, S. *et al.* Protein structure determination using metagenome sequence data. *Science* **355**, 294–298 (2017).

11. Greener, J. G., Kandathil, S. M. & Jones, D. T. Deep learning extends de novo protein modelling coverage of genomes using iteratively predicted structural constraints. *Nature communications* **10**, 1–13 (2019).

12. Waterhouse, A. *et al.* SWISS-MODEL: homology modelling of protein structures and complexes. *Nucleic Acids Research* **46**, W296–W303 (2018).

13. Vallat, B., Webb, B., Westbrook, J., Sali, A. & Berman, H. M. Archiving and disseminating integrative structure models. *Journal of Biomolecular NMR* **73**, 385–398 (2019).

14. Webb, B. & Sali, A. *Protein Structure Modeling with MODELLER*, 1–15 (Springer New York, New York, NY, 2014).

15. Shigematsu, H. Electron cryo-microscopy for elucidating the dynamic nature of live-protein complexes. *Biochimica et Biophysica Acta (BBA) - General Subjects* 129436 (2019).

16. García-Nafría, J. & Tate, C. G. Cryo-electron microscopy: Moving beyond x-ray crystal structures for drug receptors and drug development. *Annual Review of Pharmacology and Toxicology* **60**, null (2020).

17. Koo, D. C. E. & Bonneau, R. Towards region-specific propagation of protein functions. *Bioinformatics* **35**, 1737–1744 (2018).

18. Torng, W. & Altman, R. B. High precision protein functional site detection using 3D convolutional neural networks. *Bioinformatics* **35**, 1503–1512 (2018).

19. Schug, J., Diskin, S., Mazzarelli, J., Brunk, B. P. & Stoeckert, C. J. Predicting gene ontology functions from prodom and cdd protein domains. *Genome Research* **12**, 648–655 (2002).

20. Das, S. *et al.* Functional classification of CATH superfamilies: a domain-based approach for protein function annotation. *Bioinformatics* **31**, 3460–3467 (2015).

21. Guan, Y. *et al.* Predicting gene function in a hierarchical context with an ensemble of classifiers. *Genome biology* **9**, S3 (2008).

22. Wass, M. N., Barton, G. & Sternberg, M. J. E. CombFunc: predicting protein function using heterogeneous data sources. *Nucleic Acids Research* **40**, W466–W470 (2012).

23. Radivojac, P. *et al.* A large-scale evaluation of computational protein function prediction. *Nature Methods* **10**, 221–227 (2013).

24. Jiang, Y., Oron, T. R., Clark, W. T. *et al.* An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biology* **17**, 184 (2016).

25. Peña-Castillo, L., Tasan, M., Myers, C. L. *et al.* A critical assessment of mus musculus gene function prediction using integrated genomic evidence. *Genome Biology* **9**, S2 (2008).

26. Cozzetto, D., Minneci, F., Currant, H. & Jones, D. T. FFPred 3: feature-based function prediction for all Gene Ontology domains. *Scientific Reports* **6**, 31865 (2016).

27. Mostafavi, S., Ray, D., Warde-Farley, D. *et al.* GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biology* **9**, S4 (2008).

28. Cho, H., Berger, B. & Peng, J. Compact integration of multi-network topology for functional analysis of genes. *Cell Systems* **3**, 540–548 (2016).

29. Barot, M., Gligorijević, V. & Bonneau, R. deepNF: deep network fusion for protein function prediction. *Bioinformatics* **34**, 3873–3881 (2018).

30. Bepler, T. & Berger, B. Learning protein sequence embeddings using information from structure. In *International Conference on Learning Representations* (2019).

31. AlQuraishi, M. End-to-end differentiable learning of protein structure. *Cell Systems* **8**, 292–301.e3 (2019).

32. Wang, S., Sun, S., Li, Z., Zhang, R. & Xu, J. Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLOS Computational Biology* **13**, 1–34 (2017).

33. Kulmanov, M., Khan, M. A. & Hoehndorf, R. DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics* **34**, 660–668 (2017).

34. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436 (2015).

35. Jiménez, J., Doerr, S., Martínez-Rosell, G., Rose, A. S. & De Fabritiis, G. DeepSite: protein-binding site predictor using 3D-convolutional neural networks. *Bioinformatics* **33**, 3036–3042 (2017).

36. Amidi, A. *et al.* Enzynet: enzyme classification using 3d convolutional neural networks on spatial representation. *PeerJ* **6**, e4750 (2018).

37. Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A. & Vandergheynst, P. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine* **34**, 18–42 (2017).

38. Henaff, M., Bruna, J. & LeCun, Y. Deep convolutional networks on graph-structured data. *CoRR* **abs/1506.05163** (2015).

39. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations (ICLR)* (2017).

40. Duvenaud, D. *et al.* Convolutional networks on graphs for learning molecular fingerprints. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, 2224–2232 (MIT Press, Cambridge, MA, USA, 2015).

