A Graphic Encoding Method for Quantitative Classification of Protein Structure and Representation of Conformational Changes

Hector Carrillo-Cabada, Jeremy Benson, Asghar M. Razavi, Brianna Mulligan,
Michel A. Cuendet, Harel Weinstein, Michela Taufer, and Trilce Estrada

Abstract—In order to successfully predict a proteins function throughout its trajectory, in addition to uncovering changes in its conformational state, it is necessary to employ techniques that maintain its 3D information while performing at scale. We extend a protein representation that encodes secondary and tertiary structure into fix-sized, color images, and a neural network architecture (called GEM-net) that leverages our encoded representation. We show the applicability of our method in two ways: (1) performing protein function prediction, hitting accuracy between 78 and 83%, and (2) visualizing and detecting conformational changes in protein trajectories during molecular dynamics simulations.

Index Terms—Protein function prediction, molecular encoding, graphic representation, neural networks.

1 Introduction

The functions carried out by proteins in biological systems depend on the specific folding of their amino acid sequence into 3D structures [1], [2]. The architecture of these structural folds is encoded in the chain of the amino acids (also called residues) connected by their chemical (peptide) bonds. While the forces underlying the interactions of residues with each other and with the environment (solvent, membrane, ligands, etc.) are well understood, predicting the fold adopted by a specific amino acid sequence and its subsequent dynamics remains a major challenge [3], as does the identification of the specific mechanism by which an individual sequence folds to adopt its 3D structure.

Some approaches to classify protein structure and understand the changes it undergoes throughout its trajectory are based on the expectation that sequence homology of a sufficiently high degree leads to structural similarity [4]. Other exercises, such as molecular dynamics (MD) simulations, are based on statistical physics and are used to evaluate the structural changes related to function [5]. The main weakness of these homology-based strategies is that they do not scale well as the number of proteins increases. Structural alignment is an instance of the traditional 3D graph matching problem, which is known to be NP-hard [6] (i.e., there is no known algorithm that can solve these problems in polynomial time $O(n^c)$, where c is a constant and n is the size of the input, which in this case is the number of proteins and their size).

We are interested in the high-throughput analyses of proteins, in uncovering their functions, and noting changes in their conformational states. In an effort to develop scalable methods for this purpose, we investigate machine learning (ML) approaches, with a particular focus on deep convolutional architectures [7], [8], [9], which are becoming the de-facto inference techniques in a variety of fields, solving previously open problems such as object recognition [10].

Our encoding in [11] formats the structural and conformational information of macromolecules into a fixed-size, color image. We avoid the complexity of 3D protein matching by turning the analysis into a more computationally tractable image-based pattern recognition problem. We aim to avoid certain constraints that arise in homology-based studies, and focus on a methodology that enables analysis of arbitrarily large protein databases or MD trajectories in an efficient, high-throughput manner. In this work we extend our previous encoding representation with additional visualization capabilities and the ability to better represent molecular conformational changes over time. Specifically, our contributions are:

1) A general representation of macromolecules that explicitly encodes secondary structural motifs and their spatial characteristics within the molecule. This representation exposes intra- and intermolecular structural patterns without having to perform protein alignments.
2) A split-input, residual, convolutional neural network architecture that is specifically geared towards manipulating and gleaning information from the above encoding setup.
3) An image classifier using our network and encod-

- *T. Estrada 0000-0001-7743-8754, H. Carrillo-Cabada 0000-0002-5823-9909, J. Benson 0000-0002-5686-2211, and B. Mulligan 0000-0002-6582-1905 are with the Department of Computer Science, University of New Mexico, Albuquerque, NM, 87131.*
  Corresponding author T. Estrada, E-mail: trilce@unm.edu
- *A. M. Razavi 0000-0003-4639-5856, Michel A. Cuendet 0000-0002-0754-3425, and Harel Weinstein 0000-0003-3473-9818 are with Weill Cornell Medical College, Cornell University, New York, NY, 10065.*
- *M. Taufer 0000-0002-0031-6377 is with University of Tennessee, Knoxville, TN, 37996.*

Manuscript received November 30, 2018; revised April 17, 2019.

ing, to predict eight different classes of protein functions, reaching a balanced accuracy of 80%.

4) A means of visualizing and quantifying changes in conformational states during molecular dynamics simulations.
5) Access to the dataset of our encoded representations and our neural network graph[1].

Our proposed encoding opens the door for structural biologists to use image processing and machine learning techniques to analyze very large macromolecular databases in an efficient, high-throughput way. Large scale analyses of this magnitude can be used to identify inter-molecular patterns that may signal function, interaction, and homology.

The remainder of the paper is organized as follows: Section 2 discusses other approaches related to protein representations and machine learning in structural biology. Section 3 introduces our macromolecular representation and the neural architecture we use for its analysis. Section 4 discusses two applications: a system for protein function prediction that serves to highlight the power of our encoding, and an approach to visualizing and detecting conformational changes in MD simulations. Section 5 summarizes our work, reflects on our lessons learned, and presents ongoing and future research directions.

2 Background and State of the Practice

A variety of work has been done on general protein analyses [1], [2], [3], [4], [5], [11], [12]. Two major elements of this type of work are *representations*, or how the protein is expressed, and *methodology of prediction*, or how the represented proteins are used to perform inference. In this section, we cover common representations of proteins. We then discuss the use of machine learning in structural biology, and conclude with the impact of data representations and predictive approaches for high-throughput analyses.

2.1 Representations

Proteins can be represented in a variety of ways, each with their own pros and cons with respect to preserving or exposing information for specific purposes. We present a summary of some standard representations and focus on their applicability to protein function prediction.

2.1.1 Sequence Representation

DNA, RNA, or proteins can be represented by their nucleotide sequences: a succession of letters using GACT for DNA, GACU for RNA, and the one-letter codes for the 20 natural amino acids for proteins. A technique to identify functional or structural relationships among proteins depends on aligning their sequences to find global or local shared motifs. Aligned sequences are usually represented through matrices, where each sequence corresponds to a row. Alignments can include gaps between columns to allow for local dissimilarities. Pairwise sequence alignment can be performed using dynamic programming (e.g., Smith–Waterman algorithm [13], Needleman–Wunsch algorithm [14], [15] both with a time complexity of $O(nm)$ [16], where n and m are lengths for a pair sequence alignment). It is inexpensive to align millions of proteins using modern parallel methods [17]. Using sequence alignment for protein function prediction is based on the idea that proteins with similar sequences (homologous) share similar functions. However, this is not always true, and it has been argued [18] that sequence alone is not enough for predicting protein functions and requires knowledge on the folding patterns of the protein's 3D structure. For example, despite the lack of sequence homology between classes, all GPCRs have a common structure and mechanism of signal transduction [19].

2.1.2 Structural Representations

Methods for protein structure determination include X-ray crystallography, NMR spectroscopy, and electron microscopy [20]. Structural representations involve expressing, in a variety of ways, the 3D arrangement of atoms in a protein. A 3D representation consists of the spatial coordinates of each of the (non-hydrogen) atoms in a Cartesian coordinate system (see Figure 1.a). An angular representation expresses the proteins backbone conformation through its dihedral angles (i.e., angles between planes of two sets of three atoms). With this representation, folds of proteins are expressed through dihedral angles formed by four consecutive alpha Carbon atoms. Due to the degrees of freedom of both of these protein representations, their space complexity grows exponentially with the number of residues.

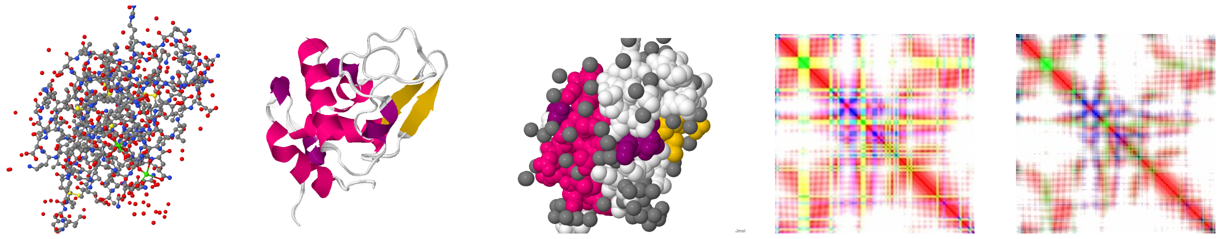
The multi-fold representation (see Figure 1.b) is based on the observation that a proteins structure can be expressed through the combination of small structural units, called folding motifs, [21], [22], [23]. This representation takes advantage of collections of motifs that occur frequently and uses them as a meta-dictionary to express the entire protein complexity in a condensed way. The most common representation of this kind uses folding motifs known as secondary structure motifs (e.g., helix, turn, and sheet).

Structural comparison and alignment of proteins is a critical aspect of multiple research problems, including protein annotation, and protein structure prediction. Structure-based function prediction often outperforms sequence-based methods because structural homologous contain similar folding patterns, even after evolution leads to their sequence similarity being undetectable [18]. Structural alignment combines sequence information with the secondary and tertiary structure of the protein or RNA molecule and is considered as the standard practice for homology-based structure and function prediction [18]. But thoroughly comparing protein structures, whose size range from tens to several thousand amino acids, is computationally expensive, as 3D matching is an NP-hard problem [6]. Moreover, for high-throughput analysis and identification of homologous structures, the alignment and comparison has to be done for multiple macromolecules at a time, limiting opportunities for parallelism.

