Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN

SCIENZE BIOTECNOLOGICHE, BIOCOMPUTAZIONALI, FARMACEUTICHE E FARMACOLOGICHE

Ciclo 35

Settore Concorsuale: 05/E1 - BIOCHIMICA GENERALE

Settore Scientifico Disciplinare: BIO/10 - BIOCHIMICA

DEEP LEARNING AND EMBEDDINGS FOR PROBLEMS OF COMPUTATIONAL BIOLOGY

Presentata da: Matteo Manfredi

Coordinatore Dottorato

Maria Laura Bolognesi

Supervisore

Pier Luigi Martelli

Esame finale anno 2023

Abstract

The development of Next Generation Sequencing promotes Biology in the Big Data era. The ever-increasing gap between proteins with known sequences and those with a complete functional annotation requires computational methods for automatic structure and functional annotation. My research has been focusing on proteins and led so far to the development of three novel tools, DeepREx, E-SNPs&GO and ISPRED-SEQ, based on Machine and Deep Learning approaches.

DeepREx computes the solvent exposure of residues in a protein chain. This problem is relevant for the definition of structural constraints regarding the possible folding of the protein. DeepREx exploits Long Short-Term Memory layers to capture residue-level interactions between positions distant in the sequence, achieving state-of-the-art performances. With DeepRex, I conducted a large-scale analysis investigating the relationship between solvent exposure of a residue and its probability to be pathogenic upon mutation.

E-SNPs&GO predicts the pathogenicity of a Single Residue Variation. Variations occurring on a protein sequence can have different effects, possibly leading to the onset of diseases. E-SNPs&GO exploits protein embeddings generated by two novel Protein Language Models (PLMs), as well as a new way of representing functional information coming from the Gene Ontology. The method achieves state-of-the-art performances and is extremely time-efficient when compared to traditional approaches.

ISPRED-SEQ predicts the presence of Protein-Protein Interaction sites in a protein sequence. Knowing how a protein interacts with other molecules is crucial for accurate functional characterization. ISPRED-SEQ exploits a convolutional layer to parse local context after embedding the protein sequence with two novel PLMs, greatly surpassing the current state-of-the-art.

All methods are published in international journals and are available as user-friendly web servers. They have been developed keeping in mind standard guidelines for FAIRness (FAIR: Findable, Accessible, Interoperable, Reusable) and are integrated into the public collection of tools provided by ELIXIR, the European infrastructure for Bioinformatics.

Index

Abstract 1
Index 3
1. Introduction 5
1.1. Protein biosynthesis 5
1.2. Protein composition, folding and misfolding 5
1.3. Proteins in the Next Generation Sequencing era 7
2. Problems of Computational Biology 9
2.1. Solvent-Accessible Surface Area 9
2.2. Single Residue Variations and Pathogenicity 10
2.3. Protein-Protein Interaction Sites 11
3. Machine Learning for Computational Biology 15
3.1. Models of Supervised Learning 17
3.1.1. Artificial Neural Networks 19
3.2. Encoding Protein Sequences 24
3.3. Protein Language Models 26
3.4. Training and validating machine-learning models 28
3.5. Scoring indexes 30
3.6. Deployment of a new tool 31
4. DeepREx: Prediction of protein solvent accessibility from sequence 33
4.1. Materials and Methods 33
4.1.1. Datasets 33
4.1.2. Model Architecture 34
4.2. Results and Discussion 36
4.2.1. Evaluation and Benchmarking 36
4.2.2. DeepREx-WS to assist surface engineering 37
4.2.3. Linking RSA and pathogenicity of SRV 38
5. E-SNPs&GO: Prediction of variant pathogenicity 43
5.1. Materials and Methods 43
5.1.1. Datasets 43
5.1.2. Model Architecture 44

5.2. Results and Discussion 46
5.2.1. Evaluation and Benchmarking 46
5.2.2. Predictions on Variations of Uncertain Significance 48
6. ISPRED-SEQ: Prediction of Protein-Protein Interaction sites 51
6.1. Materials and Methods 51
6.1.1. Datasets 51
6.1.2. Model Architecture 52
6.2. Results and Discussion 54
6.2.1. Evaluation and Benchmarking 54
7. Conclusions and perspectives 57
Acknowledgments 59
References 61
Appendix: Publications 67

1. Introduction

1.1. Protein biosynthesis

Proteins are involved in every process which takes place inside a living cell, including their own biosynthesis, regulatory activities and stimuli response. Furthermore, they confer structure to cells and generate transmembrane channels for the selective transport of ions and other molecules. Proteins involved in biocatalysis are called enzymes. They enable important reactions whose products power life processes. Protein synthesis is a fundamental biological process that stems out of the genetic information included in the DeoxyriboNucleic Acid (DNA) structure. The biological process as a whole is composed of several intermediate steps including transcription, where the information encoded into DNA is transcribed into messenger RiboNucleic Acid (mRNA) molecules, and translation, where mRNA encodes protein synthesis at the level of the ribosomes. Encoding is made possible through a redundant and species-specific genetic code which relies on nucleotide triplets. Therefore, the genetic material of both prokaryotes and eukaryotes is at the origin of the flow of chemical information which promotes protein biosynthesis (Voet and Voet, 2011; Nelson and Cox, 2021).

1.2. Protein composition, folding and misfolding

At the level of the ribosomes, twenty different amino acids, each bound to specific RNA transfer molecules, such as transfer aminoacyl RNAs (aa-tRNAs), are selected from the cell cytoplasm on the basis of the codon (exposed in the mRNA-ribosome complex) -anticodon (exposed in the aa-tRNA) base-pairing principle. By this, and thanks to specific enzymes in the ribosomes, a peptide bond is formed among different amino acids. A protein can therefore be regarded as a heteropolymer of twenty different residues, which differs by the presence of a

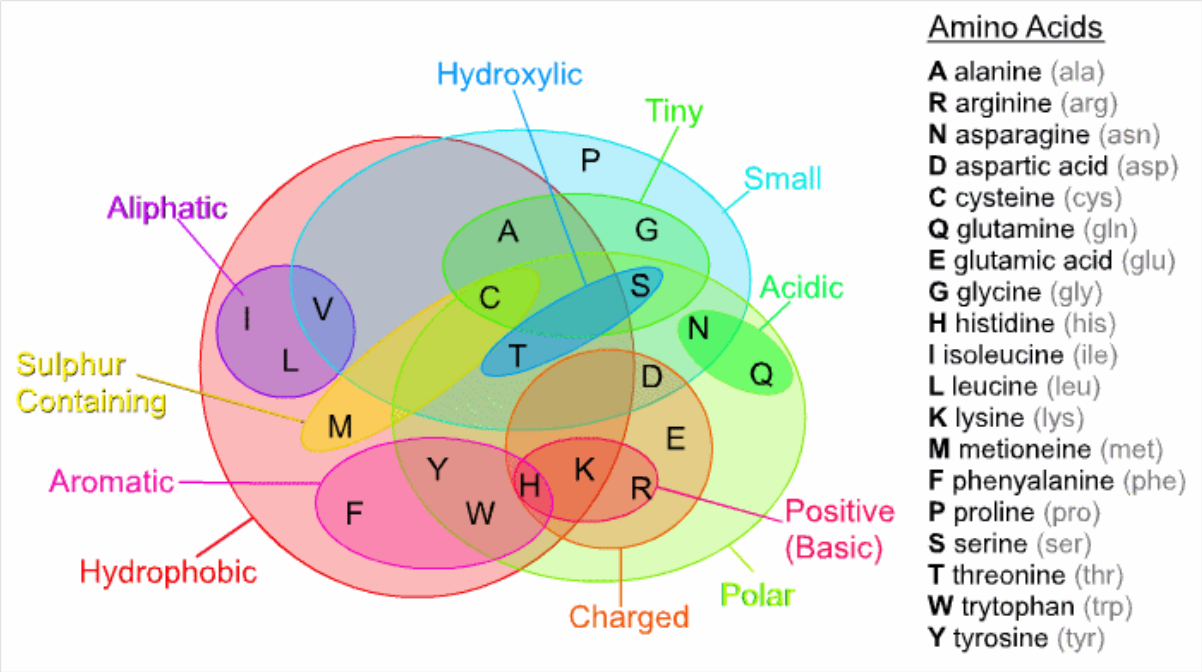


Figure 1. Venn diagram showing the classification of residues based on the physicochemical properties of their lateral side chains. The rightmost column reports the one-letter code, the full name and the three-letter code of each residue (Esquivel *et al.*, 2013).

peculiar side chain. Based on this, all residues have different physicochemical properties. As shown in Figure 1, they are routinely classified into apolar (Glycine/G, Alanine/A, Valine/V, Proline/P, Leucine/L, Isoleucine/I, Methionine/M), aromatic (Phenylalanine/F, Tryptophan/W, Tyrosine/Y), polar (Serine/S, Threonine/T, Cysteine/C, Asparagine/N, Glutamine/Q, Histidine/H) and charged (Aspartic Acid/D, Glutamic Acid/E, Lysine/K, Arginine/R) residues. The peculiar sequence of residues is what differentiates a protein and it is known as protein primary structure (1D structure). The heteropolymer in a polar environment tends to acquire a stable structure that ultimately is due to a balancing of the different residue propensities to interact with the water dipoles.

The process that leads proteins to assume a stable three-dimensional structure (3D) in polar solvents, and therefore to assume the conformation with the lowest Gibbs free energy ($\Delta G_{folding}$), is called protein folding (Anfinsen, 1973). The folding process is spontaneous and is characterised by ΔG values ranging from 0 to -50 kcal/mole, depending on the protein (Voet

and Voet, 2011; Nelson and Cox, 2021). The correct folding, or 3D structure, is important due to the structure-function relationship which implies that if residues are changed in the wild-type protein, the overall structure can be compromised. Sequence, structure and function are therefore tightly correlated and alterations at the level of the sequence can lead to changes in the 3D structure, possibly altering the functionality of the protein. In humans, this may be associated with genetic and/or somatic diseases.

Since proteins are gene products, such alterations can derive from errors during DNA replication, namely mutations. A mutation in a protein-coding portion of the gene can modify the translation of the protein, leading to a different residue in the corresponding position called variation. Variations in a protein can be neutral to its final behaviour, especially when they interest a region which is not related to the function. Otherwise, the final activity of the molecule can be altered. Given the crucial role of proteins in all pathways inside cells, disruptive mutations can in some cases lead to severe diseases (Krebs *et al.*, 2017).

1.3. Proteins in the Next Generation Sequencing era

In the era of Next Generation Sequencing (NGS), an increasing number of protein-coding genes have been sequenced. Nonetheless, experimental methods for resolving the three-dimensional structure of a protein can hardly cope with the number of new sequences published in public databases. The two main sources for protein sequences and structures are UniProt (UniProt Consortium, 2023) and the Protein Data Bank (PDB) (Berman *et al.*, 2000), respectively. UniProt is split into two sections, Swiss-Prot containing the protein sequences that are manually annotated and TrEMBL containing all known sequences lacking manual annotation. A gap is evident when comparing the number of entries in the three datasets. Indeed, Swiss-Prot accounts for 568,744 sequences (20,404 of which are human), TrEMBL accounts for 229,580,745 sequences (187,001 of which are human), and PDB accounts for 200,375 protein

structures (of which 60,808 are human). Provided that knowledge of the structure of a protein is crucial for its functional annotation, there is a strong urgency to develop computational methods suited to perform it automatically starting from the sequence alone. Moreover, it is of the utmost importance to make these tools easily available to the scientific community through public infrastructures such as ELIXIR, the European infrastructure for Bioinformatics (https://elixir-europe.org/).

One of the most prominent breakthroughs of recent years in the field of structure prediction from the sequence is the development of AlphaFold (Jumper *et al.*, 2021). This machine learning-based method solves the problem of inferring the most likely three-dimensional protein conformation given its sequence, learning Multiple Sequence Alignments, residue contact maps and correlated mutations of all the structures available in PDB. AlphaFold achieved very high results in the 14th Critical Assessment of Protein Structure Prediction (CASP14) (Kryshtafovych *et al.*, 2021). Despite its undeniable success, AlphaFold is still unable to produce acceptable results for many proteins. Research in the field is still very active. Indeed, even when the structure of a protein is experimentally known, many questions can still arise, such as those that I'm addressing in this work, including the understanding of the effect of protein variations, especially in relation to the onset of pathological conditions.

2. Problems of Computational Biology

2.1. Solvent-Accessible Surface Area

The Solvent-Accessible Surface Area (SASA) or Accessible Surface Area (ASA) in a folded protein is the surface that the molecule exposes to the polar solvent including all the exposed lateral side chains (Lee and Richards, 1971). The value, expressed in Ångströms^2, can be accurately computed when the structure of the protein is known, by adopting several computational methods (Ausaf Ali *et al.*, 2014). The most widely used is called Define Secondary Structure of Proteins (DSSP) (Touw *et al.*, 2015), which is based on the rolling ball algorithm developed by Shrake and Rupley (Shrake and Rupley, 1973). ASA values are routinely transformed into relative ones, denoted as Relative Solvent Accessibility (RSA), allowing for a better comparison between different residue types as well as providing values scaled between 0 and 1. This conversion is done by dividing the ASA of a residue by its theoretical maximal accessibility, routinely computed by considering it into the tripeptide Gly-X-Gly. Several scales have been proposed in the literature (Rost and Sander, 1994; Tien *et al.*, 2013; Rose *et al.*, 1985). For our studies, we adopted the one compiled by Rost and Sanders, 1994.

When a protein is missing its experimentally solved 3D structure, it can be helpful to know the exposure of its residues for determining folding and stability (Miller *et al.*, 1987), This knowledge can also help in determining possible interaction interfaces (Savojardo *et al.*, 2017; Porollo and Meller, 2007) and in characterizing structural and functional motifs (Savojardo, Manfredi, *et al.*, 2020; Martelli *et al.*, 2016; Savojardo *et al.*, 2019). With the advent of machine learning in Bioinformatics, many tools have been developed for RSA prediction. The methods can produce as output a putative value for the residue RSA (Klausen *et al.*, 2019; Hanson *et al.*, 2019; Singh *et al.*, 2021) and/or eventually a classification into two (Drozdetskiy *et al.*,

2015; Wu *et al.*, 2017) or more (Deng *et al.*, 2017; Kaleel *et al.*, 2019) classes of exposure. Methods of the first type score with Pearson Correlation Coefficient (PCC) values up to 0.82, while methods of the second type score with Matthews Correlation Coefficient (MCC) values up to 0.65.

Due to the high correlation of RSA to other residue-level characteristics, many tools are also trained to compute multiple outputs, including secondary structure (Drozdetskiy *et al.*, 2015; Klausen *et al.*, 2019; Hanson *et al.*, 2019), coiled-coil regions (Drozdetskiy *et al.*, 2015), contact numbers (Deng *et al.*, 2017; Hanson *et al.*, 2019), structural disorder (Klausen *et al.*, 2019) or backbone dihedral angles (Klausen *et al.*, 2019; Hanson *et al.*, 2019). In other cases, external tools predicting features are exploited to construct the input for the prediction of RSA (Wu *et al.*, 2017; Fan *et al.*, 2016; Tarafder *et al.*, 2018). Recent methods routinely employ deep architectures that proved very effective, especially when adopting complex architectures of neural networks that can capture context information from the whole sequence (Kaleel *et al.*, 2019; Deng *et al.*, 2017; Hanson *et al.*, 2019). More details regarding different types of architectures are given in Section 3.1.

2.2. Single Residue Variations and Pathogenicity

Single Nucleotide Polymorphisms (SNPs) occurring in protein-coding regions can lead to Single Residue Variations (SRVs) on the protein residue sequence. These variations may have several effects on the protein function, changing its abundance, activity, specificity and/or affinity towards the interaction with other molecules (Vihinen, 2021). Residue changes in the sequence of a protein are not necessarily harmful, but they lead in some cases to the onset of a pathological condition (Lappalainen and MacArthur, 2021). Public databases such as HUMSAVAR (UniProt Consortium, 2023) and ClinVar (Landrum *et al.*, 2018) include this information, classifying SRVs into neutral (or benign) and disease-related (or pathogenic).

However, most known variations remain of Uncertain Significance (VUS) lacking a correlation with specific diseases. The task of discriminating between pathogenic and neutral variations has been tackled computationally. Several methods have been proposed in the past and recent years to solve this problem.

Early methods such as SIFT (Ng and Henikoff, 2001) and PROVEAN (Choi *et al.*, 2012) were based on a statistical analysis of the conservation of residues in a set of homologous sequences from different organisms, as derived from Multiple Sequence Alignments (MSAs). The higher the level of conservation, the less the probability of being disease-associated. Although this simple strategy allowed them to reach satisfactory results (Matthews Correlation Coefficient values of 0.57) and MSA-based features are still widely adopted, recent methods exploit different machine learning architectures for the discriminative task (Adzhubei *et al.*, 2010; Calabrese *et al.*, 2009; Carter *et al.*, 2013; Jagadeesh *et al.*, 2016; Li *et al.*, 2009; Niroula *et al.*, 2015; Pejaver *et al.*, 2020; Raimondi *et al.*, 2017; Schwarz *et al.*, 2010; Yang *et al.*, 2022). Smart representations of different features of the protein and/or of the involved variations can improve method scores. Most notably, SNPs&GO (Calabrese *et al.*, 2009) was the first method to propose a way to encode functional annotations derived from Gene Ontology (GO) (Ashburner *et al.*, 2000) as a log-odd score, demonstrating that knowing the protein function could improve the overall performances. More recent methods are based on canonical approaches such as random forests (Raimondi *et al.*, 2017) and gradient tree boosting (Yang *et al.*, 2022), with very few examples of successful training with complex architectures (Pejaver *et al.*, 2020).

2.3. Protein-Protein Interaction Sites

Knowing how proteins interact with other biological entities is of extreme importance for understanding their function in the context of cell complexity. It is known that functional

membraneless protein aggregates play a role in metabolic biological processes (Savojardo, Martelli, *et al.*, 2020). In particular, the identification of residues involved in protein interactions, referred to as Protein-Protein Interaction (PPI) sites, can help in characterising molecular mechanisms at the basis of important biological processes. A common strategy relies on deriving from existing protein-protein interfaces, like those known with atomic resolution, the knowledge that needs to be transferred to other proteins in order to compute whether they can be part of a complex. Protein-protein interfaces are derived from protein complexes solved with X-ray crystallography, Nuclear Magnetic Resonance (NMR), alanine scanning mutagenesis or chemical cross-linking (Rodrigues *et al.*, 2015). Nevertheless, experimental methods can be too costly to be applied to large-scale studies and it is important to develop different computational tools to complement their applicability.

When the interacting partner is known, docking programs are routinely used to run accurate simulations, allowing the identification of the most likely interface. Machine learning-based approaches can identify pairs of interacting partners (Pan *et al.*, 2010). Alternatively, non-partner-specific PPI sites can be directly identified (Casadio *et al.*, 2022). Amongst these, some methods adopt structure-derived features to encode the input (Li *et al.*, 2012; Liu *et al.*, 2009; Savojardo *et al.*, 2012; Šikić *et al.*, 2009; Dong *et al.*, 2014). This makes them generally more accurate but limits their potential application. Other tools do not require as input the protein 3D structure and are more suited for large-scale analysis of proteomes, although with overall lower performances (Dhole *et al.*, 2014; Hosseini and Ilie, 2022; Li *et al.*, 2021; Stringer *et al.*, 2022; Wei *et al.*, 2016, 2015; B. Zhang *et al.*, 2019; Zhang and Kurgan, 2019). Most of these methods routinely employ external predictors to include putative structural properties in the input features, including Relative Surface Accessibility (Dhole *et al.*, 2014; Li *et al.*, 2021; Stringer *et al.*, 2022; Wei *et al.*, 2016, 2015; B. Zhang *et al.*, 2019; Zhang and Kurgan, 2019),

disorder (Li *et al.*, 2021; Zhang and Kurgan, 2019), secondary structure (Zhang and Kurgan, 2019; Stringer *et al.*, 2022) and/or protrusion indices (B. Zhang *et al.*, 2019).