41. Coley, C. W., Barzilay, R., Green, W. H., Jaakkola, T. S. & Jensen, K. F. Convolutional embedding of attributed molecular graphs for physical property prediction. *Journal of chemical information and modeling* **57**, 1757–1772 (2017).

42. Fout, A., Byrd, J., Shariat, B. & Ben-Hur, A. Protein interface prediction using graph convolutional networks. In Guyon, I. *et al.* (eds.) *Advances in Neural Information Processing Systems 30*, 6530–6539 (Curran Associates, Inc., 2017).

43. Peters, M. *et al.* Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237 (Association for Computational Linguistics, New Orleans, Louisiana, 2018).

44. Selvaraju, R. R. *et al.* Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 618–626 (2017).

45. zu Belzen, J. U. *et al.* Leveraging implicit knowledge in neural networks for functional dissection and engineering of proteins. *Nature Machine Intelligence* **1**, 225 (2019).

46. Zołna, K., Geras, K. J. & Cho, K. Classifier-agnostic saliency map extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 10087–10088 (2019).

47. Ribeiro, M. T., Singh, S. & Guestrin, C. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence (AAAI)* (2018).

48. Adebayo, J. *et al.* Sanity checks for saliency maps. In Bengio, S. *et al.* (eds.) *Advances in Neural Information Processing Systems 31*, 9505–9515 (Curran Associates, Inc., 2018).

49. Finn, R. D. *et al.* Pfam: the protein families database. *Nucleic Acids Research* **42**, D222–D230 (2013).

50. Graves, A. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* (2013).

51. Leaver-Fay, A. *et al.* Rosetta3: an object-oriented software suite for the simulation and design of macromolecules. In *Methods in enzymology*, vol. 487, 545–574 (Elsevier, 2011).

52. Pope, P. E., Kolouri, S., Rostami, M., Martin, C. E. & Hoffmann, H. Explainability methods for graph convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

53. Montavon, G., Samek, W. & Müller, K.-R. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing* **73**, 1 – 15 (2018).

54. Denil, M., Demiraj, A., Kalchbrenner, N., Blunsom, P. & de Freitas, N. Modelling, visualising and summarising documents with a single convolutional neural network. *arXiv preprint arXiv:1406.3830* (2014).

55. Yang, J., Roy, A. & Zhang, Y. BioLiP: a semi-manually curated database for biologically relevant ligand-protein interactions. *Nucleic Acids Research* **41**, D1096–D1103 (2012).

56. Putignano, V., Rosato, A., Banci, L. & Andreini, C. MetalPDB in 2018: a database of metal sites in biological macromolecular structures. *Nucleic Acids Research* **46**, D459–D464 (2017).

57. Pieper, U. *et al.* ModBase, a database of annotated comparative protein structure models and associated resources. *Nucleic Acids Research* **42**, D336–D346 (2013).

58. Geirhos, R. *et al.* Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations* (2019).

59. Ilyas, A. *et al.* Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175* (2019).

60. Gilliland, G. *et al.* The Protein Data Bank. *Nucleic Acids Research* **28**, 235–242 (2000).

61. Gutmanas, A. *et al.* SIFTS: updated Structure Integration with Function, Taxonomy and Sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins. *Nucleic Acids Research* **47**, D482–D489 (2018).

62. Lovell, S. C. *et al.* Structure validation by C_α geometry: ϕ,ψ and C_β deviation. *Proteins: Structure, Function, and Bioinformatics* **50**, 437–450 (2003).

63. Wang, G. & Dunbrack, J., Roland L. PISCES: a protein sequence culling server. *Bioinformatics* **19**, 1589–1591 (2003).

64. Nielsen, H., Almagro Armenteros, J. J., Sønderby, C. K., Sønderby, S. K. & Winther, O. DeepLoc: prediction of protein subcellular localization using deep learning. *Bioinformatics* **33**, 3387–3395 (2017).

65. Zhou, J. & Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning-based sequence model. *Nature Methods* **12**, 931–4 (2015).

66. Hou, J., Adhikari, B. & Cheng, J. DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics* **34**, 1295–1303 (2017).

67. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural message passing for quantum chemistry. In Precup, D. & Teh, Y. W. (eds.) *Proceedings of the 34th International Conference on Machine Learning*, vol. 70 of *Proceedings of Machine Learning Research*, 1263–1272 (PMLR, International Convention Centre, Sydney, Australia, 2017).

68. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (2015).

Acknowledgements The project is partly funded by the Polish National Agency for Academic Exchange

grant PPN/PPO/2018/1/00014 to TK. RB, VG, PDR, DB and JKL are supported by Simons Foundation funding to the Flatiron Institute. KC is partly supported by Samsung AI and Samsung Advanced Institute of Technology.

Competing Interests The authors declare that they have no competing financial interests.

Correspondence Correspondence and requests for materials should be addressed to: *vgligorijevic@flatironinstitute.org, rb133@nyu.edu*