2.1.3 Other Protein Representations

Other representations are being used to describe proteins, their components, or binding pockets for example [24]. One such representation expresses only the molecular surface [25] as a set of functions (e.g., triangulation, polygons, distance distributions and landmark theory) on a

[1] https://lobogit.unm.edu/datascience/graphicencoding_tcbb19



(a) Cartesian (b) Multi-fold (c) Surface (d) Encoding in [11] (e) This encoding

Fig. 1: Visual depiction of multiple representations for the human alpha-lactalbumin protein (PDBid: 1A4V). Figures (d) and (e) contrast our previous encoding in [11] and our improved version in this work

unit sphere. This particular representation makes multiple protein comparison relatively easy [26], [27], [28], but does not account for the internal structure of the protein, which is crucial for determining function (see Figure 1.c for an abstraction of this representation).

Another representation treats the residues in a protein as if they were vectors in a 20-dimensional space [29]. In this case, a protein is represented as a random walk and proteins can be compared to each other through their vectorized profile. Ultimately, this representation loses global folds which are helpful in characterizing protein functions. The multipolar representation [30] offers a hierarchical, parametric approach to characterizing the shape of a molecule. This representation uses multipoles (i.e., mathematical series that describe functions in terms of spherical harmonics) associated with coordinates of the alpha Carbon of each residue as shape descriptors. The multipolar model reduces a protein to a vectorized format; calculating distances between proteins can be done through vector operations rather than detailed alignment and spatial superimposition.

Our work differs from all of the described related work in that we propose an encoding mechanism that captures secondary and tertiary information of proteins into an easy-to-analyze format. Figure 1 shows a visual comparison between three structural protein representations described above (i.e., 3D Cartesian atoms, multi-fold, and surface) against our proposed graphic encoding. Our contribution is in the general and homogeneous data representation of molecules and their subsequent analysis. The specific changes made in this paper with respect to our encoding in [11], as described in Section 3.1.3 allow us to use it not only for classification, but also to perform visualization and complex local and global analyses on temporal data, such as molecular dynamics simulations. We present two applications of our encoding that were not possible with the previous version: protein function prediction 4.1 and detection of conformational changes in molecular dynamics (MD) simulations 4.2. Our ongoing work uses this encoding to extend in-situ analysis [31] and indexing of molecular dynamics trajectories [32].

2.2 Machine Learning in Structural Biology

Machine learning (and more recently, *deep learning*) has been used extensively in structural biology [33], [34], [35]. One of the main uses is in the prediction of secondary and tertiary structure of macromolecules. Li et al. [36] use a convolutional neural network to extract multi-scale features and predict secondary structure from protein sequences. Wang et al. [37] use two deep residual neural networks to perform contact prediction from protein sequences to improve folding accuracy. Recently, a team from Google (AlphaFold) used deep neural networks and images to reconstruct protein structures. Their work won the CASP18 competition [38]. Similarly to our work, they use distance matrices and angles to to analyse proteins. But in their case these two sources of information are decoupled, while our work puts the two together into a single input.

Hou et al. [39] use a deep convolution neural network (DeepSF) to classify a protein sequence into known folds. Nguyen et al. [40] propose an ensemble of classifiers like nearest neighbors, deep convolutional neural networks, and residual neural networks to predict a variety of angular and structural information to predict loops. Li et al. [41] compare the effectiveness of a deep neural network, a deep restricted Boltzmann machine, a deep recurrent neural network, and a deep recurrent restricted Boltzmann machine to predict phi and psi torsion angles of protein backbones. Poplin et. al [42] introduce a deep convolutional neural network (DeepVariant) used to determine the sequence of an individual's genome by learning likelihoods between images of read pileups around putative variant sites and ground-truth genotype calls.

More closely related to our work, deep learning has also been used for a variety of protein function prediction problems. Kulmanov et al. [43] propose DeepGO, a deep learning architecture used to learn features from protein sequences to predict function in the form of the Gene Ontology hierarchy. Similarly Liu et al. [44] use a recurrent neural network to predict four types of functions from protein sequences. In both cases, the neural network architecture is employed to form low-level feature representations from a simple input format as is the protein sequence. Cao et al. [45] propose ProLanGO, a deep recurrent neural network that deals with protein function prediction as if it was an analogous problem to language translation. This approach maps the protein sequence to a sequence of functions defined in the Gene Ontology.

2.3 High-throughput analysis of molecular simulations

Work has been done to understand structural properties of proteins as they fold. The most common approaches are based on structural homology, and are usually performed by computing sub-graph similarity between sets of structural patterns and a target protein [46]. This process is known

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2019.2945291, IEEE/ACM Transactions on Computational Biology and Bioinformatics

to be NP-hard [6] (i.e., there is no known algorithm that can solve these problems in polynomial time $O(n^c)$, where c is a constant and n is the size of the input, which in this case is the number of proteins and their size). Optimizations such as efficient filtration of spaced k-mer neighbors [47], extraction of family specific packing motifs [48], Laplacian characterization of tertiary structures [49], and use of multidimensional scaling to index conformational space [50], [51] have been proposed to speed up the search and comparison of structures for homology analysis. However, a major limitation remains, and it is the need to access the large homology database, which can be in the order of 50 to 80 GB and will continue to grow.

Both types of analyses start by collecting a large number of protein structures. For this comparison, we can divide the analysis into offline stage and online stage. The offline stage is independent of the specific protein(s) to be analyzed, it is performed ahead of time and can be computationally expensive, but it is done only once (for a given model). The online stage is the inference step, where a protein is analysed. For the online stage, if performed in-situ, resources are limited and speed of computation, memory usage, and I/O access need to be optimized.

2.3.1 Homology-based Approach

In this approach, a database or data organization is built offline. When a new structure needs to be analyzed online, it is compared with the structures in the database. Even by using searching and comparison optimizations, accessing the database can become a problem; compute nodes would have to keep the database in memory or retrieve the data from secondary storage. As I/O has become the bottleneck in high performance computing, this problem renders the homology-based approach prohibitive. As the number of proteins increases over time (e.g., with the advancing of crystallography and NMR techniques), more scalable analysis techniques are needed to fully take advantage of high performance computing resources.

2.3.2 Model-based Approach

In this case, the offline stage consists of building a model (for example, training a neural network) that can be used at a later stage to perform predictions without requiring expensive memory accesses. The major drawback of model-based methods is that the model needs to be recomputed every time new data is added. Depending on the time and resources it takes to compute the model, this may or may not be a limitation.

The way in which proteins are represented affects the ways in which we make inference or draw predictions from them. Classic homology-based approaches leverage sequential and structural representations and require querying large data collections at run time. Whereas model-based approaches can take advantage of inference methods. In this work we provide the encoding mechanism and a convolutional neural network architecture that enable a particular type of model-based protein analysis approach. In practice, training our neural network (see 3.2) takes less than one hour in a commodity GPU. Once the network is trained, it can be used as an online prediction model. This network can be kept in RAM, and when a new structure has to be analyzed, it is enough to encode the structure as an image and pass it through the network. The whole process takes less than 2 seconds in the modest Intel Xeon E5-1620 v4.

3 Methods

In this section, we present our graphic encoding of secondary and tertiary structure of proteins and note five key advantages over other structural representations:

- It is invariant to the protein size (i.e., number of residues). Proteins vary in size through thousands of residues but our graphic encoding can represent them all in a standard way.
- It is invariant to the protein orientation and does not require any sort of alignment.
- It exposes structural domains and folding motifs as patterns in an image.
- It enables efficient model-based approaches for querying structures in a high-throughput fashion.
- It provides a visual interpretation of proteins and can be used to visually and efficiently inspect large collections of data efficiently.

We also introduce our neural network architecture, the Graphic Encoding of Macromolecules Network, which is used in subsequent sections for particular applications in classification proteins and detection conformational changes in simulations.

3.1 Structural Encoding

Our encoding mechanism translates the complex structural and conformational information in 3D proteins into a much simpler-to-analyze format: a 3D *NxNx3* tensor that can be visualized as an NxN image. The three NxN matrices encode proteins' secondary and tertiary structure in three channels in a Red-Green-Blue (RGB) color model. The advantages of this representation is that it enables the use of state-of-the-art pattern recognition techniques in machine learning to automatically find structural motifs in data collections or to detect conformational changes in folding trajectories. While these color images are aesthetically pleasing, they map back to the original 3D protein in a tractable way.

The encoding process consists of four steps, also depicted in Figure 2 and explained in detail in the following subsections:

1) Extracting secondary structural information using the Ramachandran plot (see 3.1.1).
2) Expressing tertiary structural information via the distance matrix (see 3.1.2).
3) Encoding secondary and tertiary information into multiple codified channels (see 3.1.3).
4) Formatting the image (or tensor) into a fixed-size final encoding (see 3.1.4).