3. Machine Learning for Computational Biology

Machine Learning (ML) is part of the field of Artificial Intelligence. It is based on models that can learn rules from data and/or experience in an automated way, surpassing the need to explicitly encode them (Bishop, 2006). ML research began in the early 1950s and was at its core based on statistical methods. Following important technological advancements, the complexity of models that were available for training kept increasing, leading in recent years to the advent of Deep Learning (DL). Nowadays, ML is adopted in an increasing number of research areas (Baldi, 2021). ML models can be classified into three main frameworks that use different training methods based on the availability of data and the type of task we want to perform, namely i) reinforcement learning, ii) supervised learning and iii) unsupervised learning.

Reinforcement learning is suited for learning optimal strategies to achieve a given goal (Kaelbling *et al.*, 1996). This requires a clear definition of all possible states of the model, including feedback that rewards the machine for reaching positive states. In this case, training is not carried out using curated datasets, but through exploration of the space of possible actions. The model is asked to try several (initially) random actions with the goal of maximizing positive rewards and adjusting the parameters that regulate its decision-making process. A clear example of the application of reinforcement learning is game theory, where a machine can learn how to optimally play a game after experiencing all possible scenarios and learning to prioritize actions that lead to a win.

Supervised Learning is mainly used to make statistical predictions based on known data, mostly for discriminative tasks, in particular, classification or regression (Mohri *et al.*, 2018). It is necessary to have large enough datasets of labelled data, i.e. data for which we know experimentally what the correct output should be. Models can be considered as functions that

take input data and use internal parameters to produce an output. During the training phase, models see the labelled dataset and perform (initially) random predictions that are compared with the real labels. The discrepancies between expected and computed values are iteratively used to adjust the set of internal parameters until the error rate drops under a certain threshold. In this way, general rules can be learnt and stored in the internal parameter values, allowing the model to annotate new data for which labels are not known.

Finally, **unsupervised learning** is the main approach for situations where the goal is discovering internal patterns within unclassified data (Hinton and Sejnowski, 1999). In this case, the main applications are clustering tasks and generative tasks, where models are asked to produce new data that mimic the one observed in training. For generative tasks, similarly to supervised learning, produced data are initially random and internal parameters are adjusted based on the dissimilarities to real data.

As research in the ML field advances, new approaches often integrate ideas from different training paradigms. An important example is self-supervised learning, a hybrid approach between supervised and unsupervised learning that aims to learn new representations of unlabelled data which can be used to perform downstream supervised learning tasks (Raina *et al.*, 2007). Regardless of the selected approach, all models are characterized by a number of free parameters and hyperparameters. The firsts represent the learning potential of the model and are adjusted during the training procedure. The latter are selected when the model is defined and they are routinely optimized through a grid search where many possible combinations are tested. Recent technological advancements focus on the optimization of high-throughput techniques. This allows the generation of huge amounts of data involving all different “omics” sciences (Pal *et al.*, 2020). In this scenario, data-driven ML-based methods become an efficient approach to exploit data available in public databases.

3.1. Models of Supervised Learning

In the field of supervised learning, several models have been proposed in the literature for solving different problems, ranging from statistical and shallow to complex ones. We propose in the following a list of the most adopted, including the ones we developed for solving our Computational Biology problems.

Naive Bayes classifiers are models based on the application of the Bayes theorem under the "naive" assumption that all features used to represent the input are conditionally independent. While this assumption is generally not true, this class of models proved to perform surprisingly well in several real-world situations (Zhang, May 17-19 2004). Even when this is not the case, they are often employed to set baselines that more complex models should surpass to prove they are indeed useful for the task.

Regression methods are based on fitting mathematical functions to model a relationship between input data and their expected output. The two most widely used methods are linear regression (Seber and Lee, 2003) and logistic regression (Menard, 2002). Both are linear models. This makes them very easy to train and interpret for simple problems.

Hidden Markov Models (HMMs) are probabilistic graphical models routinely used to model protein families starting from multiple sequence alignments (Durbin *et al.*, 1998). The graphical representation is composed of "state" nodes and "transition" edges, both modelled as probability distributions. Given a sequence of input symbols (residues), HMMs can emit a sequence of output symbols (features or labels). It is then possible to adopt them in supervised learning settings, adjusting the transition and emission probabilities based on available data. Common applications involve the prediction of residue-level protein features such as transmembrane topologies (Bystroff and Krogh, 2008; Martelli *et al.*, 2002) and sorting signals (Käll *et al.*, 2004).

Decision trees are models that learn to iteratively use features of the input to branch into a tree-like structure until it can generate a proper output. During the training phase, a decision tree learns from the dataset which features carry the most information gain, prioritizing those in the early stages of the predictions and setting appropriate thresholds to perform decisions. The main advantage of decision trees is their easy interpretability since the final tree provides meaningful information regarding the importance of the input features. This is especially helpful for selecting the most relevant features that can sufficiently characterize a dataset, reducing the dimensionality of the representation. Conversely, their main drawback is a prominent tendency to overfit, so that small changes in the training dataset will generate very different models, reducing their ability to generalize to new data.

Random Forests are ensemble methods based on a set of decision trees (Ho, 1995), offering a trade-off between interpretability and overfitting (Hastie *et al.*, 2009). In this case, the strategy is to build many decision trees and iteratively sample from the training dataset to train them. Each tree will learn from different data to generate different outputs. The final discrimination is made through a consensus of all the trained trees. Random Forests are more stable than Decision Trees. However, the number of trees may reduce the interpretability of the final results.

Support Vector Machines are models which can perform binary classification tasks by learning the optimal separating hyperplane in the space of the input features (Cortes and Vapnik, 1995). When data are linearly separable, the hyperplane maximizes the margin between the two classes, given by the distance from the plane of the closest data points from both sides, called support vectors. Learning is performed considering iteratively training examples and adjusting the internal parameters which define the position of the hyperplane. When the problem is not linearly separable, a different approach is adopted called soft margin.

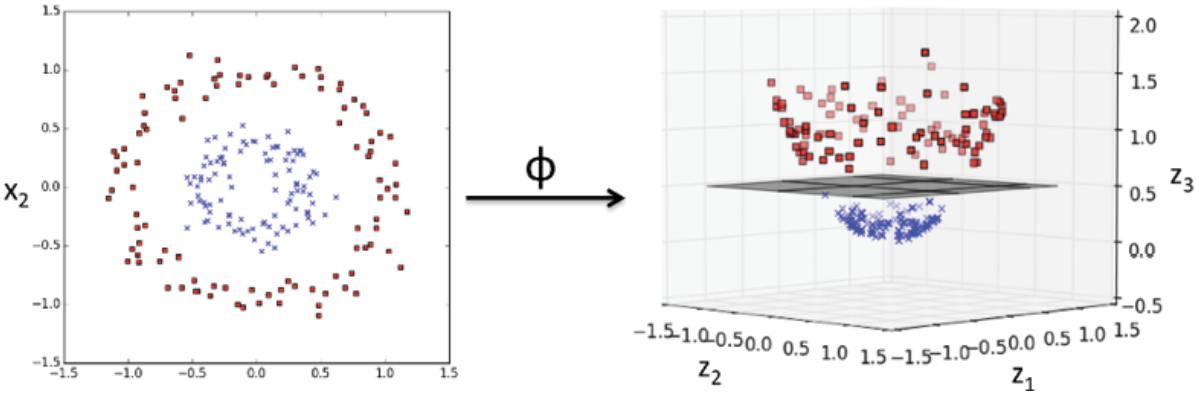


Figure 2. A kernel transformation to map a non-linearly separable two-dimensional problem into a linearly-separable problem in three-dimension. ϕ is an example of a transformation that defines a kernel, as the scalar product in the mapped space $[K(x, y) = \phi(x) \times \phi(x)]$.

Soft margin adopts a different loss function that maximizes the margin while minimizing the number of incorrect classifications. As displayed in Figure 2, it is also possible to include an approach called kernel trick, based on a kernel function which transforms the input space into a new space where a linear classification can solve the task (Boser *et al.*, 1992). Common kernels include polynomial, gaussian or sigmoid functions. This mathematical approach makes Support Vector Machines widely applicable. When adopting SVMs, the type of kernel function, the parameters of the kernel and the trade-off for the soft margin should always be extensively fine-tuned through a grid search. Due to their implementation, SVMs cannot be directly tuned to perform a multilabel classification, although it is possible to build an ensemble of models in which each SVM learns to discriminate one class from all the others.

3.1.1. Artificial Neural Networks

Artificial Neural Networks (ANNs) are models based on the connection between several basic computational units, referred to as artificial neurons. Initially developed to mimic biological neurons and their connections, artificial neurons are simple units which take one or more inputs and combine them through an activation function to produce an output. During the training

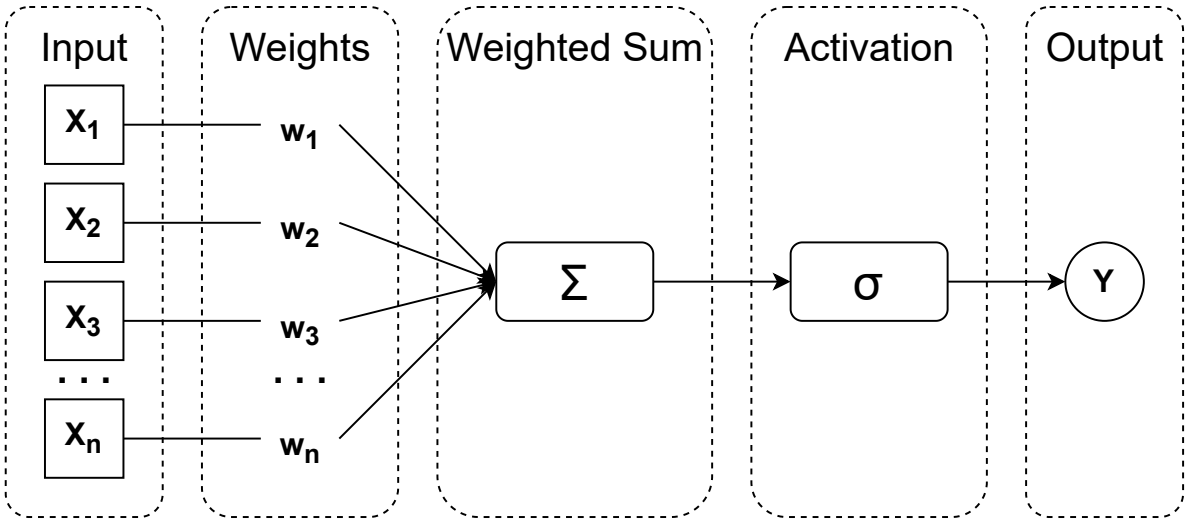


Figure 3. Representation of a perceptron. Features of the input (x_i) are multiplied by weights (w_i, free parameters of the model) and summed together (Σ). An activation function ($\boldsymbol{\sigma}$) then produces the output.

phase, weights corresponding to the connections between neurons are updated, adjusting the output produced by the network (Krogh, 2008). The type of neurons, the activation functions they use and the way they are combined determine the type of Neural Network we use.

Perceptrons are the simplest form of ANN. A perceptron, depicted in Figure 3, is a simple linear classifier that computes a weighted sum of the inputs and uses an activation function to produce an output. During training, the weights corresponding to the input features are adjusted with the goal of reducing the error (Rosenblatt, 1958). Similarly to the SVM, strategies are needed to allow the network to solve non-linearly separable problems.

Multilayer Perceptrons (MLPs) are networks composed of i) an input layer with as many neurons as the input dimension, ii) one or more hidden layers with a variable number of hidden neurons and iii) an output layer with as many neurons as the number of classes to be discriminated. This kind of network is also called feed-forward fully connected because each node receives the output of all the neurons from the previous layer and broadcasts its output to all neurons of the following layer. The presence of the hidden layers allows the model to map input data in a new space which can be used to better discriminate the output (Hornik *et al.*,

1989). Following technological advancements, including better-performing hardware infrastructures and algorithmic innovations (such as the inclusion of different loss and activation functions), it is now possible to properly train more complex network architectures with two or more hidden layers. These are referred to as **Deep Learning** procedures (Goodfellow *et al.*, 2016). Given large enough training datasets, Deep Learning can be employed to extract patterns that are impossible to capture with simple networks, referred to as shallow networks.

Convolutional Neural Network. Following the huge success of ANNs, researchers developed specialized networks for processing different types of data. Convolutional Neural Networks (CNNs) were specifically designed to analyze images, although they have been shown to perform very well even for sequence analysis (Abdel-Hamid *et al.*, 2014). Images are composed of adjacent pixels, each represented with a vector of colour intensities. Using a fully connected perceptron, we would need one neuron for each feature of each pixel, resulting in an exploding number of trainable parameters. Moreover, we would not exploit any information regarding the proximity of pixels to one another. The idea of CNNs is to use filters of fixed size to slide over the image (Lecun *et al.*, 1998), effectively sharing the same set of weights to process different parts of the input. While greatly reducing the number of needed parameters, the filter learns to pick up the presence of patterns in the image. In this way, several CNN layers using filters of different sizes can be stacked together, each learning to represent different kinds of local patterns and features. Not limited to the analysis of images, CNN can be successfully applied to sequential data (1-dimensional CNN, 1D CNN) to highlight specific patterns in the sequence (Kiranyaz *et al.*, 2021). 1D CNNs can be particularly useful when working with proteins. The use of a convolutional filter sliding over the length of the chain can mimic the widely adopted approach of sliding windows centred on the target residue. Each residue is thus encoded with a variable number of features (channels), similar to how pixels of an image are

encoded with their colour intensities. The filter iteratively processes groups of contiguous residues, capturing information relative to the local context in the protein sequence.

Recurrent Neural Network. MLPs and CNNs are both examples of feedforward neural networks. Alternatively, Recurrent Neural Networks (RNNs) adopt node cycles to generate an internal state which allows them to process data of variable length. This allows for taking into consideration the evolution in time of the analysed system. Similarly to convolution, a weight-sharing strategy allows reusing the same set of weights to parse sequential data. In the case of RNNs, neurons have feedback loops that keep track of analysed data by updating the internal weights after every timestep. In this way, the produced output will not only depend on features of the current timestep but rather on the whole series of observations. When considering protein sequences each residue is taken as a different timestep and two networks processing the sequence in both directions (bidirectional RNNs) are routinely combined together. This allows the network to produce an output at each position which depends on the whole sequence context, capturing even long-range dependencies. A very popular and powerful type of RNN is called Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). LSTMs adopt architectures based on so-called gated cells, complex neurons with an internal state which adjust the information flow, learning to "remember" or to "forget" in order to capture the most meaningful associations.

Transformer. Similarly to RNNs, Transformers are networks mainly built to process sequential data (Vaswani *et al.*, 2017). Their main advantage is the ability to process the whole input at the same time using a procedure called Attention which enables them to learn which parts of the input should be prioritized in order to optimize the output. As shown in Figure 4, Transformers are composed of two main components, a stack of encoder layers and a stack of decoder layers. The encoders generate representations which cast correlations between

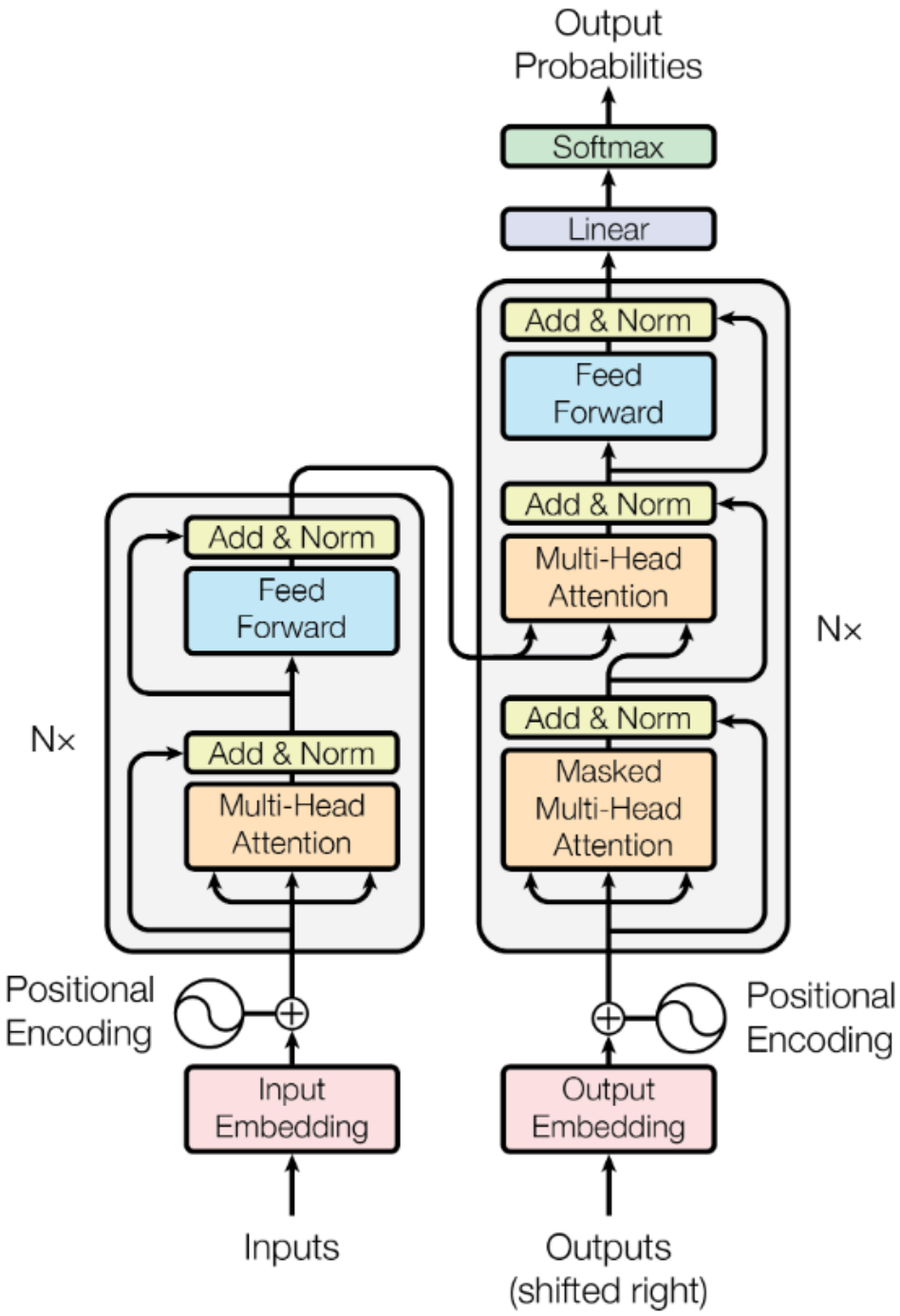


Figure 4. Model architecture of the Transformer, showing the encoder (left block) and the decoder (right block) (Vaswani *et al.*, 2017).

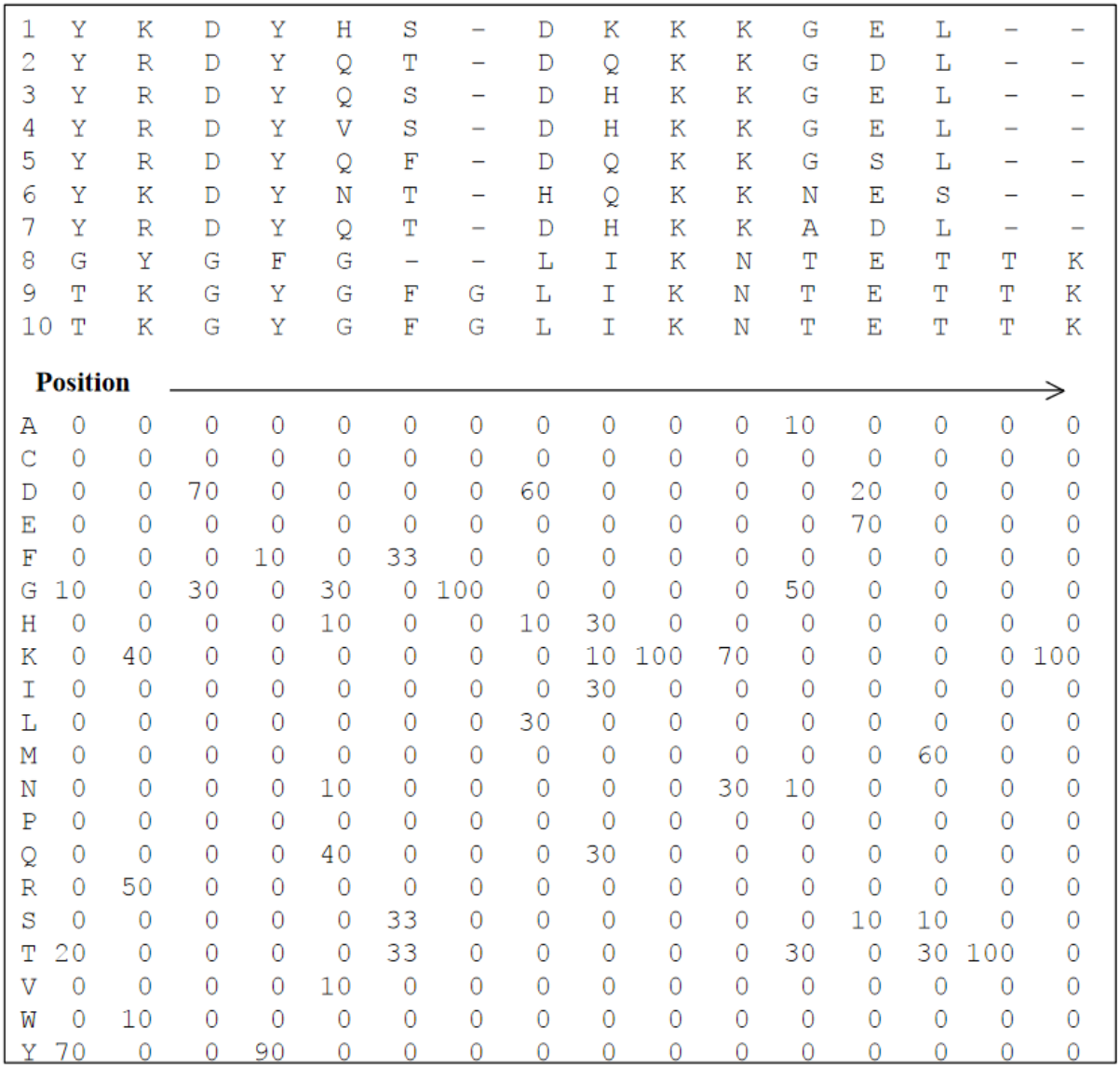
different parts of the input. Each decoder layer does the reverse operation, exploiting the information represented in the encodings to generate an output sequence. Transformers are particularly prone to parallelization, allowing full exploitation of the capabilities of modern Graphical Processing Units (GPUs) and/or Tensor Processing Units (TPUs). Transformers

have a major influence on the field of Natural Language Processing (NLP) and their application has been recently successful in developing Protein Language Models (PLMs) (Elnaggar *et al.*, 2020).

3.2. Encoding Protein Sequences

One of the main challenges in the application of machine learning-based methods to biological problems is representing input data. In the case of protein sequences, this requires the generation of vectors encoding each residue in the sequence. The simplest way to do this is by using a one-hot encoding representation. In this way, every residue is represented with a 20-dimensional vector where the position corresponding to the specific residue type (total of 20 different types) is set to 1, 0 otherwise. The one-hot encoding representation represents the 20 residues in the same way without taking into consideration the different physicochemical properties, which can be added as additional features. Moreover, it does not take into consideration the local context.

A major step forward is the introduction of the so-called evolutionary information, in the form of Position-Specific Scoring Matrices (PSSMs) or sequence profiles derived from the comparison of homologous proteins with a Multiple Sequence Alignment (MSA) procedure. As shown in Figure 5, from each column in the MSA, corresponding to a different residue position in the target sequence, a 20-dimensional vector is extracted, where each component is the frequency of a given residue type in the alignment position. This allows the inclusion in the input representation of the residue conservation at a given position in the sequence, possibly related to functionally relevant sites. The choice of the reference database is of the utmost importance for the quality of the MSA. The UniProt Reference Clusters (UniRef) (Suzek *et al.*, 2015) provide databases which are clustered at various levels of sequence identity, limiting their overall size while ensuring a balanced distribution between different protein families.



1	Y	K	D	Y	H	S	-	D	K	K	K	G	E	L	-	-
2	Y	R	D	Y	Q	T	-	D	Q	K	K	G	D	L	-	-
3	Y	R	D	Y	Q	S	-	D	H	K	K	G	E	L	-	-
4	Y	R	D	Y	V	S	-	D	H	K	K	G	E	L	-	-
5	Y	R	D	Y	Q	F	-	D	Q	K	K	G	S	L	-	-
6	Y	K	D	Y	N	T	-	H	Q	K	K	N	E	S	-	-
7	Y	R	D	Y	Q	T	-	D	H	K	K	A	D	L	-	-
8	G	Y	G	F	G	-	-	L	I	K	N	T	E	T	T	K
9	T	K	G	Y	G	F	G	L	I	K	N	T	E	T	T	K
10	T	K	G	Y	G	F	G	L	I	K	N	T	E	T	T	K

Position →

A	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	70	0	0	0	0	60	0	0	0	0	20	0	0	0
E	0	0	0	0	0	0	0	0	0	0	0	0	70	0	0	0
F	0	0	0	10	0	33	0	0	0	0	0	0	0	0	0	0
G	10	0	30	0	30	0	100	0	0	0	0	50	0	0	0	0
H	0	0	0	0	10	0	0	10	30	0	0	0	0	0	0	0
K	0	40	0	0	0	0	0	0	10	100	70	0	0	0	0	100
I	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0
N	0	0	0	0	10	0	0	0	0	0	30	10	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	40	0	0	0	30	0	0	0	0	0	0	0
R	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	33	0	0	0	0	0	0	10	10	0	0
T	20	0	0	0	0	33	0	0	0	0	0	30	0	30	100	0
V	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0
W	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y	70	0	0	90	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5. Example of a sequence profile (bottom) computed from a Multiple Sequence Alignment (MSA, top). In the MSA, 10 sequences are aligned and the sequence profile reports, position by position, the frequency of each residue type (in percentage).

While proving extremely successful for many different applications, sequence profiles have two main drawbacks. First, computing MSAs is a time-consuming procedure that scales linearly with the number of sequences and this affects the time needed in large-scale analysis. Second, the performance of a method adopting PSSMs to encode sequences is very dependent on the quality of the MSA.

3.3. Protein Language Models

Taking inspiration from recent successes in the Natural Language Processing (NLP) field (Ofer *et al.*, 2021), Protein Language Models (PLMs) leverage the huge amount of proteins stored in public databases for learning mathematical representations (embeddings) of protein sequences and residues. PLMs require very deep neural network architectures to be trained on hundreds of millions of protein sequences, an effort needing huge resource investments and weeks, if not months, of computation on high-performance Tensor Processing Units (TPUs) and/or Graphical Processing Units (GPUs) (Elnaggar *et al.*, 2020; Rives *et al.*, 2021). While this is indeed a huge limiting factor for their development, the main advantage is that, once trained, PLMs are extremely fast in the generation of new embeddings, requiring seconds of computation for a single sequence, or at most hours for whole proteomes, even on a single consumer-grade desktop machine. Embeddings generated by PLMs have been proven to be effective in encoding important properties regarding the evolution and the syntax of proteins (Bepler and Berger, 2021). For this reason, representations based on PLMs can be adopted in place of traditional PSSMs or other hand-crafted features.

Several pre-trained PLMs are available in the literature (Alley *et al.*, 2019; Asgari and Mofrad, 2015; Elnaggar *et al.*, 2020; Heinzinger *et al.*, 2019; Rives *et al.*, 2021; Strodthoff *et al.*, 2020; Meier *et al.*, 2021; Lin *et al.*, 2022), mainly differing in the type of architecture and in the training dataset (Bepler and Berger, 2021). Most models are based on Transformers (see Section 3.1.1 for a brief description), with three prominent strategies. The first utilizes decoder-only layers characterized by an attention method where each position in the decoder is connected to all positions of the previous layer up to that point (Figure 6, left). These models, called auto-regressive, are able to generate protein-like sequences. This allows a training procedure based on the prediction of the next residue in a sequence and the comparison with

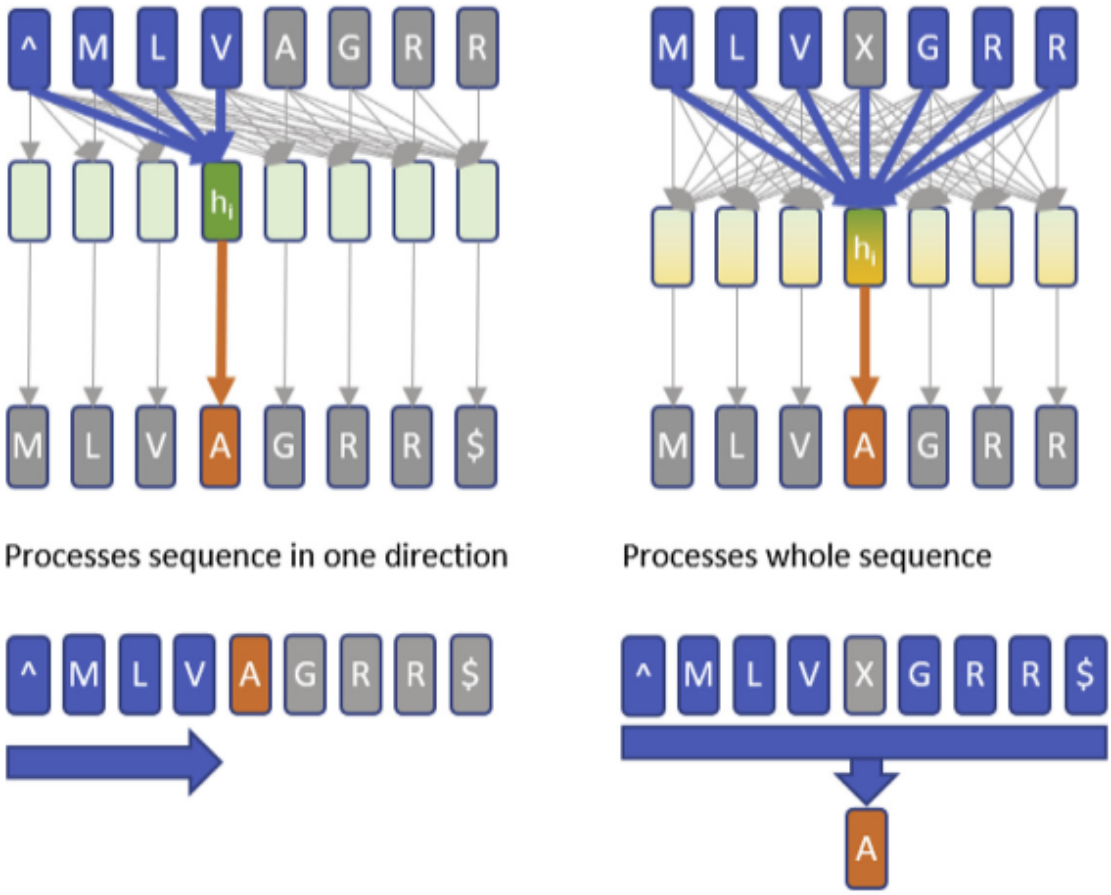


Figure 6. Diagram showing the differences between a decoder-only auto-regressive model (left) and an encoder-only model (right) (adapted from (Bepler and Berger, 2021). The latter considers the whole context information, producing more meaningful representations. However, it requires a more complex training strategy where parts of the input are masked, and an additional layer reconstructs the missing information.

the real one. Conversely, the second strategy adopts encoder-only layers characterized by a self-attention method where each position in the encoder is connected to all positions of the previous layer (Figure 6, right). This allows embedding the input very efficiently. The model is not able to generate new sequences and therefore training requires different strategies. The most commonly adopted is mask reconstruction, where parts of the training sequences are masked and the model has to exploit the generated embeddings to recognize the missing residues (Vaswani *et al.*, 2017). Encoder-only PLMs are mostly based on a very successful model for Natural Language Processing called Bidirectional Encoder Representations from

Transformers (BERT) (Devlin *et al.*, 2018). The third possible approach, although less common, is based on Text-to-Text models that adopt encoder and decoder layers together (Raffel *et al.*, 2019). Leveraging the best of both methodologies, these models generate high-quality encoder embeddings while retaining some generative abilities (Elnaggar *et al.*, 2020, 2023).

3.4. Training and validating machine-learning models

A critical point in the development of any machine learning-based method is ensuring a fair and unbiased evaluation of its generalization performances (Walsh *et al.*, 2021). To this aim, different strategies can be pursued; the most common is the execution of statistical procedures such as N-fold cross-validation and the adoption of independent blind test sets to evaluate the model generalization performance, simulating the never-seen-before condition.

N-fold cross-validation is a statistical resampling procedure routinely adopted to evaluate the performance of machine-learning models on limited datasets. Moreover, N-fold cross-validation is also used to select the optimal values of the model hyper-parameters and/or to compare different model architectures and input encodings. In general terms, the procedure consists of first partitioning the dataset into N groups or subsets. Therefore, a single subset is retained as testing data and the remaining N-1 subsets are used for training the model. The procedure is repeated N times, using each of the N subsets as a test set exactly once. The number of subsets (N) is set according to the size of the dataset. Routinely, cross-validation is performed by setting N equal to 5 or 10.

In the context of training neural networks, it is useful to further identify, during each cross-validation run, an additional subset referred to as validation set. This set is used, during each training phase, to evaluate the model error rate and stop the training when the error starts

increasing on validation data. This procedure is referred to as early stopping and it is adopted for avoiding overfitting on training data.

A blind test set is also adopted to test the final generalization performance of a machine-learning model. This procedure consists in holding out a significant fraction of the initial dataset (routinely in the range of 10-20% of the available data) and using it to assess the performance of the final model. The remaining fraction (80-90% of the data) is used for model training and selection using N-fold cross-validation. In this way, the never-seen-before condition is ensured, since all the model hyper-parameters are optimized and selected by means of N-fold cross-validation on a fraction of data which is different and independent of the one used for testing the final model.

A graphical representation of a 5-fold cross-validation procedure in combination with a blind test set is shown in Figure 7.

A key issue when performing any data split (either cross-validation or blind test data splits), is ensuring a sufficient level of independence between data used for training and testing the model (Walsh *et al.*, 2021). When dealing with biological sequences, independence can be achieved by ensuring that sequences included in the training data are sufficiently different from those used for testing the model. To this aim, pairwise sequence similarity must be computed among all pairs of sequences included in a dataset and data splits performed consistently, such that no training/testing sequences share sequence similarity above a predefined threshold. Routinely, to ensure proper independence, the sequence similarity threshold is chosen in the range of 25-30% pairwise sequence identity.

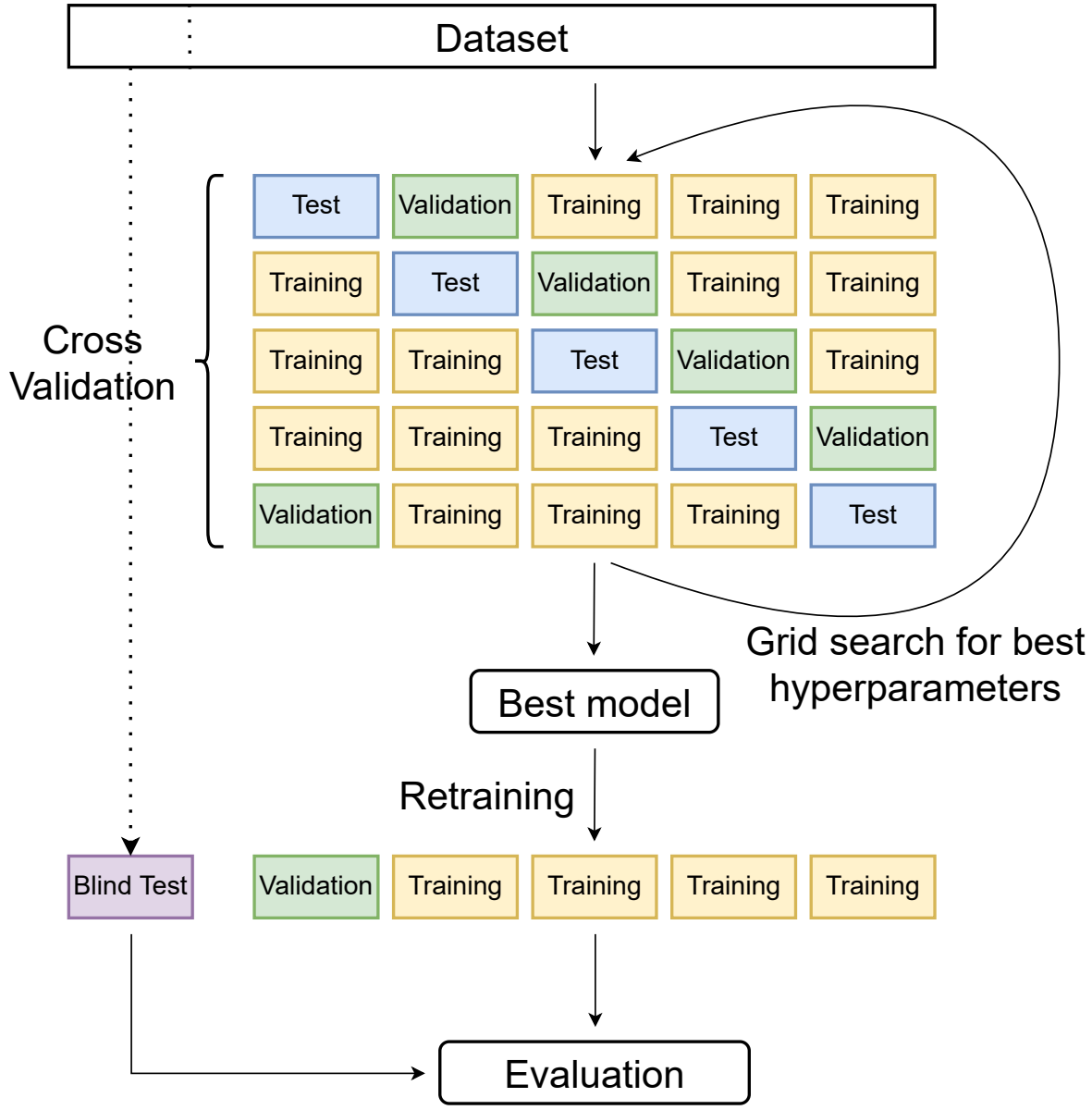


Figure 7. A schematic representation of a 5-fold cross-validation coupled with a blind test set. The blind test set is firstly isolated from the initial dataset, considering about 10-20% of the data. The remaining data (about 80-90%) are used for 5-fold cross-validation. All data splits are performed by ensuring that pairwise sequence similarity between training/testing sequences is below a predefined threshold (25-30% pairwise sequence identity). After performing a grid search to optimize the hyperparameters, the best model is retrained on the full dataset used for cross-validation (80-90% of the data) and tested on the blind test set.

3.5. Scoring indexes

Several scoring indexes can be computed for assessing the quality of an ML-based method. Here, we report a list of the most popular metrics used for the evaluation of classification

algorithms. Equations 3.1 to 3.5 are computed from a confusion matrix, obtained by counting the number of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN).

- Accuracy (Q2):

$$Q2 = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

- Precision:

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

- Recall:

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

- F1 score, the harmonic mean of precision and recall:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3.4}$$

- Area under the receiver operating characteristic curve (ROC-AUC).
- Matthews Correlation Coefficient (MCC):

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{3.5}$$

3.6. Deployment of a new tool

Once a new tool has been developed, it is important to make it accessible to the scientific community. This can be done in several ways, the most common including i) the release of a web server application, ii) the release of the source code and iii) the release of a containerized version of the tool. When creating a web server, it should be made with an intuitive user interface and clear user guides should be available explaining its functionality. Output data should be visualized in a clear and concise form, and made easily available for download and integration into pipelines.

Source code is routinely released in public repositories such as GitHub or GitLab. In this case, special care should be made to list all dependencies and versions of programs needed to run the code, alongside clear instructions on how to install them. Proper usage should be well documented, and scripts for recreating the training procedure as well as for using the method on new data should be available.

Platforms such as Docker or Singularity for the deployment of containerized applications are extremely useful as they allow replication of the same environment requested by the program without additional effort on the user end. Independently of the chosen strategies, it is of the utmost importance to release original datasets used for training, testing and benchmarking, enabling reproducibility for the research community (Walsh *et al.*, 2021).