3.1.1 The Ramachandran Plot for Secondary Structures

The first step for our encoding is to identify the basic molecular structures forming the protein. One way of doing this is through the analysis of backbone dihedral angles of the amino acid residues in the macromolecular structure. The Ramachandran [52] plot determines the energetically

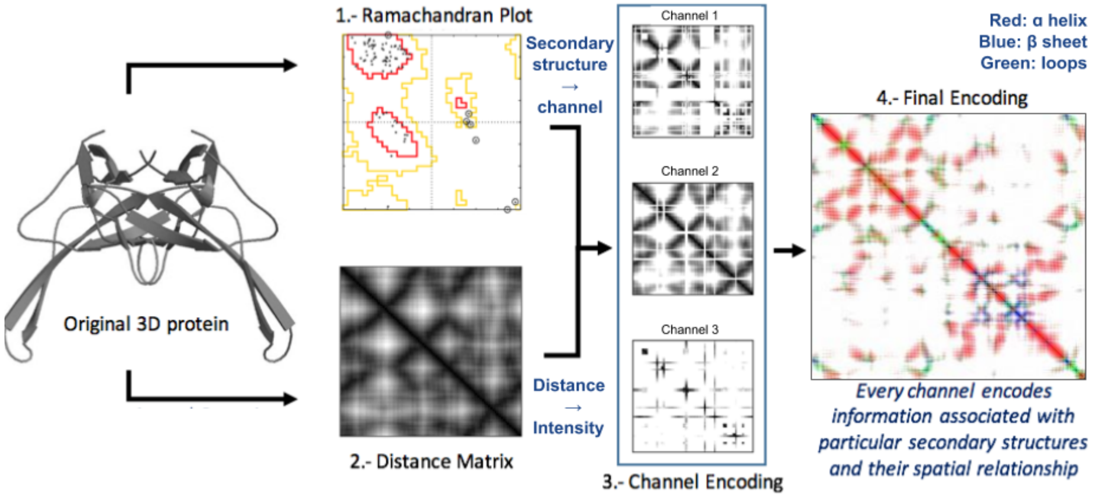


Fig. 2: Steps of encoding procedure. Pictured, gene V protein (PDBid: 1AE2).

allowable regions for the torsion angle ϕ, (angle between the C-N-CA-C atoms) versus the torsion angle ψ, (angle between the N-CA-C-N atoms), and ω (usually restricted to be $180\deg$ for the typical trans case or $0\deg$ for the rare cis case), for each residue of a protein sequence. Based on the constraints of the torsion angles (ϕ, ψ, and ω) as described by the Ramachandran plot, we can associate each amino acid residue in the protein with one of six types of secondary structures: α-helix, β-strand, Polyproline PII-helix, γ'-turn, γ-turn, and cis-peptide bonds. For each residue in the protein, we compute its torsion angles and determine its corresponding secondary structure given. For our interpretability purposes, we create 3 groupings of secondary structures: α-helix, β-strand, and *other*, which is explained further in 3.1.3.

3.1.2 Expressing Tertiary Structure through Distances

The second step seeks to establish a spatial correlation between the different residues in the protein. In this step, we use the protein's distance matrix [53], which has been used as an aid to perform enzyme structural analysis and modeling [54]. We first identify the alpha carbon (i.e., the first carbon atom of an aliphatic chain that is attached to a functional group. For amino acids, this is the carbon atom next to the carboxyl group) in each residue. Then, for a protein with M alpha carbon atoms ($C\alpha$), its distance matrix is a squared matrix D of size $M \times M$, where the element in $D(i,j)$ corresponds to the euclidean distance (originally calculated in Angstroms $\mathring{A}$) between atoms $C\alpha_i$ and $C\alpha_j$. In turn, this is a symmetric matrix. Note that the matrix is not restricted to a particular distance metric and we could use any metric or correlation coefficient for this purpose (e.g., Euclidean, squared Euclidean, Minkowsky, Chevychev, cosine, spearman, or hamming). However, to be able to capture conformational changes as the protein folds, we require our encoding to be robust to rotational changes in the protein and thus, we opt to use the Euclidean distance. To encode the raw secondary and tertiary structure of the protein, we compute the distance between every pair of alpha carbon atoms ($C\alpha$) in the backbone of each residue and use it as the skeleton of our graphic encoding. An example of a distance matrix is shown in Figure 2.2.

3.1.3 Encoding Structures in an Image

The third step combines the extracted secondary structures and distance matrix to represent the protein into a tensor. For practical purposes, and to take advantage of image processing models, we decided to use a tensor of dimensions $M \times M \times 3$, where M is again the number of amino acid residues in the protein, and 3 indicates the Red-Green-Blue channels in an image. We use color to encode secondary structure and use intensity, or color saturation, to proportionally represent distances. Recall in 3.1.1, amino acid residues were classified according to their dihedral angles into three secondary structures: α-helix, β-strand, and *other*. Then, we can use the RGB model to differentiate each structure as follows: α-helix, red; β-strand, blue; other, green. If the residue cannot be associated to any of the secondary structures defined as energetically allowed in the Ramachandran plot, we color that residue gray. Interaction between the three secondary structures take the remainder color shades (e.g., interactions between α-helices and β-strands are colored magenta, interactions between α-helices and other structures are colored yellow).

To encode a protein into its image representation, we define a function $dist(i,j)$ over the distance matrix D, where $dist(i,j)$ is a returns a scaled distance value between 0 and 1, proportional to the euclidean distance between the 3D coordinates of $C\alpha_i$ and $C\alpha_j$. This function scales a distance of 10$\mathring{A}$ to 1, and everything else proportionally (e.g., $dist(i,j) \longleftarrow \frac{D(i,j)}{10}$). A scaling factor of 10 was chosen because typical cutoffs for electrostatic calculations of atomic interactions in molecular dynamics simulations range from 9$\mathring{A}$ to 15$\mathring{A}$ [55]. We determine the saturation of each color channel using the colors assigned to the particular residue of that channel and its interactions with all the other residues in the protein. Helices are red, strands are blue, and all others are green. Unidentified regions are treated as gray and interaction between different structures produce a mix of colors. As an example, a red α-helix in position i, the saturation for channels red, green, and

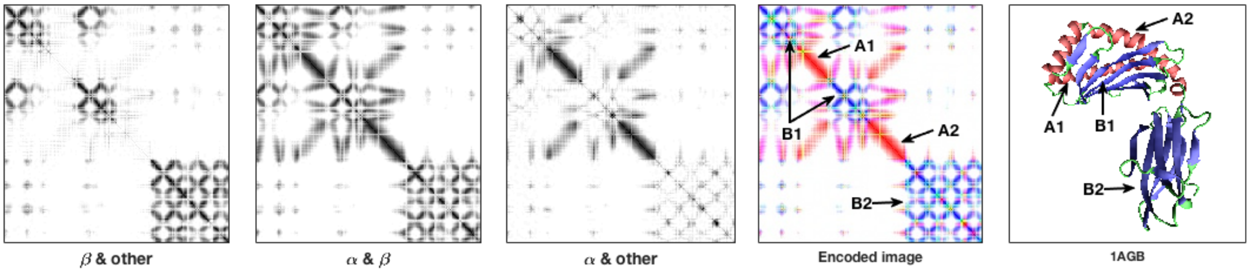
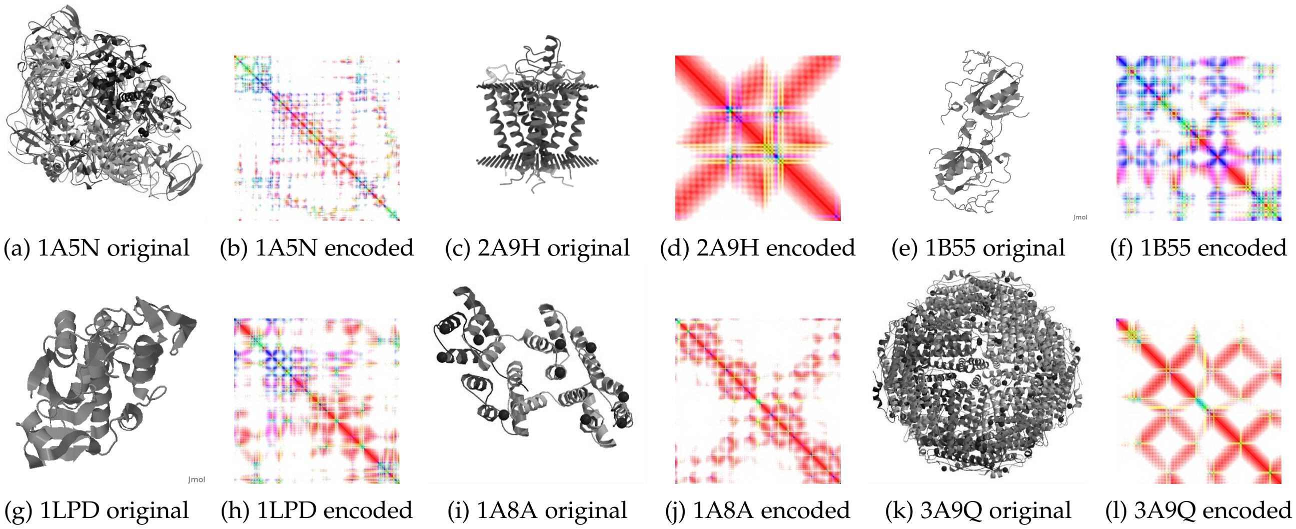


Fig. 3: Example of encoding in three channels of Antagonist HIV-1 GAG peptide (PDBid: 1AGB).


(a) 1A5N original (b) 1A5N encoded (c) 2A9H original (d) 2A9H encoded (e) 1B55 original (f) 1B55 encoded

(g) 1LPD original (h) 1LPD encoded (i) 1A8A original (j) 1A8A encoded (k) 3A9Q original (l) 3A9Q encoded

Fig. 4: Examples of encodings for a diverse set of proteins.

blue is $[1, dist(i,j), dist(i,j)]$ and $[1, dist(j,i), dist(j,i)]$ $\forall j \in D$. In the same way, the saturation for other structures in blue, green, and gray are $[dist(i,j), dist(i,j), 1]$, $[dist(i,j), 1, dist(i,j)]$, $[dist(i,j), dist(i,j), dist(i,j)]$, respectively. The color of two interacting residues i, j is given by the element-wise maximum between the two colors. Note that if residue i is of the same type of j, then their contributions to pixels i, j and j, i are the same. If they are of different types, they contribute to different channels and lead to a larger span of colors. Compared to our encoding in [11], we capture only 3 different secondary structures rather than the six we did before. The rationale for this change, that effectively reduces the information contained in an image, was necessary to enable coherent visualization. Previously, interaction between two different types of secondary structures was not immediately apparent, as one secondary structure color would override the other. In this new version, the colors get combined to showcase these interactions. This seemingly simple change, allows us to extend the work to an entirely new scope, such as the MD analysis in Section 4.2.