When deploying a web application, the stability of the server during the years following the release should also be monitored, together with the proper increase in the training data set. Finally, released methods should follow standard guidelines, such as the ones decided by ELIXIR, the European infrastructure for bioinformatics, ensuring FAIRness (FAIR: Findable, Accessible, Interoperable, Reusable) of the tool, and be included in public collections of methods for specific communities, such as Bio.tools.

With all this in mind, I faced the Computational biology problems detailed in Chapter 2 and I successfully developed three predictors which score at the state-of-the-art. All methods have been described in international journals and are available to the community as web servers which are freely accessible at http://www.biocomp.unibo.it/predictors.html. A detailed description of the methods and uses of each predictor is given in the following chapters.

4. DeepREx: Prediction of protein solvent accessibility from sequence

DeepREx (Manfredi *et al.*, 2021) is a machine learning-based method for the prediction of the Accessible Surface Area of residues starting from the protein sequence. It performs a binary classification distinguishing into Buried or Accessible residues, based on a threshold of 20% of Relative Solvent Accessibility. The method, based on a deep architecture mainly composed of Long Short-Term Memory layers, achieves an MCC value of 0.63 when tested on a blind set, reaching the level of the state-of-the-art. DeepREx is freely available as a web server at https://deeprex.biocomp.unibo.it, as a standalone source code at https://github.com/BolognaBiocomp/deeprex or as a Docker container at https://hub.docker.com/r/bolognabiocomp/deeprex.

4.1. Materials and Methods

4.1.1. Datasets

For training and testing DeepREx, we extracted protein chains from the Protein Data Bank (PDB) (Berman *et al.*, 2000) (accessed October 15, 2019) that are obtained through X-ray crystallography at a resolution lesser or equal to 2.5 Å and that are declared by the authors to be functional monomers. We then used SIFTS (Dana *et al.*, 2019) to map all chains to the corresponding UniProt (UniProt Consortium, 2023) sequence and we removed those with a coverage lesser or equal to 70%. We further removed all remaining proteins that are cross-annotated on the Orientations of Proteins in Membranes (OPM) (Lomize *et al.*, 2012) database, so as not to include membrane proteins. Finally, we clustered all proteins using MMseqs2

Table 1. Composition of training and blind test sets used for DeepREx.

Dataset	N. Proteins	N. Residues	N. Buried	N. Exposed
Training set	2,332	636,440	327,118	309,322
Blind test set	200	56,206	29,068	27,138
Total	2,552	692,646	356,186	336,460

(Steinegger and Söding, 2017) with single-linkage clustering, a threshold of 30% of sequence identity and no threshold for coverage, to remove any internal redundancy. Retaining only one sequence per cluster, we ended up with 2,532 monomeric proteins. Amongst those, 200 were randomly selected to create a blind test set, while the remaining 2,332 were randomly split into 10 equally sized subsets for cross-validation.

For more accurate benchmarking, we also created a second blind test set composed of 9 targets from the CASP14 experiment belonging to the free modelling category, meaning that they have no homologous sequences.

For each chain in the dataset, we computed the Solvent Accessible Surface Area (SASA) of each residue with the program DSSP (Touw *et al.*, 2015). Absolute values were then converted into relative ones using the Sander and Rost scale (Rost and Sander, 1994). To perform a binary classification, we adopted a threshold of 20% to differentiate between buried and exposed residues. As shown in Table 1, this allowed us to have a very balanced dataset.

4.1.2. Model Architecture

DeepREx takes in input a protein sequence and classifies each residue as either buried or exposed. A schema of the whole architecture is shown in Figure 8. The first step of the model computes a Multiple Sequence Alignment (MSA) using HHblits version 3 (Steinegger *et al.*, 2019) with two iterations and default parameters, against the Uniclust30 (Mirdita *et al.*, 2017)

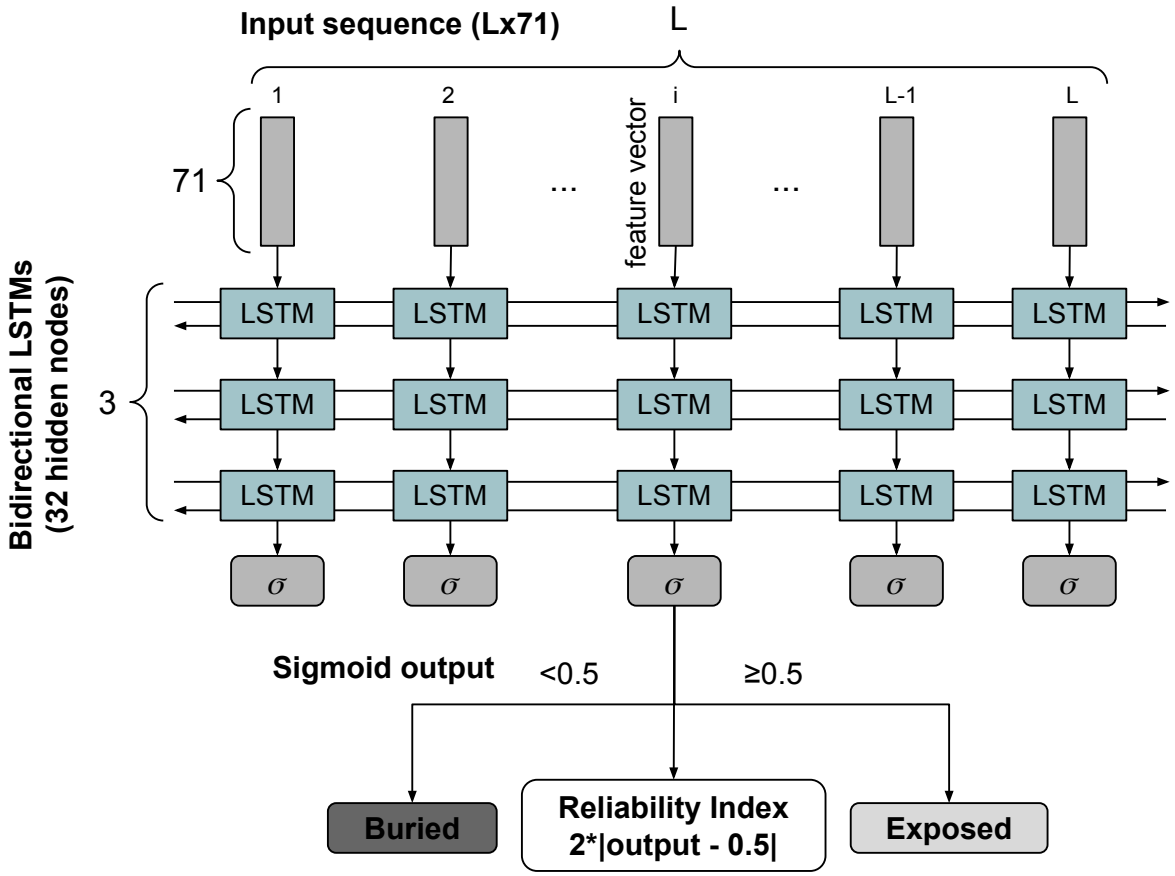


Figure 8. Schema of the architecture adopted for implementing DeepREx. Each residue of the input protein is encoded as a vector of 71 features. The whole sequence is then processed by three cascading Bidirectional Long Short-Term Memory (LSTM) layers, generating an output value for each position. Each output is then converted into a binary classification (Buried if the output is lower than 0.5; Exposed if the output is greater than or equal to 0.5) and a corresponding Reliability Index (Equation 4.1) is computed.

database. Using this MSA, an Lx71 input matrix is generated, where L is the length of the protein. For each residue, the 71 included features are i) 20 values for the canonical one-hot encoding of the residue, ii) 21 values for the sequence profile consisting of the relative frequencies of each residue type, plus the gaps, in the corresponding position, iii) 20 values for the HMM emission probabilities obtained from the match state in the corresponding position, iv) 7 values for the HMM transition probabilities in the corresponding position, and v) 3 values for the Neff_Match, Neff_Insertion and Neff_Deletion scores computed by HHblits.

In the second step, the input matrix is processed by three cascading Bidirectional Long-Short Term Memory (BLSTM) layers with 32 activation units each, followed by a time-distributed fully-connected layer with a sigmoid activation function, producing a real value between 0 and 1 for each residue of the input sequence.

Finally, output values are used for performing a binary classification using a threshold of 0.5 where lower and higher values are respectively associated with the buried and exposed classes. A corresponding reliability index between 0 and 1 is also computed using Equation 4.1.

$$RI = 2 \times |o - 0.5| \tag{4.1}$$

4.2. Results and Discussion

4.2.1. Evaluation and Benchmarking

DeepREx was optimized with 10-fold cross-validation and only the best-scoring model was tested on our blind test set. Results are reported in Table 2, alongside the performances of two state-of-the-art methods for the same task, namely PaleAle5.0 (Kaleel *et al.*, 2019) and NetSurfP-2.0 (Klausen *et al.*, 2019). It is important to mention that while our blind test set was constructed to be non-redundant with respect to our training set, it could share some similarities with the training sets of other tools, leading to a possible overestimation of their performances. Overall, we can observe that the performances of DeepREx do not decrease when tested on a blind test set, meaning that the method is very robust to generalization. The three tested methods have comparable performances on both test sets, with DeepREx having the most balanced results with close values of precision and recall. Remarkably, our method achieves the same levels with a smaller model adopting the lowest number of parameters.

Table 2. Benchmarking of DeepREx.

Dataset	Method	Q2	Precision	Recall	F1	MCC
CrossValidation	DeepREx	0.81	0.82	0.80	0.81	0.62
BlindTest	DeepREx	0.82	0.82	0.80	0.82	0.63
BlindTest	PaleAle5.0[1]	0.82	0.78	0.85	0.82	0.65
BlindTest	NetSurfP2.0[2]	0.83	0.92	0.77	0.82	0.66
CASP14	DeepREx	0.79	0.87	0.76	0.81	0.57
CASP14	PaleAle5.0[1]	0.78	0.90	0.72	0.80	0.58
CASP14	NetSurfP2.0[2]	0.81	0.81	0.89	0.85	0.59

See Section 3.5 for a definition of all the metrics reported in the table. [1] (Kaleel *et al.*, 2019), [2] (Klausen *et al.*, 2019).

4.2.2. DeepREx-WS to assist surface engineering

We decided to build a case study for testing the possible application of the DeepREx web server. In particular, we focused on an example from the field of protein surface charge engineering. In a recent study (Warden *et al.*, 2015), authors were interested in conferring halotolerance to the bovine carbonic anhydrase II (bCAII, UniProtKB: P00921) via the increase of the abundance of acidic residues in the protein surface. Specifically, they studied the available PDB structure (1V9E) to select 18 positions to mutate into negative residues by considering amongst other properties their solvent accessibility and conservation. Our goal was to understand if the use of our web server could help in a hypothetical scenario in which the experimental 3D structure was not available. Most notably, after submission of a query sequence, DeepREx-WS provides, along with the predicted RSA, a set of additional features including i) the Kyte-Doolittle hydropathy score (Kyte and Doolittle, 1982) averaged over a window of five residues, ii) a conservation index computed from the MSA used for the input of DeepREx, iii) the three-class secondary structure as predicted by PYTHIA (Cretin *et al.*, 2021), iv) the five-class flexibility as predicted by MEDUSA (Meersche *et al.*, 2021) and v)

the classification of intrinsically disordered regions as predicted by MobiDB-lite3.0 (Necci *et al.*, 2020).

When looking at the results for bCAII, the first thing we notice is that the predictions achieve a high MCC value of 0.81 and that all of the 18 residues chosen by the authors are correctly classified as exposed with high values of reliability. Moreover, looking only at residues predicted as exposed and excluding Glutamic Acids or Aspartic Acids, we are able to reduce the search space to only 43% of the 260 residues in the protein sequence. This can be further reduced by considering filters based on the computed conservation or the predicted flexibility, both being criteria that are routinely used for this task. In the first case, filtering out residues with a conservation score higher than the median of the protein (0.20) leaves only 78 target exposed and lowly conserved residues (30% of the sequence). Amongst those, five of the 18 positions are left out, two of which have a conservation score slightly above the average (0.22) and three of which are declared by the author to not be lowly conserved, but are nonetheless selected based on other criteria. In the second case, filtering out positions with low flexibility (MEDUSA score lower than 3), only 66 exposed and highly flexible target residues remain (25% of the sequence). In this case, we would exclude six out of 18 positions. Remarkably, none are predicted as rigid.

The aforementioned considerations all point to the fact that DeepREx-WS can be helpful in scenarios where the structure is unknown and the consideration of a set of predicted features can guide our choices.

4.2.3. Linking RSA and pathogenicity of SRV

Knowing the Accessible Surface Area of a residue in a protein sequence can be important for functionally annotating variations occurring at that position. Despite this, RSA is rarely

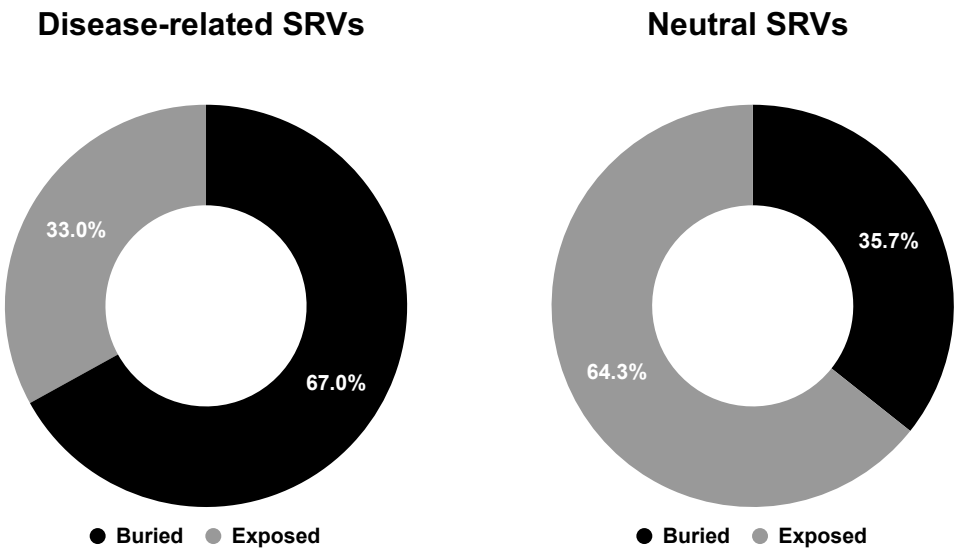


Figure 9. Fraction of disease-related and neutral variations that are classified as Buried or Exposed using a 20% threshold of Relative Surface Accessibility as computed on HVAR3D-2.0.

adopted amongst features used to encode input proteins in methods predicting the effect of Single Residue Variations (SRVs) (Chen and Zhou, 2005; Martelli *et al.*, 2016; Savojardo *et al.*, 2019). Following the development of DeepREx, we investigated the correlation between the pathogenicity of an SRV and the exposure of the residue undergoing variation.

For this purpose, we curated two datasets. The first, referred to as HVARSEQ, was obtained by collecting all SRVs from HUMSAVAR (UniProt Consortium, 2023) (accessed August 2020) and filtering out all variations labelled as "Unclassified", as well as all disease-related SRVs not associated with diseases included in OMIM (McKusick, 1998). In total, we included 69,385 variations mapped on 12,494 distinct protein sequences, 29,949 of which are disease-related and 39,436 are neutral. The second, referred to as HVAR3D-2.0, is a subsect of HVARSEQ including variations mapped on proteins with a PDB structure that i) was experimentally resolved with X-ray crystallography, ii) has a resolution lower or equal to 3Å and iii) has a coverage of the corresponding UniProt sequence of at least 70%. This subset accounts for 10,760 SRVs mapped on 1,255 unique proteins, 6,778 of which are disease-related

and 3,982 are neutral. For each SRV included in HVAR3D-2.0, we first divided residues into Buried and Exposed classes following the same procedure described in Section 4.1.1. Figure 9 shows that disease-related variations have a tendency to occur in positions buried in the structure. Conversely, neutral variations tend to be located on the surface of the protein. After that, for each of the 20 residue types, we decided to analyze how their probability of leading to the onset of diseases when variated would vary when the residue is known to be either buried or exposed. In order to do so, we computed the following conditional probabilities:

- The number of residues that are disease-related when variated over the total number of residues in the dataset is the conditional probability of a residue to be disease-related when variated

$$P_D = \frac{n_D}{N} \tag{4.2}$$

- The number of residues of type R that are disease-related when variated over the total number of residues of type R in the dataset is the conditional probability of a residue of type R to be disease-related when variated

$$P_{D|R} = \frac{n_{DR}}{n_R} \tag{4.3}$$

- The number of buried residues of type R that are disease-related when variated over the total number of buried residues of type R in the dataset is the conditional probability of a residue of type R to be disease-related when variated given that it is buried

$$P_{D|B,R} = \frac{n_{DBR}}{n_{BR}} \tag{4.4}$$

- The number of exposed residues of type R that are disease-related when variated over the total number of exposed residues of type R in the dataset is the conditional probability of a residue of type R to be disease-related when variated given that it is exposed

$$P_{D|E,R} = \frac{n_{DER}}{n_{ER}} \tag{4.5}$$

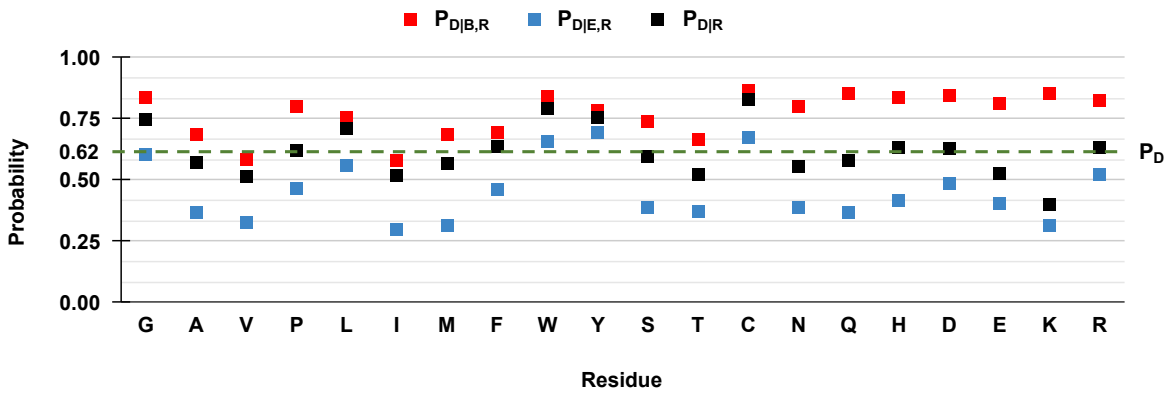


Figure 10. Conditional probabilities of the 20 residue types to be disease-related when variated as computed on HVAR3D-2.0. P_D, $P_{D|R}$, $P_{D|B,R}$ and $P_{D|E,R}$ are all defined in Section 4.2.3.

Figure 10 shows that, although different residue types have different propensities, it is always the case that knowing that a residue is buried increases its a priori probability to be associated with a disease. Conversely, knowing that a residue is exposed lowers it. This difference appears to be particularly marked for asparagine (N), glutamine (Q), histidine (H), and lysine (K), all residues that are polar and that in our dataset are abundant on the protein surface. We also observe that, independently of their exposure status, three residue types [tryptophan (W), tyrosine (Y) and cysteine (C)] have a probability to be disease-related when variated higher than the baseline, while two [valine (V) and isoleucine (I)] are lower.

Similarly, we computed the same statistics on the extended dataset HVARSEQ after running DeepREx on all of its sequences, considering putative classifications into Buried or Exposed residues. Results are reported in Figures 11 and 12. Interestingly, these findings agree with data computed on the structural dataset, showing an abundance of buried residues for disease-related SRVs and of exposed residues for neutral SRVs. When looking at individual conditional probabilities, the relation $P_{D|B,R} > P_{D|R} > P_{D|E,R}$ still holds for all 20 residue types. The main observable difference is relative to tryptophan (W), tyrosine (Y) and cysteine (C), which now present values of $P_{D|E,R}$ lower than the baseline. However, this is most likely due to prediction errors given their low abundance in the datasets.