To provide a more concrete example of the information contained in each channel and the resulting image, we walk through Figure 3. The first three blocks (from left to right) correspond to the three different channels in the RGB image. The fourth block corresponds to the resulting encoded image. The right-most block is the encoded protein. The protein of reference in this example is the agonist HIV-1 GAG peptide, which (broadly) consists of two helical domains (which we denote *A1* and *A2*), a β-sheet (denoted *B1*) joining the two helices, and a collection of β-strands interacting with each other (denoted *B2*). Recall that the diagonal encodes the secondary structure for a particular residue and elements off the diagonal encode interactions between residues. Pixel intensity corresponds to distance between residues and color shades correspond to different secondary structures. Given this information, and recalling that red encodes α-helices, blue encodes β-strands and green encodes everything else, we can visually find different structures in the encoded image. Note how the B1 β-sheet is partitioned in four squares. This indicates that one of the helices is coming out from the middle of the sheet. Also, this *4-square pattern* indicates that all of these β strands are very close to each other, as opposed as being part of two different β-sheets, which would generate a *2-square pattern* along the diagonal. Similarly, the distinctive cross formed by the red regions indicate that these two α-helices are in close proximity. Now, for the specific information encoded in the three different channels, a darker shade of gray corresponds to distance between particular secondary structures. The first block shows proximity regions between β-strands and *other* structures, the second is α-helices and β-strands, and the third one is α-helices and *other*, which all combined result in the colored image in the fourth block. Figure 4 shows examples of some very different macromolecules in a multi fold representation and our graphic encoding. Note that helices tend (although not always) to produce narrower representations (almost like bands along the diagonal) and their interactions produce X's. This is because residues arrange in elongated structures rather than in packed spaces, which is the case for β-sheets that generally produce blocks. Loops and coils can be seen mostly as elongated crosses through the image. Different domains can be seen as diamond-like shapes through the diagonal. By looking at these images it is easy to distinguish how our encoding exposes motifs at different granularities in the image.

3.1.4 Formatting and Resizing

The final step consists of performing an image resizing (by applying a bi-cubic interpolation) to produce an output of consistent dimensions across proteins regardless of their original length. Assuming a new size N the output is a $N \times N \times 3$ tensor, where N can be smaller or larger than the original M, and 3 is again the number of channels used in the RGB encoding. The output image either encodes more than one residue per pixel, or uses multiple pixels to encode one residue. The size of N can be chosen differently to optimize different performance metrics. For example, N can be equal to the number of residues in the longest protein in a dataset to optimize fidelity of the encoding; it can be the average number of residues in the dataset to keep a trade off between fidelity and efficiency; or it can be set to an smaller size to enhance processing speeds. For our experiments we use $N = 227$, which allows us to experiment with a variety of neural network architectures [7], [9], [12].

Our encoding process for proteins, depicted in Figure 2, produces symmetric images that visually highlight the secondary and tertiary structure of a protein. Small differences between similar structures can be noticeable by a sharp change in color. For example, when a helix unfolds, this maps to a turn from red to yellow/green. Note that, in this particular encoding approach, we are building images; however, the number of channels that could be used is not restricted to three. In addition to structure and distances, other information like charge, or physical properties like hydrophobicity, could be encoded into supplementary channels. One of our goals is to visualize the proteins, so three channels give us the best trade off between information and interpretation.

3.2 Graphic Encoding of Macromolecules Network

As our method provides a structural representation of proteins that is different from other formats, its analysis mechanisms are also different. Identifying structural motifs across a large database or performing protein modeling for function prediction does not require alignment and/or superimposition; thus, breaking a performance barrier for high-throughput analysis. Our representation transforms traditional structural biology problems into an image pattern recognition problem, and it enables a straightforward use of image processing and machine learning techniques for analysis and prediction.

Then, as a supplement for our encoding, we present GEM-net, a convolutional neural network architecture that we use in different ways to perform protein function prediction 4.1 and detect conformational changes in protein folding trajectories 4.2. A convolutional neural network, also known as a *CNN*, is a mathematical construction that trains complex non-linear functions out of linear compositions. CNNs handle matrix-oriented input and can produce a classification output. When applied to images, convolutions are used to preserve spatial relationships between pixels and learn visual patterns. By representing secondary and tertiary structural information of proteins as *NxNx3* tensors, we are able to lean on these image-based classification techniques.

TABLE 1: Our neural network architecture

GEM-net		
input (227 x 227 x 3)		
red channel (227 x 227 x 1)	green channel (227 x 227 x 1)	blue channel (227 x 227 x 1)
conv3-16-1 (227 x 227 x 16)	conv3-16-1 (227 x 227 x 16)	conv3-16-1 (227 x 227 x 16)
Add (227 x 227 x 16)	Add (227 x 227 x 16)	Add (227 x 227 x 16)
merge (227 x 227 x 48)		
conv3-64-2 (114 x 114 x 64)		
Batch Normalization		
conv3-32-2 (57 x 57 x 32)		
Batch Normalization		
conv3-32-2 (29 x 29 x 32)		
Batch Normalization		
conv3-16-2 (15 x 15 x 16)		
Batch Normalization		
FC-64 (1 x 64)		
Dropout 25%		
FC-16 (1 x 16)		
softmax (1 x 8)		

Noting that our encoding method relies on the channel separation of secondary structures, we opted to develop an architecture that was specific for our task. Common neural architectures take a 3-color channel image as input and apply convolutions and other operations directly. This immediate convolution means that the input channels are handled together, such that filter kernels attempt to capture the relationships between different color channels. However, in our encoding method, we particularly aim to maintain different secondary structures in the different color channels. It follows, then, that we treat each channel independently, and aim at learning filters that are relevant for each type of secondary structure.

Our Graphic Encoding of Macromolecules Network, or *GEM-net*, is a split-input residual network architecture designed to extract the most information from each channel, independently. We employ skip-connections [12] to propagate information from the input of the convolutional layer (i.e., the red, green, or blue channels) to the output of the convolution using the "Add" function. This is equivalent to adding the values of the input back into each output channel. Table 1 describes the general architecture of GEM-net, in which we use a setup that first treats each color channel independently through the residual blocks and then sends the combined tensor onward through four convolutional layers with padded input. A classification layer is followed after two dense layers. Batch normalization between layers serves to denoise the intermediate output tensors and lead to stable convergence. Dropout is also used to help reduce overfitting. Tensor sizes are provided in parenthesis and convolution parameters are denoted by the name of the layer as *conv<receptive field>-<channels>-<stride>*.

4 Applications

In this section, we cover two major applications that are made possible through use of our graphical encoding and our neural network architecture. First, we use it for protein function prediction, and more specifically, to determine if the changes we did to our encoding from the previous version in [11] still hold a predictive value. Thus, we train, test, and validate our encoding and neural network on over 70K encoded images of proteins and we compare against a set of publicly available networks in the task of protein function prediction. The second application, for detecting and visualizing conformational changes in protein trajectories, is only possible because of the slight modifications we made to the encoding; by allowing us to visualize and quantify in s straight manner the interaction between the different parts of the protein. In this second case, after demonstrating a robust model for function prediction, we transfer our network's knowledge to perform a completely different task without retraining, that is detecting conformational changes in protein trajectories. We discuss particular use cases and the insights that can be derived from our methodology for two molecular dynamics simulations.

4.1 Protein Function Prediction

Proteins contain a wide variety of structural motifs, which can also constitute functional microdomains that support the protein's functions. In this section we test the ability of our graphic encoding to expose structural information necessary to perform basic protein function prediction.

4.1.1 Dataset Description

Our dataset consists of 73,337 proteins from the Protein Data Bank [56][2]. The protein data bank format (PDB) provides a standard representation for macromolecular structural data derived from X-ray diffraction and NMR studies. A PDB file encodes a protein as a sequence of atoms, their type, and their 3D coordinates. This representation can be easily converted to our encoding as explained in Section 3. Proteins in the dataset range in size from less than 100 non-hydrogen atoms to more than 50,000. The mean size is 6508 atoms with a standard deviation of 19495. The mean resolution is 2.2 Angstroms, with a 1.7 standard deviation. The main source organism in this dataset is the *Homo Sapiens*, but the collection also includes *Escherichia coli*, *Mus musculus*, *Saccharomyces cerevisiae*, *Rattus norvegicus*, and *Mycobacterium tuberculosis* among others. Figure 4 depicts multiple examples of proteins in our dataset that were transformed from a 3D structure to our graphic encoding.

To perform function prediction in this dataset, we obtain *GO terms* through the RCSB Protein Data Bank [20] and their biological details report. *GO terms* are established by the Gene Ontology Consortium [57], [58], [59] (GOC). GOC provides a standardized and consistent way of describing and documenting gene products across databases. The GO project comprises three structured ontologies with a well defined vocabulary to express gene product properties over three domains: cellular component, molecular function, and biological process in a species-independent manner. Terms in the cellular component describe the parts of a cell or its extracellular environment, for example a ribosome. Terms in the molecular function describe activities that are performed by individual gene products or assembled complexes. Examples of such activities include binding or catalysis. Finally, terms identifying biological processes encompass series of events carried out by molecular function with a well defined beginning and end. To label our dataset with specific functions, we use a biological process taxonomy provided by RCSB-PDB [20]. From this taxonomy we selected eight biological processes with the largest number of proteins (i.e., more than 5,000) and use these groups as our classification targets. Table 2 describes this classification.

TABLE 2: Dataset breakdown by their biological processes

Label	Function	GO-term	Number
0	Biological regulation	GO:0065007	5,872
1	Immune system process	GO:0002376	7,106
2	Signaling	GO:0023052	8,829
3	Multi-organism process	GO:0051704	8,309
4	Catabolic process	GO:0009056	9,889
5	Localization	GO:0051179	6,732
6	Oxidation-reduction process	GO:0055114	12,344
7	Biosynthetic process	GO:0009058	14,248

* *GO source http://amigo.geneontology.org*

4.1.2 Evaluation of Function Prediction

To test our approach we compare two general purpose pre-trained deep neural networks: Google's Inception-v3 [7] and MobileNet [8]; one other established image classification architecture trained from the ground up: VGG-net [9]; and our split input residual architecture designed specifically to take advantage of individual channel encoding. For all of our tests we perform 5-fold cross validation, which splits the dataset into 5 disjoint partitions, each worth about 20% of the data. Then, training is done with 4 out of 5 partitions (i.e., 80%) and testing is done with the unseen partition. The process is repeated 5 times, using a different set of partitions each time for training and testing. Through this process, every protein in the dataset is used for training four times and for testing once. We use a learning rate of 0.005, a batch size of 100, and cross-entropy as our loss-function.

[2] PDB Dataset download date — *August 30, 2017*

The number of epochs we used varied per architecture and is indicated in Table 2.

The pre-trained networks needed longer training periods because they only change weights in the last layer and use features learned from general image classification in the other layers. The networks we trained from scratch converged quite quickly (within 10-20 epochs), further training steps only increased overfitting. The hardware used for building our models is an Intel Xeon 8 core E5-1620 v4 at 3.50GHz with 8 GPU Tesla P100. A summary of our results is presented in Table 3, with the main performance metric being balanced accuracy, as defined by Mosley [60] to avoid inflated performance estimates on imbalanced datasets for multiclassification problems. We also include confidence intervals at 95% and training times in minutes.

TABLE 3: Results

Encoding: proposed 3-channel representation

Architecture	Epochs	Accuracy ± CI	Training time
MobileNet	500	36.3% ± 0.349	32 min.
Inception-v3	500	46.8% ± 0.362	75 min.
VGG-net	15	24.2% ± 0.833	58 min.
GEM-net	15	**80.8% ± 0.285**	45 min.

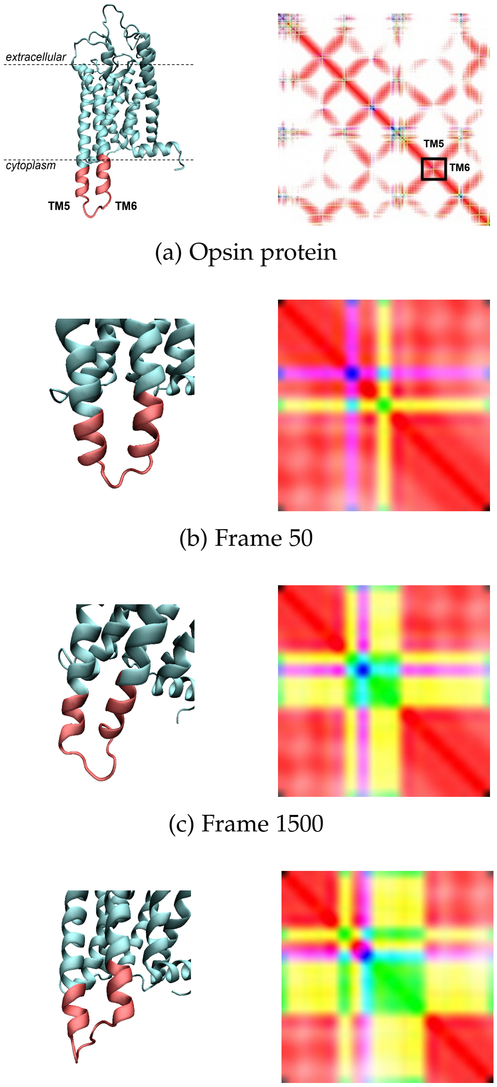
The class assignments are based on the protein's *GO term* classification, as explained above. Note that none of this information is provided to the classifiers. Like many convolutional architectures, the networks rely solely on the images to learn distinguishing characteristics from the groups and perform a final classification. Our results in Table 3 indicate that the general purpose networks are not able to reach high accuracy with this dataset. This result is expected because to build a deep network of these characteristics, with the hopes of it converging to a state that is practical for prediction, typically requires a very large number of labeled images (the original Inception network for ImageNet was trained on 1.2 million images, with 50,000 images for validation and 100,000 images for testing [10]). Note that, for this specific problem, GEM-net achieves the highest performance.

By using GEM-net and only 15 epochs, we are able to reach 80.8 balanced accuracy, and above 80% for five of the classes (Oxidation-reduction process, immune system process, catabolic process, signaling, and biosynthetic process) and below 75% for two of the classes (Localization and biological regulation). These results indicate that our encoding captures characteristics that differentiate functions among proteins and that GEM-net is able to find relevant discriminating patterns. With this in mind, our next goal (as presented in Section 4.2) is to determine if a trained GEM-net can be used to detect conformational changes in molecular dynamics simulations.

4.2 Detecting Conformational Changes in Trajectories

Protein molecular dynamics (MD) simulations study structural changes in the protein as a trajectory of conformations that evolve over time: as the protein folds, unfolds, or performs steps of its physiological function. During the process, it is important to identify the different structural changes that a protein undergoes over time (e.g., specific changes within a functional domain or correlated changes between multiple domains), as well as structural changes that occur in a similar way for different proteins. Our encoding method coupled with GEM-net enable a model-based structural analysis that is lightweight (i.e., prediction cost in GEM-net is O(C) or about 2 seconds in a commodity computer), can be easily performed on the fly, and requires minimal memory accesses compared to the homology-based approach (i.e., GEM-net takes only 2.4MB compared to the ≈ 50GB of a protein database). In the following sections we describe two ways in which GEM-net or the encoding itself can be used to detect conformational changes in protein folding trajectories.

4.2.1 Case Study: Opsin



(a) Opsin protein

(b) Frame 50

(c) Frame 1500

(d) Frame 1950

Fig. 5: Opsin (a) shows the entire protein and its encoding; (b,c,d) show residues 230-250 (left, red) and encoding (from the bounded box in (a))

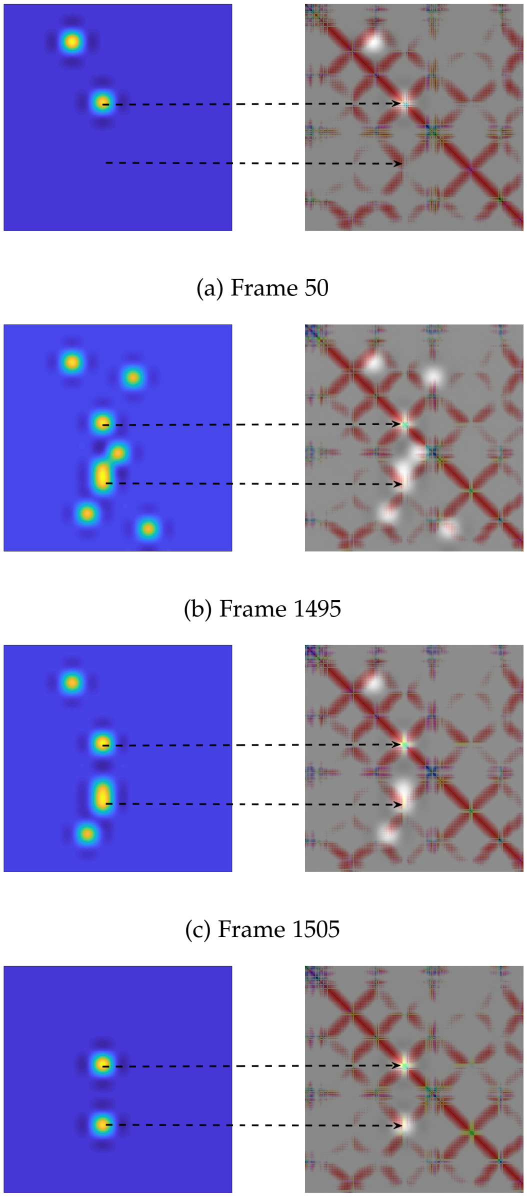
Opsins are a group of proteins containing seven transmembrane α-helical domains connected by three extracellular and three cytoplasmic loops. They belong to the G protein-coupled receptor (GPCR) superfamily. The structures of activated or agonist-bound GPCRs indicate how ligand binding at the extracellular side of a receptor leads to conformational changes in the cytoplasmic side of the receptor [19]. According to Rasmussen et al. [61] the change consists of an outward movement of the cytoplasmic part of the 5th and 6th transmembrane helices (TM5 and TM6 respectively).