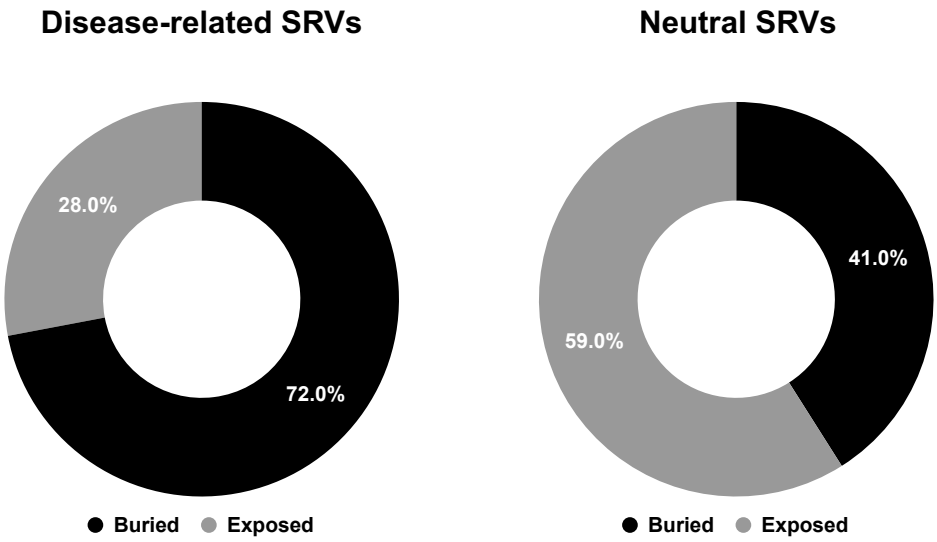


Figure 11. Fraction of disease-related and neutral variations that are classified as Buried or Exposed using a 20% threshold of Relative Surface Accessibility as computed on HVARSEQ.

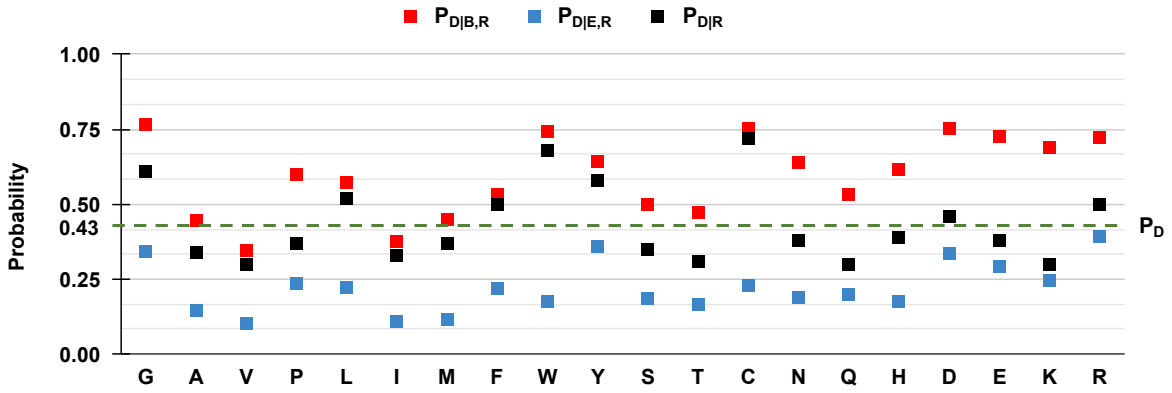


Figure 12. Conditional probabilities of the 20 residue types to be disease-related when variated as computed on HVARSEQ. P_D, $P_{D|R}$, $P_{D|B,R}$ and $P_{D|E,R}$ are all defined in Section 4.2.3.

Overall, these findings show a remarkable correlation between the pathogenicity of an SRV and its exposure and they indicate that accurate predicting tools like DeepREx can be adopted for large-scale analysis without the limitation of using only proteins with an experimentally resolved structure (Savojardo, Manfredi, *et al.*, 2020).

5. E-SNPs&GO: Prediction of variant pathogenicity

E-SNPs&GO (Manfredi *et al.*, 2022) is a machine learning-based method for the prediction of the Pathogenicity of Single Residue Variations starting from the protein sequence. It performs a binary classification distinguishing into Pathogenic/Likely Pathogenic or Benign/Likely Benign variations. The method, based on a Support Vector Machine and adopting Protein Language Models for embedding the input, achieves an MCC of 0.86 when tested on a blind set, reaching the level of the state-of-the-art. E-SNPs&GO is freely available as a web server at https://esnpsandgo.biocomp.unibo.it.

5.1. Materials and Methods

5.1.1. Datasets

For training and testing E-SNPs&GO, we extracted Single Residue Variations from two sources, HUMSAVAR (UniProt Consortium, 2023) (accessed on August 4, 2021) and ClinVar (Landrum *et al.*, 2018) (accessed on March 29, 2021). We then removed all Pathogenic/Likely Pathogenic (P/LP) variations that were not clearly associated with diseases catalogued in either OMIM (McKusick, 1998) or MONDO (Shefchek *et al.*, 2020), as well as all variations that were either of Uncertain Significance (US), somatic or with contrasting annotations in the two source databases. This resulted in 111,412 SRVs annotated on 13,661 unique protein sequences. To avoid biases during both training and benchmarking, we then clustered all proteins using MMseqs2 (Steinegger and Söding, 2017) with connected-component clustering and a threshold of 25% of sequence identity over an alignment coverage of at least 40%. We then generated 11 equally sized subsets, making sure that for every pair of proteins taken from two different subsets they belong to different clusters, thus reducing any cross-redundancy.

Table 3. Composition of training and blind test sets used for E-SNPs&GO.

Dataset	Proteins	N. SRV	N. P/LP SRV	N. B/LB SRV
Training set	12,347	101,146	39,812	61,334
Blind test set	1,314	10,266	4,083	6,183
Total	13,661	111,412	43,895	67,517

SRV: Single Residue Variations.

P/LP: Pathogenic/Likely Pathogenic, B/LB: Benign/Likely Benign.

One of the subsets was then randomly selected to be used as a blind test set for benchmarking purposes, while the other 10 were used in cross-validation.

Because the dataset is slightly unbalanced in favour of Benign/Likely Benign (B/LB) variations (data shown in Table 3), we also made sure that the sampling of the subsets would keep the same proportion of P/LP to B/LB variations in each subset.

Finally, for each protein in the dataset, we also extracted Gene Ontology (Ashburner *et al.*, 2000) terms annotated in the corresponding UniProt entry, including a total of 17,076 GO annotations divided into 11,476 Biological Process (BP), 3,955 Molecular Function (MF) and 1,645 Cellular Component (CC).

5.1.2. Model Architecture

E-SNPs&GO takes in input a protein sequence and an SRV mapped onto it and classifies it as either P/LP or B/LB. Figure 13 shows a schema of the whole architecture.

The first step is input encoding. Here, the protein is embedded using two different models, ESM-1v (Meier *et al.*, 2021) and ProtT5 (Elnaggar *et al.*, 2020), generating respectively 1,280 and 1,024 features for each residue. The same procedure is done with the variant sequence, generated by substituting the variation in the corresponding position. Most notably, because

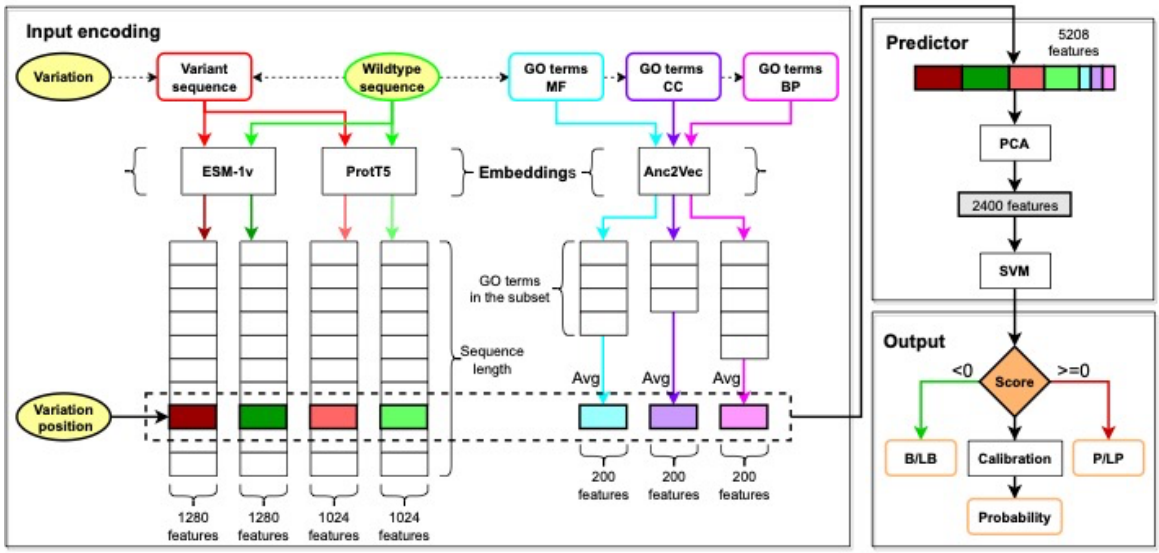


Figure 13. Schema of the architecture adopted for implementing E-SNPs&GO. In the Input encoding phase, both the variant and the wildtype sequences are embedded using ESM-1v and ProtT5 and the vectors encoding the variated residue are extracted (red and green boxes). GO annotations are extracted for the target protein and Anc2Vec is used to obtain a vectorial representation of each term. Terms belonging to the same subontology (MF = Molecular Function, CC = Cellular Component, BP = Biological Process) are averaged together (blue, purple and pink boxes). In the Predictor phase, the final encoding consisting of 5208 features is processed by a Principal Component Analysis (PCA) and a Support Vector Machine (SVM). In the Output phase, the score computed by the SVM is transformed into a binary prediction (Benign/Likely Benign if the output is lower than 0; Pathogenic/Likely Pathogenic if the output is greater than or equal to 0) and an Isotonic Regression is adopted to compute a calibrated pathogenicity probability.

protein embedding models are context-dependent, we can expect the encoding of all residues to change with respect to the original sequence. Subsequently, all GO annotations are extracted for the input protein. Each term is then embedded with Anc2Vec (Edera *et al.*, 2022), a model trained to generate a vectorial embedding for each term in the Gene Ontology by taking into account their ancestry in the tree-like structure of the ontology. This generates corresponding vectors of 200 features that, for each of the three existing sub-ontologies, are then averaged. Finally, we concatenate the 4 vectors (2 models times 2 sequences) corresponding to the variated residue with the 3 vectors obtained from the GO terms to generate an input vector of 1280x2+1024x2+200x3=5208 features.

In the second step, a Principal Component Analysis (PCA) is first applied to the input vector to reduce its dimensionality from 5208 to 2400. This is then processed by a Support Vector Machine (SVM) classifier generating a real value.

Finally, the output value is used for performing a binary classification using a threshold of 0, where lower and higher values are respectively associated with the B/LB (negative) and P/LP (positive) classes. An Isotonic Regression (Niculescu-Mizil and Caruana, 2005) is also used to compute a calibrated pathogenicity probability, which we can use to obtain an integer reliability index between 1 and 10 using Equation 5.1.

$$RI = round\left(20 \times \left|P_{P/LP} - 0.5\right|\right) \quad (5.1)$$

5.2. Results and Discussion

5.2.1. Evaluation and Benchmarking

DeepREx was optimized in a 10-fold cross-validation and only the best-scoring model was tested on our blind test set. Results are reported in Table 4, alongside the performances of other state-of-the-art methods for the same task. In particular, we confront ourselves with SNPs&GO (Calabrese *et al.*, 2009), a previous version of this method, with three tools widely used in the literature, namely PROVEAN (Choi *et al.*, 2012), SIFT (Ng and Henikoff, 2001) and PolyPhen-2 (Adzhubei *et al.*, 2010), and with the best-performing method to date, MutPred2.0 (Pejaver *et al.*, 2020). It is important to mention that while our blind test set was constructed to be non-redundant with respect to our training set, it could share some similarities with the training sets of other tools, leading to a possible overestimation of their performances.

Overall, we can see that E-SNPs&GO and MutPred2.0 have comparable results, while both performing notably better than the other methods proposed for the benchmark. The two methods also appear to be complementary, as the first has a much higher precision and the second has a much higher recall.

Table 4. Benchmarking of E-SNPs&GO.

Dataset	Method	Q2	Precision	Recall	F1	MCC	AUC
CrossValidation	E-SNPs&GO	0.85	0.82	0.79	0.81	0.69	0.84
BlindTest	E-SNPs&GO	0.87	0.86	0.80	0.83	0.72	0.86
BlindTest	SNPs&GO[1]	0.80	0.85	0.63	0.72	0.58	0.78
BlindTest	MutPred2.0[2]	0.86	0.79	0.88	0.83	0.71	0.86
BlindTest	PROVEAN[3]	0.78	0.69	0.83	0.75	0.57	0.79
BlindTest	SIFT[4]	0.74	0.63	0.88	0.73	0.53	0.77
BlindTest	PolyPhen-2[5]	0.72	0.61	0.90	0.72	0.50	0.75

See Section 3.5 for a definition of all the metrics reported in the table.

[1] (Calabrese *et al.*, 2009), [2] (Pejaver *et al.*, 2020), [3] (Choi *et al.*, 2012), [4] (Ng and Henikoff, 2001), [5] (Adzhubei *et al.*, 2010).

The main advantage of our tool is however the use of protein embedding models for encoding the input, which allows us to avoid the costly computation of MSAs, making E-SNPs&GO much faster than any other proposed method.

With E-SNPs&GO, we first introduce the idea of adopting two different models, ProtT5 (Elnaggar *et al.*, 2020) and ESM-1v (Meier *et al.*, 2021), for embedding input proteins. Table 5 reports the results of an ablation study confronting the performances of the final method with an identical architecture that utilizes only one or the other PLM to embed the input. We argue that this effect could be due to the two different PLMs learning complementary features that our model learns to pick up and combine. Most notably, thanks to the general efficiency of PLMs, using two for constructing the input representation has a negligible effect on the time efficiency.

Table 5. Comparison of performances for E-SNPs&GO when using two different embedding methods vs using their combination.

Method	Input	Q2	Precision	Recall	F1	MCC	AUC
E-SNPs&GO	ESM-1v[1]	0.83	0.82	0.78	0.80	0.66	0.83
E-SNPs&GO	ProtT5[2]	0.84	0.82	0.79	0.81	0.67	0.83
E-SNPs&GO	ESM-1v[1] + ProtT5[2]	0.85	0.82	0.79	0.81	0.69	0.84

See Section 3.5 for a definition of all the metrics reported in the table.

[1] (Elnaggar *et al.*, 2020), [2] (Meier *et al.*, 2021).

5.2.2. Predictions on Variations of Uncertain Significance

For further evaluation of our method, we decided to classify a dataset of 9,165 Variations of Uncertain Significance (VUS) extracted from HUMSAVAR (accessed on May 12, 2022). Since their possible correlation to the onset of diseases is not experimentally annotated, we cannot verify the correctness of our predictions. Nonetheless, thanks to the calibration procedure we adopt when computing the pathogenicity probabilities, we can have a fair estimate (Benevenuta *et al.*, 2021). Figure 14 shows that 67% of all VUS are predicted with a Reliability Index greater or equal to 6. Assuming that the corresponding probabilities are correct (e.g. a VUS with a predicted pathogenicity probability of 0.8 is a true positive 80% of the time and a false positive 20% of the time), we can compute an estimated MCC of 0.67 and an estimated Q2 of 0.85. These results are in line with the performances computed on the blind test set and show that E-SNPs&GO has a good level of confidence even when predicting variants that are difficult to annotate.

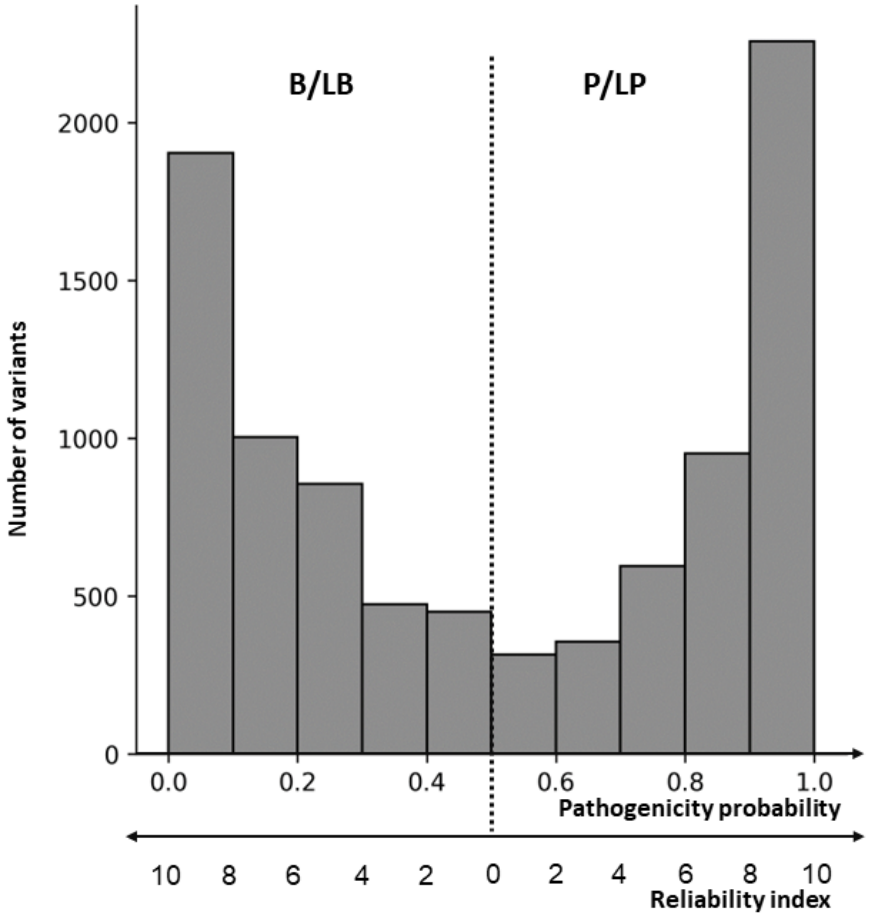


Figure 14. Distribution of predicted pathogenicity probabilities and corresponding Reliability Indices for the dataset of Variants of Uncertain Significance.

6. ISPRED-SEQ: Prediction of Protein-Protein Interaction sites

ISPRED-SEQ (Manfredi *et al.*, 2023) is a machine learning-based method for the prediction of Interaction Sites for non-partner-specific Protein-Protein Binding starting from the protein sequence. It performs a binary classification for each residue in the target chain marking putative interaction sites. The method, based on a deep architecture adopting a Convolutional Layer and Protein Language Models for embedding the input, achieves an MCC of 0.38 when tested on a blind set, surpassing the next best-performing method by 6 percentage points. ISPRED-SEQ is freely available as a web server at https://ispredws.biocomp.unibo.it.

6.1. Materials and Methods

6.1.1. Datasets

For training and testing ISPRED-SEQ, we extracted datasets from the literature. For the training set, we adopted the same one used by DELPHI (Li *et al.*, 2021), composed of 9,982 protein sequences filtered from a previous study (J. Zhang *et al.*, 2019). All sequences are guaranteed by the authors to have less than 25% of sequence identity with one another, as well as with all of the test sets. For our purposes, we further decided to curate the dataset by removing all proteins with less than 80% coverage of the corresponding PDB (Berman *et al.*, 2000) structures, ending up with 6,066 sequences.

For benchmarking, we also adopted four datasets widely used in the literature. The first one, referred to as Dset448, comprises 448 sequences used for comparing state-of-the-art methods for the prediction of interaction sites from the sequence. The second one, referred to as Dset335,

Table 6. Composition of training and blind test sets used for ISPRED-SEQ.

Dataset	Proteins	Residues	Interaction Sites	Non-Interaction Sites
Training set	6,066	1,757,296	285,751	1,471,545
Dset448	448	116,500	15,810	100,690
Dset335	355	95,940	11,467	84,473
Homo_TE	95	24,766	5,564	19,202
Hetero_TE	48	14,056	1,313	12,743

is a subset of the first one obtained by removing all proteins with sequence identity higher than 25% against the training set of DLPred (B. Zhang *et al.*, 2019). The other two, namely HomoTE and HeteroTE, include respectively 479 and 48 protein chains from homomeric and heteromeric complexes. We use these to compare our method with the performance of PIPENN (Stringer *et al.*, 2022), as well as to check our ability to make predictions for these two different kinds of protein complexes. Table 6 reports comprehensive details on all the datasets adopted.