Our graphic encoding makes the conformational changes between TM5 and TM6 easy to visualize and identify. In Figure 5.(a) we visualize the protein and its encoded image. The top corresponds to the extracellular side and the bottom to the cytoplasm. The different helices connected by loops can be seen in the encoded image as thick red diagonals connected by yellow/magenta lines. Labeled in the figure are the transmembrane helices TM5 and TM6 respectively. Figures 5.(b), (c), (d) zoom in the cytoplasm loop bounded by TM5 and TM6. In frame 0, the two helices connected to the loop are tightly formed (they are intense areas of red in the zoomed encoding, joined by small yellow and magenta bands), but as the simulation progresses, the loop performs an outward movement and the two helices unravel to some degree [62]. Using our encoding we can easily identify this process as the yellow color begins to dominate the encoded image.

Beyond just visualization, we can also use GEM-net to perform analysis of trajectories. We trained GEM-net with the ≈ 70K proteins to determine function prediction as discussed in Section 4.1. But for this analysis we leverage the learned features from that step to analyze MD simulations. This concept, of taking an existing neural network that has been trained on some dataset and re-purposing it for a new task is known as Transfer Learning. In fact, Google's Inception [7] and VGG [9] are general purpose image classification architectures that are successfully used for transfer learning.

To identify conformational changes in MD simulations we use the network body of GEM-net to analyze the way in which the network responds to different frames in a trajectory as the protein undergoes conformational changes. When a convolutional network is trained, its filters (also called kernels) converge to a variety of patterns that are used to form a composite (called feature maps) that signals the existence of complex patterns. When the network receives input, the feature maps highlight the variety of trained patterns within the input. Then, activation functions propagate patterns that the network deemed relevant. With a classifier attached to the end of the neural network, these filtered patterns are used in prediction. Using only the network body (without the classification layer), we can extract activations at each layer and visualize which locations in the image contributed to a particular activation. We then continue through the network body, passing through each layer up to the final convolution, (see Table 1). This is similar to Grad-CAM [63] that propagates input through layers of the neural network and highlights the activations.

Figure 6 shows class activation maps (left) for the Opsin protein at several simulation frames. We can use these activation maps to draw insight from our network architecture, to see which regions in the image (and the corresponding residues in the protein) are deemed relevant for classification. It is interesting to see how some regions of the protein become more or less relevant over time. For example, in Figure 6 we focus our attention on the two cytoplasm loops (denoted by the two dotted arrows). At first only the central loop is relevant, but as the simulation goes on, the second loop, or more specifically, the distance between the second loop and the first one (see region off the diagonal that corresponds to both of these loops) becomes relevant. These time frames, where the second loop in TM5 and TM6 becomes relevant correspond to times in the simulation where the outward movement is detected and TM5 and TM6 start to unravel (as depicted in Figure 5). It is important to note that additional research is needed to understand which of these interactions are truly relevant functional mechanisms and which are picked up by the network due to their high degrees of freedom. However, they provide us with concise

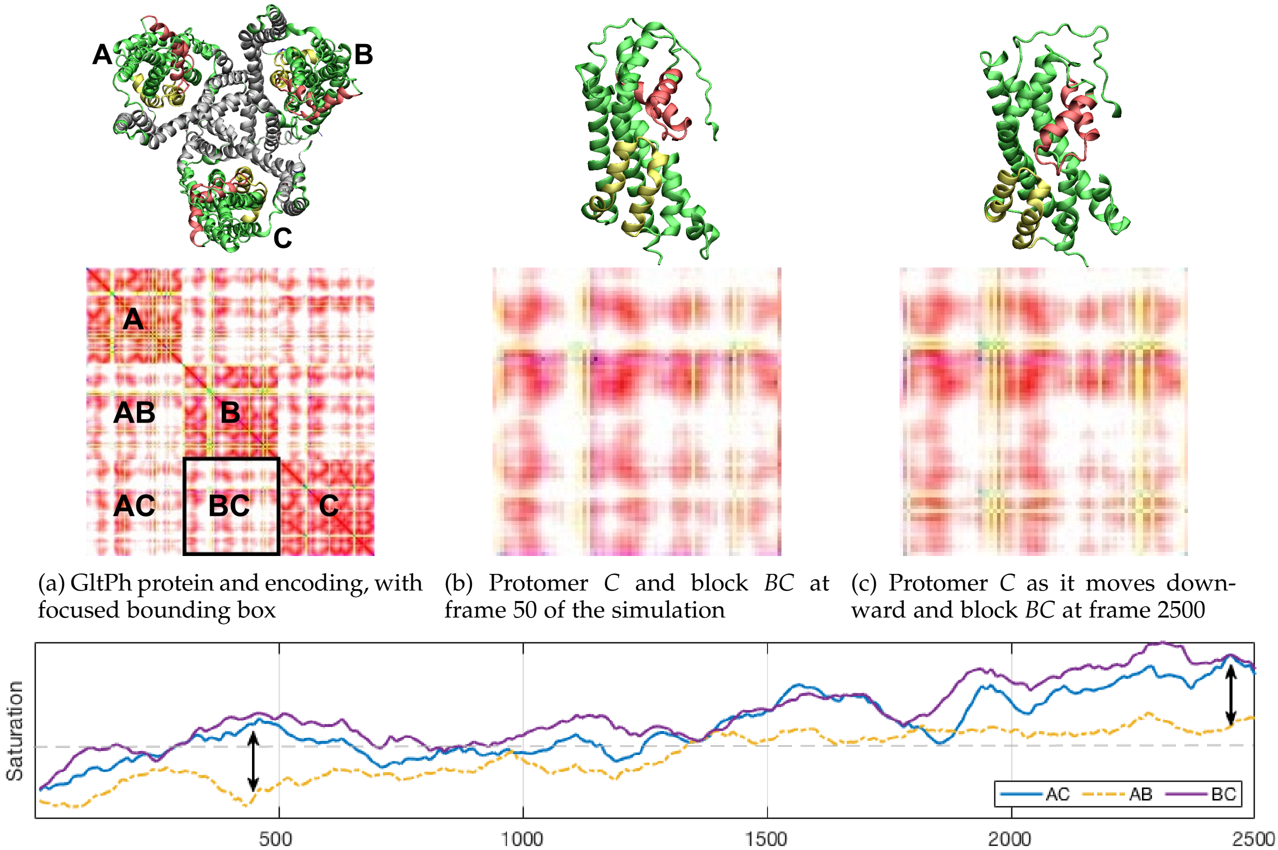

(a) Frame 50

(b) Frame 1495

(c) Frame 1505

(d) Frame 1950

Fig. 6: Activation maps for Opsin (left) and highlights with respect to the encoding (right).



(a) GltPh protein and encoding, with focused bounding box

(b) Protomer *C* and block *BC* at frame 50 of the simulation

(c) Protomer *C* as it moves downward and block *BC* at frame 2500

(d) Mean saturation in blocks AC, AB, and BC of GltPh's codified images as a function of time

Fig. 7: Gltph - a homotrimeric Asp/Glu transporter. The three transport domains are shown in green, and the scaffolding trimerization domains are in grey. Transport of substrate bound in the protein region shown in red, requires inward displacement of the substrate cage in an elevator-like motion.

information to further explore the dynamics of this protein.

4.2.2 Case Study: GltPh

GltPh is an aspartate transporter. It is a bowl-shaped homotrimer (i.e, a protein composed of three identical units of polypeptide), in which each protomer consists of two domains: a rigid trimerization domain formed by four transmembrane helices, and a peripheral transport domain formed by four transmembane helices and two re-entrant loops [64]. Protomers in GltPh exhibit a rigid body movement, sometimes called *elevator-like* motion that is considered a crucial part of the transport cycle [65].

In Figure 7.(a), we show the extracellular side in the outward-facing state of GltPh (top) and its encoded image (bottom). Its three protomers are easily identifiable in the encoded image as the three dark blocks in the diagonal (*A*, *B*, *C*). The blocks off the diagonal represent proximity among the protomers (e.g. block *BC* represents distance between protomer *B* and protomer *C*). Our graphic encoding is able to detect the *elevator-like* movement of protomer *C* with respect to *A* and *B*. This is expressed by increased saturation in blocks *AC* and *BC* as the cage moves downward.

If we quantify the saturation of the different blocks over time, we can automatically identify the simulation frames where a particular protomer performs the *elevator movement*. In Figure 7.(d) we present time series for the mean saturation intensity of the off-diagonal blocks in the GltPh encoded images as the simulation progresses. The solid lines represent saturation for blocks AC and BC in blue and purple respectively. The dotted line is saturation for block AB in yellow. Patterns that can provide us with insights are steady increase or decrease in saturation, correlated behavior, and pronounced differences. First, note that all the captured movement is relative. Then, off-diagonal blocks represent the movement of one protomer with respect to another (e.g., block BC represents relative movement of protomer B with respect to protomer C). Increased saturation in off-diagonal blocks indicate that the protomers are moving closer to each other, while decreased saturation indicates that they are moving apart. Correlated movement between blocks indicate that their common protomer is moving in a different direction with respect to the others. For example, in Figure 7.(d) we observe a correlated behavior between AC (in blue) and BC (in purple), which means that block C is moving in a different direction with respect to A and B. The difference becomes more evident around frame 500 and again towards the end of the simulation, starting at frame 2000. Specifically during the later stages of the simulation, it was manually verified that protomer C was performing an *elevator-like* movement, as predicted by the gap in saturations. With this, we provide a qualitative and quantitative way to estimate conformational changes at a coarse grain.

However, it is important to note that this method is not failsafe though, if two or more protomers move in synchrony, this analysis will confuse the movement source.

5 Conclusions and Future Work

Modern homology-based approaches to protein folding analyses can be computationally expensive or rely on heuristics that lose information about a protein's 3D shape. In order to leverage modern machine learning technologies and successfully predict a protein's function throughout its trajectory, in addition to uncovering changes in its conformational state, it is necessary to employ techniques that maintain the 3D information while performing at scale. We present a novel approach to encode proteins that potentially boosts the capabilities of scientists seeking high-throughput analysis techniques for their ever-increasing molecular datasets. We found that distance matrices coupled with angles of secondary structures provide a meaningful data representation for proteins. Our approach does not rely on homology calculations and we can create that encoding in isolation, in addition to performing predictions concurrently and avoiding costly computations from traditional homology-based approaches.

One of our future directions is to perform more focused function prediction (i.e., finer grained protein function definitions). We expect that by looking at a narrower scope and a better defined biological function, the classifier will be able to achieve even better accuracy. Another direction is experimenting with enriched representations, for example by including more channels (that is, instead of only RGB images, working with multidimensional tensors). This approach would sacrifice interpretability and visualization, but can include information such as energy and specify more than three secondary structures. Finally, to expand the potential impact of our work, we aim to provice it as an analytics tool for MD packages. We have released our model and datasets openly in hopes that other researchers may find it useful.

Acknowledgments

This material is supported by the NSF grants entitled *CAREER: Enabling Distributed and In-Situ Analysis for Multidimensional Structured Data* (NSF 1453430) and *BIGDATA: IA: Collaborative Research: In Situ Data Analytics for Next Generation Molecular Dynamics Workflows* (NSF 1741057, 1841758). MD simulations used resources of the Oak Ridge Leadership Computing Facility (ALCC allocation BIP109), which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725, and an allocation at the National Energy Research Scientific Computing Center (NERSC, repository m1710) supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

References

[1] M. Osadchy and R. Kolodny, "Maps of protein structure space reveal a fundamental relationship between protein structure and function," *Biophysics and Computational Biology*, vol. 108, no. 30, 2011.

[2] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *The Shape and Structure of Proteins*. New York: Garland Science, 2002.

[3] D. KA and C. J, "The protein-folding problem, 50 years on," *Science*, vol. 338, no. 6110, pp. 1042–1046, November 2012.

[4] E. Krissinel, "On the relationship between sequence and structure similarities in proteomics," *Bioinformatics*, vol. 23, no. 6, pp. 717–723, March 2007.

[5] JeffWereszczynski and J. McCammon, "Statistical mechanics and molecular dynamics in evaluating thermodynamic properties of biomolecular recognition," *Q Rev Biophys*, vol. 45, no. 1, pp. 1–25, February 2012.

[6] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990.

[7] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015.

[8] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 04 2017.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[11] T. Estrada, J. Benson, H. Carrillo-Cabada, A. M. Razavi, M. A. Cuendet, H. Weinstein, E. Deelman, and M. Taufer, "Graphic encoding of macromolecules for efficient high-throughput analysis," in *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, ser. BCB '18. New York, NY, USA: ACM, 2018, pp. 315–324.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 770–778.

[13] T. Smith and M. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, no. 1, pp. 195 – 197, 1981.

[14] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443 – 453, 1970.

[15] D. Barkan, "A parallel implementation of the needleman-wunsch algorithm for global gapped pair-wise alignment," *J. Comput. Sci. Coll.*, vol. 17, no. 6, pp. 238–239, May 2002.

[16] S. Maleki, M. Musuvathi, and T. Mytkowicz, "Low-rank methods for parallelizing dynamic programming algorithms," *ACM Trans. Parallel Comput.*, vol. 2, no. 4, pp. 26:1–26:32, Feb. 2016.

[17] N. Kolker, R. Higdon, W. Broomall, L. Stanberry, D. Welch, W. Lu, W. Haynes, R. Barga, and E. Kolker, "Classifying proteins into functional groups based on all-versus-all blast of 10 million proteins," *OMICS*, vol. 15, no. 513, 2011.

[18] J. Whisstock and A. Lesk, "Prediction of protein function from protein sequence and structure," *Q Rev Biophys*, vol. 36, no. 3, 2003.

[19] O. Sensoy and H. Weinstein, "A mechanistic role of Helix 8 in GPCRs: Computational modeling of the dopamine D2 receptor interaction with the GIPC1-PDZ-domain," *Biochimica et biophysica acta*, vol. 1848, no. 4, pp. 976–983, Apr. 2015.

[20] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "Nucleic acids research," *Nature Structural Biology*, vol. 28, no. 1, 2000.

[21] J. Hou, G. E. Sims, C. Zhang, and S.-H. Kim, "A global representation of the protein fold space," *PNAS*, vol. 100, no. 5, 2002.

[22] E. Ie, J. Weston, W. S. Noble, and C. Leslie, "Multi-class protein fold recognition using adaptive codes," in *Proceedings of the 22Nd International Conference on Machine Learning*, ser. ICML '05. New York, NY, USA: ACM, 2005, pp. 329–336.

[23] M. Biba, F. Esposito, S. Ferilli, T. M. A. Basile, and N. Di Mauro, *Multi-class Protein Fold Recognition Through a Symbolic-Statistical Framework*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 666–673.

[24] Y. Nakamura, A. Kaneko, and T. Itoh, "An accelerated pocket extraction and evaluation technique for druggability analysis with

protein surfaces," in *SIGGRAPH Asia 2011 Posters*, ser. SA '11. New York, NY, USA: ACM, 2011, pp. 31:1–31:1.

[25] R. J. Morris, R. J. Najmanovich, A. Kahraman, and J. M. Thornton, "Real spherical harmonic expansion coefficients as 3d shape descriptors for protein binding pocket and ligand comparisons," *Bioinformatics*, vol. 21, no. 10, 2005.

[26] Y. Wang, W. Ling-Yun, J.-H. Zhang, Z.-W. Zhan, Z. Xiang-Sun, and C. Luonan, "Evaluating protein similarity from coarse structures," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 6, no. 4, pp. 583–593, Oct. 2009.

[27] L. Ellingson and J. Zhang, "An efficient algorithm for matching protein binding sites for protein function prediction," in *Proceedings of the 2Nd ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, ser. BCB '11. ACM, 2011, pp. 289–293.

[28] S. Kim, S. Lee, and H. Yu, "Indexing methods for efficient protein 3d surface search," in *Proceedings of the ACM Sixth International Workshop on Data and Text Mining in Biomedical Informatics*, ser. DTMBIO '12. New York, NY, USA: ACM, 2012, pp. 41–48.

[29] M. Novic and M. Randic, "Representation of proteins as walks in 20-d space," *SAR QSAR Environ Res*, vol. 19, no. 3, 2008.

[30] A. Gramada and P. E. Bourne, "Multipolar representation of protein structure," *BMC Bioinformatics*, vol. 67, no. 242, 2006.

[31] B. Zhang, T. Estrada, P. Cicotti, P. Balaji, and M. Taufer, "Enabling scalable and accurate clustering of distributed ligand geometries on supercomputers," *Parallel Computing*, vol. 63, pp. 38 – 60, 2017.

[32] B. Zhang and T. Estrada and P. Cicotti and P. Balaji and M. Taufer, "Accurate scoring of drug conformations at the extreme scale," in *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, May 2015, pp. 817–822.

[33] H. Chen, O. Engkvist, Y. Wang, M. Olivecrona, and T. Blaschke, "The rise of deep learning in drug discovery," *Drug Discovery Today*, Jan. 2018.

[34] M. Zhang, Q. Su, Y. Lu, M. Zhao, and B. Niu, "Application of Machine Learning Approaches for Protein-protein Interactions Prediction," *Medicinal Chemistry (Shariqah (United Arab Emirates))*, vol. 13, no. 6, pp. 506–514, 2017.

[35] K. Paliwal, J. Lyons, and R. Heffernan, "A Short Review of Deep Learning Neural Networks in Protein Structure Prediction Problems," *Advanced Techniques in Biology & Medicine*, vol. 3, no. 3, pp. 1–2, Sep. 2015.

[36] Z. Li and Y. Yu, "Protein secondary structure prediction using cascaded convolutional and recurrent neural networks," ser. IJCAI'16. New York, New York, USA: AAAI Press, 2016, pp. 2560–2567.

[37] S. Wang and J. Xu, "De Novo Protein Structure Prediction by Big Data and Deep Learning," *Biophysical Journal*, vol. 112, no. 3, p. 55a, Feb. 2017.

[38] R.Evans, J.Jumper, J.Kirkpatrick, L.Sifre, T.F.G.Green, A. C.Qin, A.Nelson, A.Bridgland, H.Penedones, S.Petersen, K.Simonyan, S.Crossan, D.T.Jones, D.Silver, K.Kavukcuoglu, D.Hassabis, and A.W.Senior, "De novo structure prediction with deep-learning based scoring," *Nature*, vol. 477, no. 7366, Dec. 2018.

[39] J. Hou, B. Adhikari, and J. Cheng, "DeepSF: deep convolutional neural network for mapping protein sequences to folds," *Bioinformatics*, vol. 34, no. 8, pp. 1295–1303, Apr. 2018.

[40] S. P. Nguyen, Z. Li, D. Xu, and Y. Shang, "New Deep Learning Methods for Protein Loop Modeling," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2017.

[41] H. Li, J. Hou, B. Adhikari, Q. Lyu, and J. Cheng, "Deep learning methods for protein torsion angle prediction," *BMC Bioinformatics*, vol. 18, p. 417, sep 2017.