6.1.2. Model Architecture

ISPRED-SEQ takes in input a protein sequence and classifies each residue as either being or not being an Interaction Site (IS) for protein-protein binding. Figure 15 shows a schema of the whole architecture.

The first step is input encoding. Here, the protein is embedded using two different models, ESM-1b (Rives *et al.*, 2021) and ProtT5 (Elnaggar *et al.*, 2020), generating respectively 1,280 and 1,024 features for each residue. We then consider as our input a matrix of size 31x2304 by taking the concatenated vectors in a padded window of 31 residues centred on every target residue.

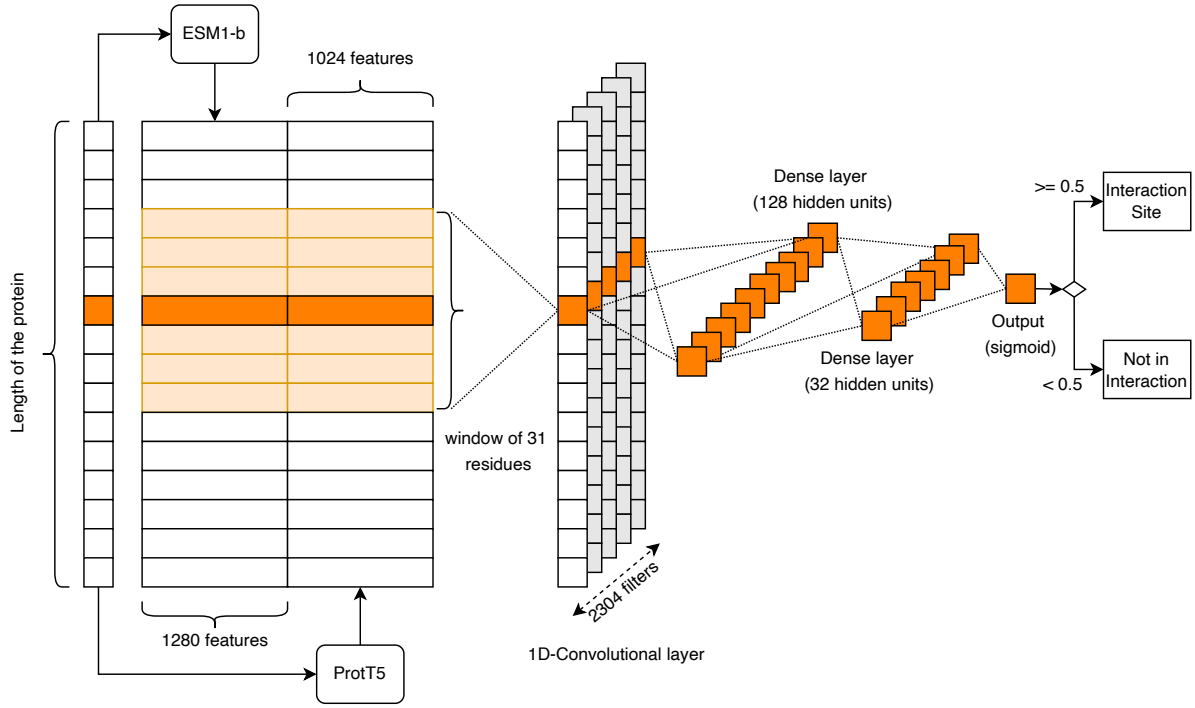


Figure 15. Schema of the architecture adopted for implementing ISPRED-SEQ. Each residue in the protein sequence is embedded using ESM1-b and ProtT5, producing a vectorial representation of 2304 features. For each position, a window of 31 residues is processed by a 1-D Convolutional layer and by three cascading fully connected layers. The output is then transformed into a binary classification (interaction site if the output is greater than or equal to 0.5, not interaction site if the output is lower than 0.5).

In the second step, each input matrix is processed by a 1D Convolutional Layer with 2304 filters and a filter width of 31, used to condense the window information over every feature into a single vector of length 2304. This is followed by two hidden fully-connected layers with 128 and 32 activation units, and by a fully-connected layer using a sigmoid function to provide in output a real value between 0 and 1.

Finally, output values are used for performing a binary classification using a threshold of 0.5 where higher values are associated with ISs.

6.2. Results and Discussion

6.2.1. Evaluation and Benchmarking

ISPRED-SEQ was evaluated on four different test sets to guarantee a fair comparison with several state-of-the-art methods. In particular, we include in our benchmarking PITHIA (Hosseini and Ilie, 2022) (Dset448 and Dset335), DELPHI (Li *et al.*, 2021) (Dset448 and Dset335), PIPENN (Stringer *et al.*, 2022) (Dset448), SCRIBER (Zhang and Kurgan, 2019) (Dset448 and Dset335), SSWRF (Wei *et al.*, 2016) (Dset448), CRFPPI (Wei *et al.*, 2015) (Dset448), LORIS (Dhole *et al.*, 2014) (Dset448), DLPred (B. Zhang *et al.*, 2019) (Dset335) and PIPENN (Stringer *et al.*, 2022) (Homo_TE and Hetero_TE). All methods, including ours, provide as output a numerical score and adopt a threshold to discriminate residues that are interaction sites for protein-protein interactions. As described in (Zhang and Kurgan, 2018), for this task performances are routinely computed by selecting a threshold such that the number of positive predictions (FP+TP) is equal to the number of real positive examples (TP+FN), from which follows that the number of residues incorrectly classified as interaction sites (FP) is equal to the number of real interaction sites incorrectly classified (FN). For the sake of a fair comparison, we decided to apply the same strategy, but we also report performances obtained by using the standard threshold of 0.5 that would be used when running predictions on the web server.

Regardless of this choice, Table 7 shows that ISPRED-SEQ outperforms all state-of-the-art methods currently present in the literature by several percentage points for all metrics considered. It is worth mentioning that when adopting the standard threshold, the recall of our method is much higher than its precision, meaning that while we tend to overpredict residues as putative interaction sites, we miss very few of the real ones despite their low relative abundance.

Table 7. Benchmarking of ISPRED-SEQ.

Dataset	Method	Q2	Precision	Recall	F1	MCC	AUC
Dset448	ISPRED-SEQ (th=0.5)	0.71	0.29	0.78	0.42	0.34	0.82
Dset448	ISPRED-SEQ (th: FP=FN)	0.86	0.47	0.47	0.47	0.38	0.82
Dset448	PITHIA[1]	0.84	0.41	0.41	0.41	0.32	0.78
Dset448	DELPHI[2]	0.83	0.37	0.37	0.37	0.27	0.74
Dset448	PIPENN[3]	0.79	0.39	0.39	0.39	0.25	0.72
Dset448	SCRIBER[4]	0.82	0.29	0.29	0.29	0.23	0.72
Dset448	SSWRF[5]	0.81	0.29	0.29	0.29	0.18	0.69
Dset448	CRFPPI[6]	0.81	0.27	0.27	0.27	0.15	0.68
Dset448	LORIS[7]	0.81	0.26	0.26	0.26	0.15	0.66
Dset335	ISPRED-SEQ (th=0.5)	0.72	0.27	0.77	0.40	0.33	0.82
Dset335	ISPRED-SEQ (th: FP=FN)	0.87	0.46	0.46	0.46	0.39	0.82
Dset335	PITHIA[1]	0.85	0.38	0.38	0.38	0.30	0.76
Dset335	DELPHI[2]	0.85	0.36	0.36	0.36	0.28	0.75
Dset335	SCRIBER[4]	0.84	0.32	0.32	0.32	0.23	0.72
Dset335	DLPred[8]	0.84	0.31	0.31	0.31	0.21	0.72
Homo_TE	ISPRED-SEQ (th=0.5)	0.71	0.42	0.83	0.56	0.42	0.84
Homo_TE	ISPRED-SEQ (th: FP=FN)	0.81	0.58	0.58	0.58	0.46	0.84
Homo_TE	PIPENN[3]	0.77	0.49	0.49	0.49	0.34	0.77
Hetero_TE	ISPRED-SEQ (th=0.5)	0.65	0.17	0.68	0.27	0.20	0.72
Hetero_TE	ISPRED-SEQ (th: FP=FN)	0.86	0.24	0.24	0.24	0.16	0.72
Hetero_TE	PIPENN[3]	0.85	0.29	0.29	0.29	0.11	0.66

See Section 3.5 for a definition of all the metrics reported in the table.

[1] (Hosseini and Ilie, 2022), [2] (Li *et al.*, 2021), [3] (Stringer *et al.*, 2022), [4] (Zhang and Kurgan, 2019), [5] (Wei *et al.*, 2016), [6] (Wei *et al.*, 2015), [7] (Dhole *et al.*, 2014), [8] (B. Zhang *et al.*, 2019).

Table 8. Comparison of performances for ISPRED-SEQ when using two different embedding methods vs using their combination.

Method	Input	Q2	Precision	Recall	F1	MCC	AUC
ISPRED-SEQ	ESM-1b[1]	0.68	0.30	0.70	0.42	0.30	0.74
ISPRED-SEQ	ProtT5[2]	0.69	0.31	0.72	0.43	0.31	0.75
ISPRED-SEQ	ESM-1b[1] + ProtT5[2]	0.70	0.32	0.75	0.45	0.34	0.80

See Section 3.5 for a definition of all the metrics reported in the table.

[1] (Rives *et al.*, 2021), [2] (Elnaggar *et al.*, 2020).

With ISPRED-SEQ we confirm what was observed during the development of E-SNPs&GO. As reported in Table 8, performances obtained using two different and complementary PLMs surpass identical networks adopting only one or the other to embed the input. Moreover, we argue that the use of both models does not impact the time efficiency of ISPRED-SEQ, especially when compared to other tools adopting traditional MSA-based input features.

7. Conclusions and perspectives

During the three years of my PhD, I developed three novel machine learning-based methods addressing important problems in the field of Computational Biology: i) the prediction of residue Accessible Solvent Area, ii) the prediction of the pathogenicity of Single Residue Variations and iii) the prediction of protein-protein interaction sites. All tools perform at the level of the state-of-the-art and are currently published as well as available to the scientific community in the form of web servers. During my research activities, I focused on the following: curation of datasets for fair training and evaluation of machine learning architectures, avoiding all possible biases; standardisation of procedures for optimizing the models while preserving their ability to transfer extracted knowledge to new data; developing of new techniques for efficiently embedding proteins into vectorial encoding suited for downstream predictive tasks, avoiding the need to construct costly canonical Position Scoring Specific Matrices (PSSMs) from Multiple Sequence Alignments (MSAs). I also investigated pathogenic Single Residue Variations (RSA) in relation to their Relative Solvent Accessibility (RSA) and to the possibility of being discriminated from benign ones.

Research always opens new problems. For this reason, in my publications, I highlight the necessity of exploring other possible PLMs applications. During my internship abroad, I deepened my knowledge of training procedures and best practices to adopt when exploiting protein embeddings.

The relationship between genetic variations and human diseases requires the integration of many sources of information to fill the gap between molecules and systemic behaviour. It is therefore desirable to include data related to different levels of complexity in order to generate a platform for structural and functional characterization of the relationship between variations and human diseases. This will require the curation of high-quality datasets of relations between

protein variations, diseases, and other phenotypic features, as well as the development of accurate predicting tools.

Finally, for all topics here discussed, it will be important to investigate the potential contribution of three-dimensional models generated by AlphaFold. When the predictive target can be directly computed from the structure like in the case of solvent accessibility, models with high enough confidence scores could be directly adopted to increase the amount of available training data. In doing so, methods such as DeepREx could become much more accurate and generate better estimates for proteins where AlphaFold cannot compute high-quality models. Other tasks like the prediction of interacting sites cannot be solved simply by knowing the structure of the protein. Nonetheless, it will be interesting to understand to which extent the combination of AlphaFold models and methods taking them as input is preferable to methods like ISPRED-SEQ adopting only sequence-based features.

Acknowledgments

The work presented here has been possible only thanks to the effort of many people who significantly helped me over the years. I would like to spend a few words to acknowledge their efforts.

Thanks, Prof. Rita Casadio, Prof. Pier Luigi Martelli and Prof. Castrense Savojardo, for everything you taught me and the opportunities you gave me. Thanks, Prof. Burkhard Rost, for showing me an exciting field of research and for having me in your laboratory. Thanks, Prof. Allegra Via, for mentoring me in my teaching experiences.

Thanks to all my colleagues in Bologna and Munich, especially Giovanni Madeo, Tobias Olenyi and Gabriele Vazzana, for the material help and, most importantly, for the fantastic working days we spent together.

Thanks to my Family, Elisabetta Giacomucci, Alessandro Manfredi, Stefano Ferrari, Elisa Torreggiani and Vikas Lodato, for the continuous and unconditional support you gave me during my whole life.

Lastly, thank you, Claudia Pasquali, for being my partner, for the path we are shaping together, and for always being at my side.

References

Abdel-Hamid,O. *et al.* (2014) Convolutional Neural Networks for Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, **22**, 1533–1545.

Adzhubei,I.A. *et al.* (2010) A method and server for predicting damaging missense mutations. *Nat. Methods*, **7**, 248–249.

Alley,E.C. *et al.* (2019) Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods*, **16**, 1315–1322.

Anfinsen,C.B. (1973) Principles that govern the folding of protein chains. *Science*, **181**, 223–230.

Asgari,E. and Mofrad,M.R.K. (2015) Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics. *PLoS One*, **10**, e0141287.

Ashburner,M. *et al.* (2000) Gene Ontology: tool for the unification of biology. *Nat. Genet.*, **25**, 25–29.

Ausaf Ali,S. *et al.* (2014) A Review of Methods Available to Estimate Solvent-Accessible Surface Areas of Soluble Proteins in the Folded and Unfolded States. *Curr. Protein Pept. Sci.*, **15**, 456–476.

Baldi,P. (2021) Deep Learning in Science.

Benevenuta,S. *et al.* (2021) Calibrating variant-scoring methods for clinical decision making. *Bioinformatics*, **36**, 5709–5711.

Bepler,T. and Berger,B. (2021) Learning the protein language: Evolution, structure, and function. *Cell Syst*, **12**, 654–669.e3.

Berman,H.M. *et al.* (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.

Bishop,C.M. (2006) Pattern Recognition and Machine Learning Springer New York.

Boser,B.E. *et al.* (1992) A training algorithm for optimal margin classifiers. In, *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92. Association for Computing Machinery, New York, NY, USA, pp. 144–152.

Bystroff,C. and Krogh,A. (2008) Hidden Markov Models for prediction of protein features. *Methods Mol. Biol.*, **413**, 173–198.

Calabrese,R. *et al.* (2009) Functional annotations improve the predictive score of human disease-related mutations in proteins. *Hum. Mutat.*, **30**, 1237–1244.

Carter,H. *et al.* (2013) Identifying Mendelian disease genes with the variant effect scoring tool. *BMC Genomics*, **14 Suppl 3**, S3.

Casadio,R. *et al.* (2022) Machine learning solutions for predicting protein–protein interactions. *Wiley Interdiscip. Rev. Comput. Mol. Sci.*, **12**.

Chen,H. and Zhou,H.-X. (2005) Prediction of solvent accessibility and sites of deleterious mutations from protein sequence. *Nucleic Acids Res.*, **33**, 3193–3199.

Choi,Y. *et al.* (2012) Predicting the functional effect of amino acid substitutions and indels. *PLoS One*, **7**, e46688.

Cortes,C. and Vapnik,V. (1995) Support-vector networks. *Mach. Learn.*, **20**, 273–297.

Cretin,G. *et al.* (2021) PYTHIA: Deep Learning Approach for Local Protein Conformation Prediction. *International Journal of Molecular Sciences*, **22**, 8831.

Dana,J.M. *et al.* (2019) SIFTS: updated Structure Integration with Function, Taxonomy and Sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins. *Nucleic Acids Res.*, **47**, D482–D489.

Deng,L. *et al.* (2017) A sparse autoencoder-based deep neural network for protein solvent accessibility and contact number prediction. *BMC Bioinformatics*, **18**, 569.

Devlin,J. *et al.* (2018) BERT: Pre-training of Deep Bidirectional Transformers for Language

Understanding. *arXiv [cs.CL]*.
Dhole,K. *et al.* (2014) Sequence-based prediction of protein–protein interaction sites with L1-logreg classifier. *J. Theor. Biol.*, **348**, 47–54.
Dong,Z. *et al.* (2014) CRF-based models of protein surfaces improve protein-protein interaction site predictions. *BMC Bioinformatics*, **15**, 277.
Drozdetskiy,A. *et al.* (2015) JPred4: a protein secondary structure prediction server. *Nucleic Acids Res.*, **43**, W389–94.
Durbin,R. *et al.* (1998) Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids Cambridge University Press.
Edera,A.A. *et al.* (2022) Anc2vec: embedding gene ontology terms by preserving ancestors relationships. *Briefings in Bioinformatics*, **23**.
Elnaggar,A. *et al.* (2023) Ankh: Optimized Protein Language Model Unlocks General-Purpose Modelling. *arXiv [cs.LG]*.
Elnaggar,A. *et al.* (2020) ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. *arXiv [cs.LG]*.
Esquivel,R.O. *et al.* (2013) Decoding the building blocks of life from the perspective of quantum information. In, *Advances in Quantum Mechanics*. IntechOpen.
Fan,C. *et al.* (2016) PredRSA: a gradient boosted regression trees approach for predicting protein solvent accessibility. *BMC Bioinformatics*, **17 Suppl 1**, 8.
Goodfellow,I. *et al.* (2016) Deep Learning MIT Press.
Hanson,J. *et al.* (2019) Improving prediction of protein secondary structure, backbone angles, solvent accessibility and contact numbers by using predicted contact maps and an ensemble of recurrent and residual convolutional neural networks. *Bioinformatics*, **35**, 2403–2410.
Hastie,T. *et al.* (2009) The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition Springer Science & Business Media.
Heinzinger,M. *et al.* (2019) Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics*, **20**, 723.
Hinton,G. and Sejnowski,T.J. (1999) Unsupervised Learning: Foundations of Neural Computation MIT Press.
Hochreiter,S. and Schmidhuber,J. (1997) Long short-term memory. *Neural Comput.*, **9**, 1735–1780.
Hornik,K. *et al.* (1989) Multilayer feedforward networks are universal approximators. *Neural Netw.*, **2**, 359–366.
Hosseini,S. and Ilie,L. (2022) PITHIA: Protein Interaction Site Prediction Using Multiple Sequence Alignments and Attention. *Int. J. Mol. Sci.*, **23**.
Ho,T.K. (1995) Random decision forests. In, *Proceedings of 3rd International Conference on Document Analysis and Recognition*., pp. 278–282 vol.1.
Jagadeesh,K.A. *et al.* (2016) M-CAP eliminates a majority of variants of uncertain significance in clinical exomes at high sensitivity. *Nat. Genet.*, **48**, 1581–1586.
Jumper,J. *et al.* (2021) Highly accurate protein structure prediction with AlphaFold. *Nature*, **596**, 583–589.
Kaelbling,L.P. *et al.* (1996) Reinforcement Learning: A Survey. *jair*, **4**, 237–285.
Kaleel,M. *et al.* (2019) PaleAle 5.0: prediction of protein relative solvent accessibility by deep learning. *Amino Acids*, **51**, 1289–1296.
Käll,L. *et al.* (2004) A combined transmembrane topology and signal peptide prediction method. *J. Mol. Biol.*, **338**, 1027–1036.
Kiranyaz,S. *et al.* (2021) 1D convolutional neural networks and applications: A survey. *Mech. Syst. Signal Process.*, **151**, 107398.
Klausen,M.S. *et al.* (2019) NetSurfP-2.0: Improved prediction of protein structural features