[42] R. Poplin, P.-C. Chang, D. Alexander, S. Schwartz, T. Colthurst, A. Ku, D. Newburger, J. Dijamco, N. Nguyen, P. T. Afshar, S. S. Gross, L. Dorfman, C. Y. McLean, and M. A. DePristo, "Creating a universal snp and small indel variant caller with deep neural networks," *bioRxiv*, 2018.

[43] M. Kulmanov, M. A. Khan, R. Hoehndorf, and J. Wren, "DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier," *Bioinformatics*, vol. 34, no. 4, pp. 660–668, Feb. 2018.

[44] X. Liu, "Deep Recurrent Neural Network for Protein Function Prediction from Sequence," *arXiv:1701.08318 [cs, q-bio, stat]*, Jan. 2017, arXiv: 1701.08318.

[45] R. Cao, C. Freitas, L. Chan, M. Sun, H. Jiang, and Z. Chen, "Prolango: Protein function prediction using neural machine translation based on a recurrent neural network," *arXiv:1710.07016 [cs, q-bio]*, Oct. 2017, arXiv: 1710.07016.

[46] A. Mitrofanova, V. Pavlovic, and B. Mishra, "Prediction of protein functions with gene ontology and interspecies protein homology data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 3, pp. 775–784, May 2011.

[47] W. Li, B. Ma, and K. Zhang, "Optimizing spaced$k$-mer neighbors for efficient filtration in protein similarity search," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 2, pp. 398–406, March 2014.

[48] D. Bandyopadhyay, J. Huan, J. Liu, J. Prins, J. Snoeyink, W. Wang, and A. Tropsha, "Functional neighbors: Inferring relationships between nonhomologous protein families using family-specific packing motifs," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 5, pp. 1137–1143, Sept 2010.

[49] N. Bonnel and P. Marteau, "Lna: Fast protein structural comparison using a laplacian characterization of tertiary structure," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 5, pp. 1451–1458, Sept 2012.

[50] B. Zhang, T. Estrada, P. Cicotti, and M. Taufer, "Enabling in-situ data analysis for large protein folding trajectory datasets," in *IEEE International Parallel and Distributed Processing Symposium*, 2014.

[51] T. Johnston, B. Zhang, A. Liwo, S. Crivelli, and M. Taufer, "In situ data analytics and indexing of protein trajectories," *J Comput Chem.*, vol. 38, no. 16, 2017.

[52] G. Ramachandran, C. Ramakrishnan, and V. Sasisekharan, "Multipolar representation of protein structure," *Journal of Molecular Biology*, vol. 7, no. 95, 1963.

[53] T. Ooi and K. Nishikawa, "Conformation of biological molecules and polymers," *E. D. and Pullman, B., Eds.*, pp. 173–187, 1973.

[54] M. N. Liebman, C. A. Venanzi, and H. Weinstein, "Structural analysis of carboxypeptidase a and its complexes with inhibitors as a basis for modeling enzyme recognition and specificity," *Biopolymers*, vol. 24, no. 9, pp. 1721–1758, 1985.

[55] H. Nymeyer and H.-X. Zhou, "A Method to Determine Dielectric Constants in Nonhomogeneous Systems: Application to Biological Membranes," *Biophysical Journal*, vol. 94, no. 4, pp. 1185–1193, Feb. 2008.

[56] H. Berman, K. Henrick, and H. Nakamura, "Announcing the worldwide Protein Data Bank," *Nature Structural Biology*, vol. 980, no. 10, 2003.

[57] T. G. O. Consortium, "Gene ontology consortium," http://www.geneontology.org/.

[58] The Gene Ontology Consortium., "Expansion of the gene ontology knowledgebase and resources," *Nucleic Acids Res.*, vol. 4, no. 45, 2017.

[59] M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, J. Cherry, A. Davis, K. Dolinski, S. Dwight, J. Eppig, M. Harris, D. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. Matese, J. Richardson, M. Ringwald, G. Rubin, and G. Sherlock, "Gene ontology: tool for the unification of biology. the gene ontology consortium," *Nat Genet.*, vol. 25, no. 1, 2000.

[60] L. Mosley, "A balanced approach to the multi-class imbalance problem," *ICJV*, 2013. [Online]. Available: https://lib.dr.iastate.edu/etd/13537

[61] S. G. Rasmussen, B. T. DeVree, Y. Zou, A. C. Kruse, K. Y. Chung, T. S. Kobilka, F. S. Thian, P. S. Chae, E. Pardon, D. Calinski, J. M. Mathiesen, S. T. A. Shah, J. A. Lyons, M. Caffrey, S. H. Gellman, J. Steyaert, G. Skiniotis, W. I. Weis, R. K. Sunahara, and B. K. Kobilka, "Crystal Structure of the 2adrenergic Receptor-Gs protein complex," *Nature*, vol. 477, no. 7366, pp. 549–555, Jul. 2011.

[62] G. Morra, A.M. Razavi, K. Pandey, H. Weinstein, A.K. Menon, G. Khelashvili, "Mechanisms of lipid scrambling by the g protein-coupled receptor opsin," *Structure*, vol. 26, no. 2, pp. 356–367, Feb 2018.

[63] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization," *CoRR*, vol. abs/1610.02391, 2016.

[64] Y. Ji, V. Postis, Y. Wang, M. Bartlam, and A. Goldman, "Transport mechanism of a glutamate transporter homologue GltPh," *Biochemical Society Transactions*, vol. 44, no. 3, pp. 898–904, Jun. 2016.

[65] N. Akyuz, E.R. Georgieva, Z. Zhou, S. Stolzenberg, M.A. Cuendet, G. Khelashvili, A.B. Altman, D.S. Terry, J.H. Freed, H. Weinstein, O. Boudker, S.C. Blanchard, "Transport domain unlocking sets the uptake rate of an aspartate transporter," *Nature*, vol. 518, no. 7537, pp. 68–73, Feb 2015.

Hector Carrillo is a first year Ph.D. student in Computer Science and research assistant for Trilce Estrada at the University of New Mexico. He was an intern at VisionQuest Biomedical from 2017 to 2018 where he researched transfer learning for diabetic retinopathy detection as well as generative adversarial networks for retinal image conversion. His research interest includes unsupervised learning and generative adversarial networks.

Jeremy Benson is a Ph.D. student in Computer Science at the University of New Mexico in Trilce Estrada's lab. His interests lie on the intersection of artificial intelligence and distributed computing, focusing on data representation and classification, especially those tied to medical applications. His ongoing work with VisionQuest Biomedical is in developing and deploying low-cost, large scale machine learning solutions in telemedical operations.

Asghar M. Razavi is a postdoctoral associate at the Department of Physiology and Biophysics at Weill Cornell Medical College of Cornell University. His Ph.D. is in computational chemistry and biophysics from Temple University, Philadelphia, USA. His research at Weinstein lab is molecular level quantitative kinetic models for thermodynamics, kinetics, and conformational pathways during function of neurotransmitter transporters and G protein-coupled receptors.

Brianna Mulligan is a Master's student in Computer Science at the University of New Mexico and a Project Assistant for Elaine L. Bearer in the Department of Pathology at the University of New Mexico School of Medicine. Her research interests address the intersection of Biochemistry and Computer Science, and include biomarker discovery, omics analysis, and molecular dynamic simulation.

Michel A. Cuendet is an Adjunct Assistant Professor of Computational Biomedicine in Physiology and Biophysics of Weill Cornell Medical College, a Senior Research Scientist in the Molecular Modeling Group at the Swiss Institute of Bioinformatics, and a Principal Investigator at the Lausanne University Hospital. His PhD is from ETH Zurich, he was Postdoctoral Fellow at the Swiss Institute of Bioinformatics, and later a Senior Research Scientist in the Department of Chemistry at New York University. His interests are in molecular modeling using enhanced sampling and free energy techniques with machine learning.

Harel Weinstein, D.Sc. is the Maxwell Upson Professor of Physiology and Biophysics and Chairman of the Department of Physiology and Biophysics, and Founder and Director of the Institute for Computational Biomedicine (ICB), a pioneering academic and research unit responsible for quantitative understandings of physiological function and disease, at Weill Cornell Medical College. A Tri-Institutional Professor, he holds appointments at Rockefeller University, Sloan-Kettering Institute and Cornell University. The Weinstein lab studies complex biomolecular systems with methods of molecular and computational biophysics, bioinformatics and mathematical modeling to learn about structural and dynamic mechanisms of cellular components. Biomedical end points include neurotransmission in health and disease, drug abuse mechanisms, and cancer.

Michela Taufer holds the Jack Dongarra Professorship in High Performance Computing in the Department of Electrical Engineering and Computer Science at the University of Tennessee Knoxville. Her undergraduate degree in Computer Engineering is from the University of Padova (Italy) and her doctoral degree in Computer Science is from the Swiss Federal Institute of Technology (Switzerland). She was a Postdoctoral Fellow at the University of California San Diego and The Scripps Research Institute, where she worked on interdisciplinary projects in computer systems and computational chemistry. Her research interests include scientific applications, scheduling and reproducibility challenges, and big data analytics.

Trilce Estrada is an assistant professor of Computer Science at the University of New Mexico. Her research interests span the intersection of Machine Learning, Distributed Systems, Big Data, and their applications to interdisciplinary problems. She obtained her PhD in computer science from University of Delaware, masters degree from the National Institute of Astrophysics Optics and Electronics (INAOE), and her undergraduate degree in Informatics from The University of Guadalajara, Mexico.