by integrated deep learning. *Proteins*, **87**, 520–527.
Krebs,J.E. *et al.* (2017) Lewin's GENES XII Jones & Bartlett Learning.
Krogh,A. (2008) What are artificial neural networks? *Nat. Biotechnol.*, **26**, 195–197.
Kryshtafovych,A. *et al.* (2021) Critical assessment of methods of protein structure prediction (CASP)-Round XIV. *Proteins*, **89**, 1607–1617.
Kyte,J. and Doolittle,R.F. (1982) A simple method for displaying the hydropathic character of a protein. *J. Mol. Biol.*, **157**, 105–132.
Landrum,M.J. *et al.* (2018) ClinVar: improving access to variant interpretations and supporting evidence. *Nucleic Acids Res.*, **46**, D1062–D1067.
Lappalainen,T. and MacArthur,D.G. (2021) From variant to function in human disease genetics. *Science*, **373**, 1464–1468.
Lecun,Y. *et al.* (1998) Gradient-based learning applied to document recognition. *Proc. IEEE*, **86**, 2278–2324.
Lee,B. and Richards,F.M. (1971) The interpretation of protein structures: estimation of static accessibility. *J. Mol. Biol.*, **55**, 379–400.
Li,B. *et al.* (2009) Automated inference of molecular mechanisms of disease from amino acid substitutions. *Bioinformatics*, **25**, 2744–2750.
Li,B.-Q. *et al.* (2012) Prediction of protein-protein interaction sites by random forest algorithm with mRMR and IFS. *PLoS One*, **7**, e43927.
Lin,Z. *et al.* (2022) Evolutionary-scale prediction of atomic level protein structure with a language model. *bioRxiv*, 2022.07.20.500902.
Liu,B. *et al.* (2009) Prediction of protein binding sites in protein structures using hidden Markov support vector machine. *BMC Bioinformatics*, **10**, 381.
Li,Y. *et al.* (2021) DELPHI: accurate deep ensemble model for protein interaction sites prediction. *Bioinformatics*, **37**, 896–904.
Lomize,M.A. *et al.* (2012) OPM database and PPM web server: resources for positioning of proteins in membranes. *Nucleic Acids Res.*, **40**, D370–6.
Manfredi,M. *et al.* (2021) DeepREx-WS: A web server for characterising protein–solvent interaction starting from sequence. *Comput. Struct. Biotechnol. J.*, **19**, 5791–5799.
Manfredi,M. *et al.* (2022) E-SNPs&GO: embedding of protein sequence and function improves the annotation of human pathogenic variants. *Bioinformatics*, **38**, 5168–5174.
Manfredi,M. *et al.* (2023) ISPRED-SEQ: Deep neural networks and embeddings for predicting interaction sites in protein sequences. *J. Mol. Biol.*, 167963.
Martelli,P.L. *et al.* (2002) A sequence-profile-based HMM for predicting and discriminating beta barrel membrane proteins. *Bioinformatics*, **18 Suppl 1**, S46–53.
Martelli,P.L. *et al.* (2016) Large scale analysis of protein stability in OMIM disease related human protein variants. *BMC Genomics*, **17 Suppl 2**, 397.
McKusick,V.A. (1998) Mendelian Inheritance in Man: A Catalog of Human Genes and Genetic Disorders Johns Hopkins University Press.
Meersche,Y.V. *et al.* (2021) MEDUSA: Prediction of Protein Flexibility from Sequence. *Journal of Molecular Biology*, **433**, 166882.
Meier,J. *et al.* (2021) Language models enable zero-shot prediction of the effects of mutations on protein function. *bioRxiv*, 29287–29303.
Menard,S. (2002) Applied Logistic Regression Analysis SAGE.
Miller,S. *et al.* (1987) The accessible surface area and stability of oligomeric proteins. *Nature*, **328**, 834–836.
Mirdita,M. *et al.* (2017) Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic Acids Res.*, **45**, D170–D176.
Mohri,M. *et al.* (2018) Foundations of Machine Learning, second edition MIT Press.
Necci,M. *et al.* (2020) MobiDB-lite 3.0: fast consensus annotation of intrinsic disorder

flavours in proteins. *Bioinformatics*.
Nelson,D.L. and Cox,M. (2021) Lehninger Principles of Biochemistry: International Edition Macmillan Learning.
Ng,P.C. and Henikoff,S. (2001) Predicting deleterious amino acid substitutions. *Genome Res.*, **11**, 863–874.
Niculescu-Mizil,A. and Caruana,R. (2005) Predicting good probabilities with supervised learning. *Proceedings of the 22nd international conference on Machine learning - ICML '05*.
Niroula,A. *et al.* (2015) PON-P2: prediction method for fast and reliable identification of harmful variants. *PLoS One*, **10**, e0117380.
Ofer,D. *et al.* (2021) The language of proteins: NLP, machine learning & protein sequences. *Comput. Struct. Biotechnol. J.*, **19**, 1750–1758.
Pal,S. *et al.* (2020) Big data in biology: The hope and present-day challenges in it. *Gene Reports*, **21**, 100869.
Pan,X.-Y. *et al.* (2010) Large-scale prediction of human protein-protein interactions from amino acid sequence based on latent topic features. *J. Proteome Res.*, **9**, 4992–5001.
Pejaver,V. *et al.* (2020) Inferring the molecular and phenotypic impact of amino acid variants with MutPred2. *Nat. Commun.*, **11**, 5918.
Porollo,A. and Meller,J. (2007) Prediction-based fingerprints of protein-protein interactions. *Proteins*, **66**, 630–645.
Raffel,C. *et al.* (2019) Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv [cs.LG]*.
Raimondi,D. *et al.* (2017) DEOGEN2: prediction and interactive visualization of single amino acid variant deleteriousness in human proteins. *Nucleic Acids Res.*, **45**, W201–W206.
Raina,R. *et al.* (2007) Self-taught learning: transfer learning from unlabeled data. In, *Proceedings of the 24th international conference on Machine learning*, ICML '07. Association for Computing Machinery, New York, NY, USA, pp. 759–766.
Rives,A. *et al.* (2021) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. U. S. A.*, **118**.
Rodrigues,J.P.G.L.M. *et al.* (2015) Information-Driven Structural Modelling of Protein–Protein Interactions. In, Kukol,A. (ed), *Molecular Modeling of Proteins*. Springer New York, New York, NY, pp. 399–424.
Rose,G.D. *et al.* (1985) Hydrophobicity of amino acid residues in globular proteins. *Science*, **229**, 834–838.
Rosenblatt,F. (1958) The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, **65**, 386–408.
Rost,B. and Sander,C. (1994) Conservation and prediction of solvent accessibility in protein families. *Proteins*, **20**, 216–226.
Savojardo,C. *et al.* (2019) Functional and Structural Features of Disease-Related Protein Variants. *Int. J. Mol. Sci.*, **20**.
Savojardo,C. *et al.* (2017) ISPRED4: interaction sites PREDiction in protein structures with a refining grammar model. *Bioinformatics*, **33**, 1656–1663.
Savojardo,C. *et al.* (2012) Machine-Learning Methods to Predict Protein Interaction Sites in Folded Proteins. In, *Computational Intelligence Methods for Bioinformatics and Biostatistics*. Springer Berlin Heidelberg, pp. 127–135.
Savojardo,C., Martelli,P.L., *et al.* (2020) Protein–Protein Interaction Methods and Protein Phase Separation. *Annu. Rev. Biomed. Data Sci.*, **3**, 89–112.
Savojardo,C., Manfredi,M., *et al.* (2020) Solvent Accessibility of Residues Undergoing Pathogenic Variations in Humans: From Protein Structures to Protein Sequences. *Front*

Mol Biosci, **7**, 626363.
Schwarz,J.M. *et al.* (2010) MutationTaster evaluates disease-causing potential of sequence alterations. *Nat. Methods*, **7**, 575–576.
Seber,G.A.F. and Lee,A.J. (2003) Linear Regression Analysis John Wiley & Sons.
Shefchek,K.A. *et al.* (2020) The Monarch Initiative in 2019: an integrative data and analytic platform connecting phenotypes to genotypes across species. *Nucleic Acids Res.*, **48**, D704–D715.
Shrake,A. and Rupley,J.A. (1973) Environment and exposure to solvent of protein atoms. Lysozyme and insulin. *J. Mol. Biol.*, **79**, 351–371.
Šikić,M. *et al.* (2009) Prediction of Protein–Protein Interaction Sites in Sequences and 3D Structures by Random Forests. *PLoS Comput. Biol.*, **5**, e1000278.
Singh,J. *et al.* (2021) SPOT-1D-Single: improving the single-sequence-based prediction of protein secondary structure, backbone angles, solvent accessibility and half-sphere exposures using a large training set and ensembled deep learning. *Bioinformatics*, **37**, 3464–3472.
Steinegger,M. *et al.* (2019) HH-suite3 for fast remote homology detection and deep protein annotation. *BMC Bioinformatics*, **20**, 473.
Steinegger,M. and Söding,J. (2017) MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.*, **35**, 1026–1028.
Stringer,B. *et al.* (2022) PIPENN: Protein Interface Prediction from sequence with an Ensemble of Neural Nets. *Bioinformatics*, **38**, 2111–2118.
Strodthoff,N. *et al.* (2020) UDSMProt: universal deep sequence models for protein classification. *Bioinformatics*, **36**, 2401–2409.
Suzek,B.E. *et al.* (2015) UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, **31**, 926–932.
Tarafder,S. *et al.* (2018) RBSURFpred: Modeling protein accessible surface area in real and binary space using regularized and optimized regression. *J. Theor. Biol.*, **441**, 44–57.
Tien,M.Z. *et al.* (2013) Maximum allowed solvent accessibilites of residues in proteins. *PLoS One*, **8**, e80635.
Touw,W.G. *et al.* (2015) A series of PDB-related databanks for everyday needs. *Nucleic Acids Res.*, **43**, D364–8.
UniProt Consortium (2023) UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Res.*, **51**, D523–D531.
Vaswani,A. *et al.* (2017) Attention is all you need. *Adv. Neural Inf. Process. Syst.*, **30**.
Vihinen,M. (2021) Functional effects of protein variants. *Biochimie*, **180**, 104–120.
Voet,D. and Voet,J.G. (2011) Biochemistry, 4-th Edition. *NewYork: John Wiley& SonsInc*.
Walsh,I. *et al.* (2021) DOME: recommendations for supervised machine learning validation in biology. *Nat. Methods*, **18**, 1122–1127.
Warden,A.C. *et al.* (2015) Rational engineering of a mesohalophilic carbonic anhydrase to an extreme halotolerant biocatalyst. *Nature Communications*, **6**.
Wei,Z.-S. *et al.* (2015) A Cascade Random Forests Algorithm for Predicting Protein-Protein Interaction Sites. *IEEE Trans. Nanobioscience*, **14**, 746–760.
Wei,Z.-S. *et al.* (2016) Protein–protein interaction sites prediction by ensembling SVM and sample-weighted random forests. *Neurocomputing*, **193**, 201–212.
Wu,W. *et al.* (2017) Accurate prediction of protein relative solvent accessibility using a balanced model. *BioData Min.*, **10**, 1.
Yang,Y. *et al.* (2022) PON-All: Amino Acid Substitution Tolerance Predictor for All Organisms. *Front Mol Biosci*, **9**, 867572.
Zhang,B. *et al.* (2019) Sequence-based prediction of protein-protein interaction sites by simplified long short-term memory network. *Neurocomputing*, **357**, 86–100.

Zhang,H. (May 17-19 2004) The optimality of naive Bayes. In, *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004)*.
Zhang,J. *et al.* (2019) Comprehensive review and empirical analysis of hallmarks of DNA-, RNA- and protein-binding residues in protein chains. *Briefings in Bioinformatics*, **20**, 1250–1268.
Zhang,J. and Kurgan,L. (2018) Review and comparative assessment of sequence-based predictors of protein-binding residues. *Brief. Bioinform.*, **19**, 821–837.
Zhang,J. and Kurgan,L. (2019) SCRIBER: accurate and partner type-specific prediction of protein-binding residues from proteins sequences. *Bioinformatics*, **35**, i343–i353.

Appendix: Publications

frontiers in Molecular Biosciences

ORIGINAL RESEARCH
published: 07 January 2021
doi: 10.3389/fmolb.2020.626363

Solvent Accessibility of Residues Undergoing Pathogenic Variations in Humans: From Protein Structures to Protein Sequences

Castrense Savojardo[1], *Matteo Manfredi*[1], *Pier Luigi Martelli*[1]* *and Rita Casadio*[1,2]

[1] Biocomputing Group, Department of Pharmacy and Biotechnologies, University of Bologna, Bologna, Italy, [2] Institute of Biomembranes, Bioenergetics and Molecular Biotechnologies of the National Research Council, Bari, Italy

Solvent accessibility (SASA) is a key feature of proteins for determining their folding and stability. SASA is computed from protein structures with different algorithms, and from protein sequences with machine-learning based approaches trained on solved structures. Here we ask the question as to which extent solvent exposure of residues can be associated to the pathogenicity of the variation. By this, SASA of the wild-type residue acquires a role in the context of functional annotation of protein single-residue variations (SRVs). By mapping variations on a curated database of human protein structures, we found that residues targeted by disease related SRVs are less accessible to solvent than residues involved in polymorphisms. The disease association is not evenly distributed among the different residue types: SRVs targeting glycine, tryptophan, tyrosine, and cysteine are more frequently disease associated than others. For all residues, the proportion of disease related SRVs largely increases when the wild-type residue is buried and decreases when it is exposed. The extent of the increase depends on the residue type. With the aid of an in house developed predictor, based on a deep learning procedure and performing at the state-of-the-art, we are able to confirm the above tendency by analyzing a large data set of residues subjected to variations and occurring in some 12,494 human protein sequences still lacking three-dimensional structure (derived from HUMSAVAR). Our data support the notion that surface accessible area is a distinguished property of residues that undergo variation and that pathogenicity is more frequently associated to the buried property than to the exposed one.

Keywords: solvent accessible surface area, relative solvent accessibility, protein variations, prediction of solvent accessible surface, pathogenic protein variations

OPEN ACCESS

Edited by:
Sarah Teichmann,
Wellcome Sanger Institute (WT),
United Kingdom

Reviewed by:
Joost Schymkowitz,
VIB & KU Leuven Center for Brain & Disease Research, Belgium
Carlo Travaglini-Allocatelli,
Sapienza University of Rome, Italy

***Correspondence:**
Pier Luigi Martelli
pierluigi.martelli@unibo.it

Specialty section:
This article was submitted to Structural Biology, a section of the journal Frontiers in Molecular Biosciences

Received: *05 November 2020*
Accepted: *07 December 2020*
Published: *07 January 2021*

Citation:
Savojardo C, Manfredi M, Martelli PL and Casadio R (2021) Solvent Accessibility of Residues Undergoing Pathogenic Variations in Humans: From Protein Structures to Protein Sequences. Front. Mol. Biosci. 7:626363. doi: 10.3389/fmolb.2020.626363

INTRODUCTION

In structural bioinformatics, Solvent Accessible Surface Area (SASA) [or briefly Accessible Surface Area (ASA)] of proteins has always been considered a main feature for determining protein folding and stability. Early computational studies (Lee and Richards, 1971; Chothia, 1976; Miller et al., 1987, and references therein) emphasized the role of solvent exposed vs. non-exposed amino acid residues in determining the protein structure. Typically, ASA is defined as the polar solvent accessible area of a given protein, and it is computed by means of a solvent molecule, which probes the protein surface beyond the van der Waals radius. After the first rolling ball algorithm

(Shrake and Rupley, 1973), many alternatives became available for computing ASA from the atomic coordinates of the protein in its folded and unfolded state [for review see Ali et al. (2014)]. Evidently, ASA is a function of the three dimensional structure of the protein and, based on ASA values, amino acid residues of a protein can be classified as buried or exposed (Kabsch and Sander, 1983), a property that is conserved through evolution in protein families (Rost and Sander, 1994). ASA is routinely computed as an absolute value or as Relative Solvent Accessibility (RSA), when the ASA value is divided by the maximum possible solvent accessible surface area of the residue (Tien et al., 2013). ASA gained also a pivot role in detecting protein-protein interfaces of molecular complexes in the Protein Data Bank (PDB) [for review see Savojardo et al. (2020), and references therein].

With the advent of machine and deep learning-based approaches (Baldi, 2018), many methods became available for predicting RSA and ASA. They differ mainly in the machine learning approach, the volume of the database of protein structures and the predicted output (ASA, RSA, or binary classification) (Rost and Sander, 1994; Pollastri et al., 2002; Drozdetskiy et al., 2015; Ma and Wang, 2015; Fan et al., 2016; Wu et al., 2017; Kaleel et al., 2019; Klausen et al., 2019).

Surface accessible area of residues can be important also for functional annotation of disease related protein variants. However, this property has been rarely included into the physico-chemical characteristics adopted to describe the residues undergoing variations (Chen and Zhou, 2005; Martelli et al., 2016; Savojardo et al., 2019).

In this study, we investigate the relation between the pathogenicity of human protein variations and the solvent exposure of the residues undergoing variation (wild-type residues). To this aim, we provide an updated version of a highly curated dataset of Single Residue Variations (SRVs) occurring in human proteins that can be mapped in high-quality structures deposited in the Protein Data Bank (PDB). The dataset, here referred to as HVAR3D-2.0, is generated from data available at the HUMASVAR database and builds on top of data previously analyzed in a different study (Savojardo et al., 2019). On this structural dataset, we explore the relationship between pathogenicity of SRVs and the solvent accessibility of the corresponding wild-type residues. In particular, we determine that the majority (67%) of disease-related SRVs occur in buried positions whereas neutral SRVs occur mostly (64.3%) in exposed residues. Moreover, SRVs targeting specific residue types such as glycine, tryptophan, tyrosine, and cysteine, are more frequently associated with disease than others are. Finally, for all residues, and in particular for asparagine, glutamine, histidine, and lysine, the proportion of disease related SRVs largely increases when the wild-type residue is buried, and decreases when it is exposed, confirming that, among other factors, the context can be associated to the pathogenicity of the variations (Casadio et al., 2011).

We extended the above analysis to a larger set of variations included in HUMSAVAR and collected in a dataset called HVARSEQ. In order to estimate the solvent accessibility of all residues undergoing disease-related or neutral SRVs in human proteins, we developed an in-house method based on deep-learning for predicting solvent exposure from sequence. Our method performance is comparable to state-of-the-art methods. We apply it to all the residues of human protein sequences, undergoing pathogenic and neutral SRVs in HVARSEQ.

Results of the large-scale analysis on protein sequences support what observed in protein structures and confirm the different distribution buried/exposed wild-type residues in disease-related and neutral SRVs. Our data suggest that solvent accessibility is a distinguished property of wild type residues undergoing pathogenic variations.

MATERIALS AND METHODS

Variation Databases

All human Single-Residue Variations (SRVs) were collected from HUMASVAR version 2020_04 (Aug 2020). As a first filtering step, we retained variations labeled as "Disease" and "Polymorphism," neglecting all variations labeled as "Unclassified." Disease-related SRVs not associated with OMIM diseases were excluded. After this procedure we ended up with a large set of SRVs occurring on human protein sequence. Here this dataset is referred to as HVARSEQ (Human VARiations in SEQuences)

In order to build the structural dataset (here referred to as HVAR3D-2.0, Human VARiations in three Dimensional structures), we firstly identified, among all the sequences included in HVARSEQ, the subset of proteins endowed with a PDB structure meeting the following criteria:

- Coverage of the corresponding UniProtKB sequence is $\geq$70%;
- Experimental method is X-ray crystallography;
- Resolution is $\leq$ 3Å.

The mapping of SRV positions on protein structure was performed using data from the Structure Integration with Function, Taxonomy and Sequence (SIFTS) project[1]. Protein structures having ambiguous or wrong SIFTS mapping files were excluded from the dataset.

Computing Solvent Exposure

The absolute Accessible Surface Area (ASA) of each wild-type residue undergoing variation has been computed using the DSSP program (Kabsch and Sander, 1983). Relative Solvent Accessibility (RSA) values were then obtained dividing absolute ASA values in $Å^2$ by residue-specific maximal accessibility values, as extracted from the Sander and Rost scale (Rost and Sander, 1994). Finally, each residue has been classified as buried (B) if its RSA was below 20%, and exposed (E) otherwise.

Computing P_D, $P_{D|R}$, $P_{D|B,R}$, and $P_{D|E,R}$

In this study, the background probability of a wild-type residue to be disease associated in a dataset of wild-type residues is computed as follows:

$$P_D = \frac{n_D}{N} \tag{1}$$

[1]https://www.ebi.ac.uk/pdbe/docs/sifts/.

where n_D and N are the number of wild-type residues undergoing disease-related variations and the total number of wild-type residues undergoing variations (disease related or not) in the dataset, respectively.

The conditional probability of being disease related when variated, given a wild-type residue R, is computed as follows:

$$P_{D|R} = \frac{n_{DR}}{n_R} \quad (2)$$

where n_{DR} and n_R are the number of wild-type residues of a given R type, which are disease related upon variations, and the total number of R residues in the whole dataset, respectively.

The conditional probability of a wild-type residue R to be disease related upon variation when buried is computed as:

$$P_{D|B,R} = \frac{n_{DBR}}{n_{BR}} \quad (3)$$

where n_{DBR} and n_{BR} are the number of buried wild type R residue in the set of wild type disease related upon variation and the total number of buried R wild type residues, respectively.

Similarly, the conditional probability of a wild-type residue R to be disease related upon variation when exposed is computed as:

$$P_{D|E,R} = \frac{n_{DER}}{n_{ER}} \quad (4)$$

where n_{DER} and n_{ER} are the number of exposed wild type R residue in the set of wild-type disease related upon variation and the total number of exposed R wild type residues, respectively.

All the above probabilities are estimated considering the structural dataset HVAR3D-2.0, and by computing the residue solvent accessibility with the DSSP program. Moreover, we extended the analysis to the whole HVARSEQ sequence dataset, by estimating the residue exposure state (buried and exposed) with a predictor implemented in-house and described in the following section.

Predicting Solvent Accessibility From the Protein Sequence

The method implements a deep-learning architecture processing an input based on the following descriptors:

- The residue one-hot encoding, representing primary sequence information;
- Evolutionary information encoded with a protein sequence profile, as extracted from multiple sequence alignment generated using the HHblits version 3 program (Steinegger et al., 2019). We performed two search iterations with default parameters against the Uniclust30 database (Mirdita et al., 2017).

Our deep architecture processes the input using three cascading Bidirectional Long-Short Term Memory (BLSTM) layers (Graves and Schmidhuber, 2005). BLSTMs belong to the class of LSTM (Hochreiter and Schmidhuber, 1997), a special recurrent neural network architecture well-suited for processing protein sequence data and extracting significant sequential relations between elements of the sequence. BLSTMs are an extension of LSTMs performing a double scanning of the input sequence, from left to right and vice versa, in order to better capture the sequential relations among sequence positions. The adoption of the recurrent BLSTM allows the method to take into consideration the local sequence context without the explicit use of a fixed-size window centered on each residue.

The output of the third recurrent layer is then provided as input to a time-distributed fully connected layer adopting a sigmoid activation function. This layer is responsible for the final, binary classification of each residue in the sequence into buried or exposed classes. In particular, the numerical output value in the range [0, 1] attached to each residue is interpreted as a probability p of being exposed: all residues with $p \geq 0.5$ are predicted as exposed while those with $p < 0.5$ are classified as buried.

The dataset adopted to train and test the predictor presented in this study has been extracted from the Protein Data Bank (interrogated Oct 15, 2019) (Berman, 2000). Overall, the dataset comprises 2532 non-redundant, author-declared functional monomeric PDB structures, obtained with X-ray crystallography at < 2.5 Å resolution and covering more than 70% of corresponding UniProtKB sequences. All proteins in the dataset share <30% sequence identity. This dataset was then randomly split into a training set, comprising 2,352 sequences, and an independent blind test set including 200 sequences. Proteins in the training set were further split into 10 equally-sized sets for setting the values of hyperparameters with a cross-validation procedure.

Solvent exposure for training/testing data has been computed using DSSP as detailed in Section: Computing solvent exposure. The residues were classified into buried and exposed using a RSA threshold of 20%. Using this threshold, the set of residues is roughly divided into equally sized subsets comprising 52% and 48% of buried and exposed residues, respectively, providing balanced datasets for training and testing.

RESULTS

HVAR3D-2.0: A Dataset of Variations Covered by 3D Structure

The structural dataset collected in this work, here referred to as HVAR3D-2.0, is an updated version of the dataset described in a previous study (Savojardo et al., 2019). The dataset has been derived by mapping on PDB structures OMIM-related and neutral SRVs annotated in the HUMSAVAR database[2], release 2020_08 (Aug, 2020). Only structures determined with X-ray crystallography with resolution ≤3Å and covering ≥70% of the corresponding UniProtKB sequences were selected. After this stringent filtering, we ended-up with a high-quality dataset comprising 10,760 human SRVs occurring on 1,255 PDB entries (corresponding to 1,285 protein chains). The set includes 6,778 and 3,982 disease-related and neutral SRVs, respectively. **Table 1** lists a summary of the HVAR3D-2.0 content. The HVAR3D-2.0 dataset is available in **Supplementary Table 1** in TSV format.

[2]https://www.uniprot.org/docs/humasavar

TABLE 1 | Statistics of HVAR3D 2.0 dataset.

Description	Counts (#)
PDB structures	1,255
PDB chains	1,285
Distinct SRV positions	9,379
SRVs	10,760
Disease-related SRVs	6,778
Neutral SRVs	3,982

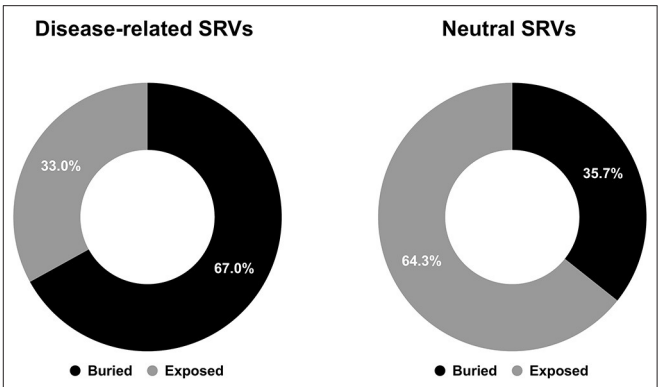


FIGURE 1 | Pie charts showing the fractions of buried/exposed wild-type residues undergoing disease-related (left) and neutral (right) SRVs in the HVAR3D-2.0 dataset, respectively.

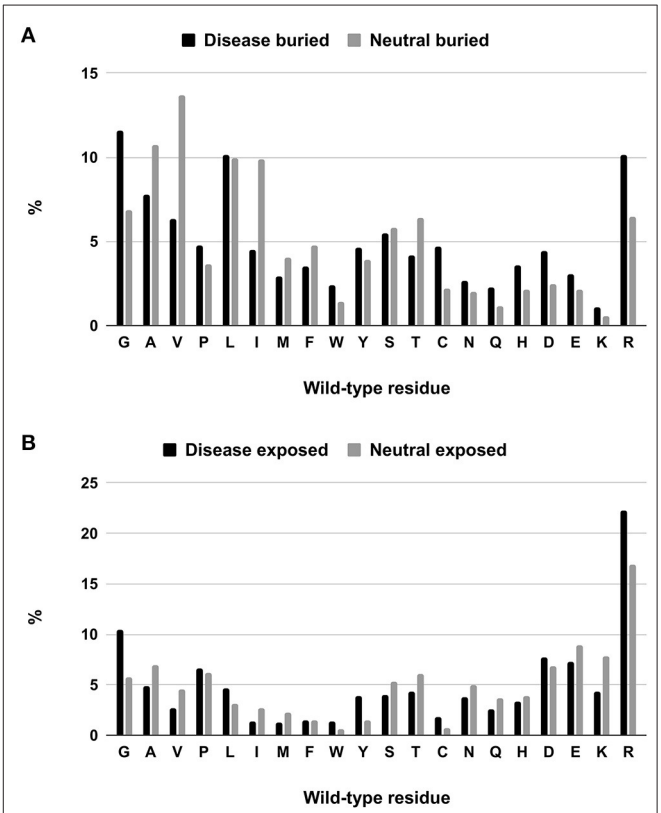


FIGURE 2 | Composition of buried **(A)** and exposed **(B)** wild-type residues undergoing disease-related and neutral variations in the HVAR3D-2.0 dataset.

In the present study, we are interested in investigating the relation between the pathogenicity of SRVs and the solvent accessibility of the residue undergoing variation. For this reason, we firstly computed Accessible Surface Area (ASA) values for all 1,285 protein chains included in the HVAR3D dataset using the DSSP program (Kabsch and Sander, 1983). Raw ASAs were then converted into Relative Solvent Accessibility (RSA) values using the Rost and Sander maximal accessibility scale (Rost and Sander, 1994). Finally, all residues with RSA $\geq$ 20% were labeled as exposed (E) or buried (B) otherwise. This threshold (or similar ones, in the range of 15–25% RSA) is routinely adopted for computing the protein surfaces and deriving classification datasets in many studies (Thompson and Goldstein, 1996; Mucchielli-Giorgi et al., 1999; Pollastri et al., 2002; Kaleel et al., 2019), since it roughly divides the set of residues in a protein in two equally-sized subsets. In HVAR3D, using a 20% RSA threshold, we obtain 55% and 45% of residues classified as buried and exposed, respectively, corresponding to a realistic characterization of the protein interior (accounting for completely and partially buried residues) and surface (Miller et al., 1987). Preliminary analysis highlighted that the choice of the RSA threshold (in the reasonable range of 15–25% RSA) only minorly affects the conclusions drawn in this study (data not shown). For this reason, all the subsequent analyses were performed using the aforementioned threshold.

Focusing our attention to structure positions undergoing SRVs, we firstly computed the different proportions of buried and exposed wild-type residues associated to disease-related and neutral SRVs. As shown in **Figure 1**, 67% of wild-type residues undergoing disease-related variations are located in buried positions and about 64% of wild-type residues involved in neutral variations are exposed. This conclusion corroborates, on a much larger structural database, results partially reported in previous studies (Martelli et al., 2016; Savojardo et al., 2019). The relative abundance of disease-related variations in buried positions of the protein and of neutral ones in exposed positions suggests that the solvent accessibility of the variated position is a further property to consider when determining the pathogenicity of a variation.

Analyzing Distributions of Variated Wild-Type Residues in the Structure Database

We tackle the problem of associating solvent exposure to a specific wild-type residue as a characteristic feature to be associated to its variation type (neutral or disease related). We compute the relative frequency of occurrence in the buried and exposed sets of each residue undergoing a disease related or neutral variation (**Figures 2A,B**). It is evident that while some residue types are more often disease related when variated in the buried state (Q, H, D, E, K), others (including G, W, C, and R) are disease related upon variation in either state.

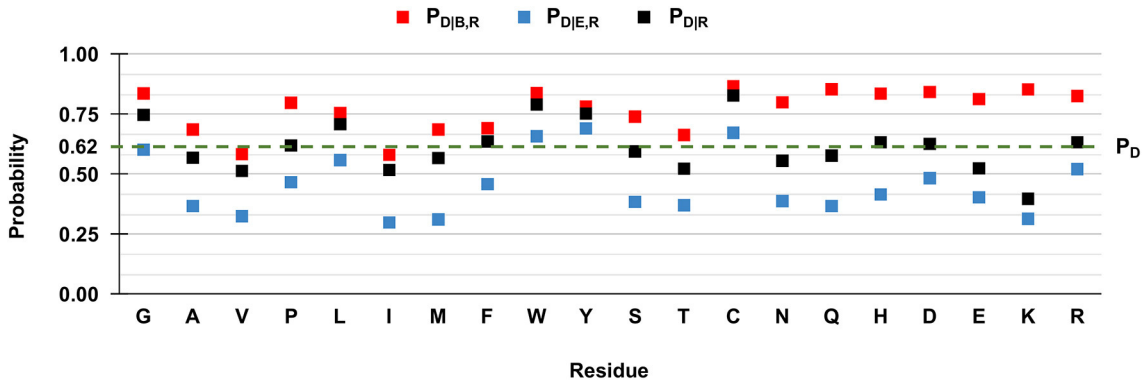


FIGURE 3 | Probabilities of the 20 wild-type residues undergoing disease-related variations, depending on the wild type residue and the exposure state in HVAR3D-2.0. Buried and exposure state of each residue position are estimated with DSSP as described in Section: Analyzing distributions of variated wild-type residues in the structure database. $\mathbf{P_D}$: the probability of a wild-type residue (position) to be disease associated in the HVAR3D-2.0 dataset [see Equation (1)]. $\mathbf{P_{D|R}}$: the conditional probability of being disease related when variated, given a wild-type residue [see Equation (2)]. $\mathbf{P_{D|B,R}}$: the conditional probability of a wild-type residue to be disease related upon variation when buried [see Equation (3)]. $\mathbf{P_{D|E,R}}$: the conditional probability of a wild-type residue to be disease related upon variation when exposed [see Equation (4)].

However, when we compute the conditional probabilities per residue type, clearly the tendency of the majority of the wild-type residues is that of being disease-related upon variation when buried (red squares in **Figure 3**). Indeed, in **Figure 3** we show to which extent the knowledge of the solvent exposure changes the *a priori* probability of a given residue type to be associated with disease. For each residue type R, we report the conditional probability of being associated to disease ($P_{D|R}$, black squares) and how the two conditional probabilities ($P_{D|B,R}$ and $P_{D|E,R}$ in red and blue squares, respectively) change, given that the variated residue is buried or exposed. We contrast these values to the baseline frequency of disease related variations in the HVAR3D-2.0 dataset, referred to as P_D and equal to 0.62.

In **Figure 3**, when comparing $P_{D|R}$ of each residue R (black squares) with the baseline value P_D, it is evident that not all the residues are equally likely to be associated with disease when variated. Residues like glycine (G), leucine (L) tryptophan (W), tyrosine (Y), and cysteine (C) show values of $P_{D|R}$ that are higher than the baseline, indicating that their variations are frequently associated to disease in the database. Furthermore, for all residues the relation $P_{D|B,R} > P_{D|R} > P_{D|E,R}$ holds. This means that for all residue types, the probability that SRVs are related to disease is higher when the wild-type residue is buried (red squares) than when it is exposed (blue squares). The extent of this difference depends on the residue type and it is remarkable for asparagine (N), glutamine (Q), histidine (H), and lysine (K). All these residues are polar and abundant on the protein surface (data not shown). On average, when variated, they are associated to disease with a frequency comparable or lower than the baseline 0.62. However, when variations of these residue types occur in buried positions, the frequency of disease related variations raises to values around 0.8, reaching 0.85 in the case of glutamine (Q) and lysine (K). Remarkably, for three residues [tryptophan (W), tyrosine (Y) and cysteine (C)] the frequency of disease-related variation is higher than the baseline, rather independently of the exposure state. Conversely, the fraction of disease-related variations of valine (V) and isoleucine (I) is lower than the baseline, independently of their accessibility.

Overall, these findings highlight a relation between the pathogenicity of the variation and the solvent accessibility of the wild-type residue and show that the extent of the association depends on the residue type. In all cases, variations occurring in buried positions are more likely to be disease-related. This is particularly so for charged residues, for polar residues such as asparagine (N), glutamine (Q) and histidine (H), and for proline (P), cysteine (C), and tryptophan (W).

HVARSEQ: A Dataset of Protein Sequences With Variations

Here we make use of computational prediction of solvent accessibility to extend our analysis to all the positions undergoing variations contained in HUMSAVAR. From the HUMSAVAR database, release 2020_08 (Aug, 2020), we collected all polymorphisms and all OMIM-related SRVs occurring in protein sequences. Unclassified SRVs were filtered-out from the set. Overall, 69,385 SRVs were collected. 29,949 and 39,436 SRVs are disease-related and neutral, respectively, occurring on 12,494 protein sequences. Here, this extended set of protein sequences is referred to as HVARSEQ. In **Table 2** we summarize the basic statistics of the dataset. The HVARSEQ dataset is available in **Supplementary Table 2** in TSV format.

Predicting Solvent Accessibility

For computing solvent accessibility from protein sequences, we implemented an in-house method for predicting solvent exposure from sequence. The method is based on deep-learning processing of several input features, which encode the protein sequence and the sequence profile (see Materials and Methods for more details on the method). Our method classifies each residue of the sequence into two classes: buried (B), corresponding

TABLE 2 | Statistics of HVARSEQ dataset.

Description	Counts (#)
UniProtKB sequences	12,494
Distinct SRV positions	64,869
SRVs	69,385
Disease-related SRVs	29,949
Neutral SRVs	39,436

TABLE 3 | Performance of our deep learning-based method for predicting solvent exposure from protein sequence.

Scoring index	Dataset		
	Cross-validation	Blind test	HVAR3D-2.0
MCC	0.63	0.63	0.60
Q2 (accuracy)	81%	82%	80%
F1	81%	82%	80%

TABLE 4 | Performance of different methods for solvent accessibility prediction on the blind test set described in this study comprising 200 protein sequences.

Method	MCC	Q2 %	F1 %
PaleAle 5.0	0.65	82	84
NetSurfP-2.0	0.67	83	81
Our method	0.63	82	82

to residues whose RSA is lower than 20%, and exposed (E), corresponding to residues with RSA $\geq$ 20%.

Performances are listed in **Table 3** and are evaluated adopting three different testing sets (by adopting a cross validation procedure (leftmost column); on the blind test (central column); on our HVAR3D-2.0 dataset, for which solvent exposure can be directly computed using DSSP). Comparing the first two columns, it is evident that our method is robust, achieving generalization performances that are as good and even better than cross-validation results. Overall, our method is able to discriminate buried from exposed residues with Q_2 (accuracy), MCC (Matthew Correlation Coefficient) and F1 equal to 82%, 0.63 and 82%, respectively. When scored on the HVAR3D-2.0 dataset, the performance is almost unchanged, suggesting that our method is quite stable across different datasets.

We also performed a side-by-side comparison between our method and two state-of-the-art approaches, namely PaleAle5.0 (Kaleel et al., 2019) and NetSurfP-2.0 (Klausen et al., 2019). Results are reported in **Table 4**. All methods perform quite well, with comparable scoring indexes. It is worth mentioning that the testing set used in this benchmark is non-redundant only with respect to our training set: this condition is not guaranteed for the other two methods evaluated, which adopt different training sets. In general, we can conclude that our method well-compares with recent tools at the state-of-the-art.

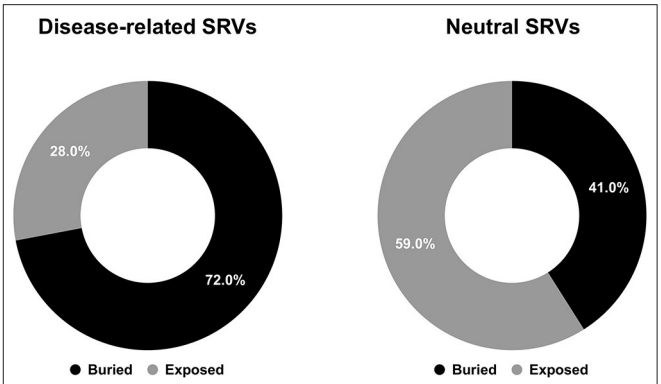


FIGURE 4 | Pie charts showing the fractions of predicted buried/exposed positions disease-related (left) and neutral (right) upon variations in the HVARSEQ dataset, respectively.

Analyzing Distributions of Variated Wild-Type Residues in the Sequence Dataset

After computing solvent accessibility over HVARSEQ, we assessed the proportions of buried and exposed predictions separately on the subsets of residues undergoing disease-related and neutral variations. Results are in **Figure 4**.

As to the prediction, 72% of disease related SRVs occurs in buried positions and 58% of neutral SRVs affect exposed residues. Interestingly, the proportions of buried/exposed positions for disease and neutral SRVs are in agreement with those assessed on the structural dataset (67% and 64.3%, respectively: compare **Figures 1**, **4**). The result further corroborates the notion that residues undergoing disease-related variations are mainly in buried positions.

We then evaluated $P_{D|R}$, $P_{D|B,R}$, and $P_{D|E,R}$ for all the residue types and results are reported in **Figure 5**. We also show the baseline probability P_D (0.43), which represents the proportion of positions that undergo disease-related variations in the HVARSEQ dataset.

The comparison between $P_{D|R}$ and $P_{D,}$ which are both independent from predictions, confirms the finding obtained on the HVAR3D-2.0 dataset: residues such as glycine (G), tryptophan (W), tyrosine (Y), and cysteine (C), when undergoing variation, are more frequently associated to disease than expected from the baseline. In the sequence set, this behavior characterizes also arginine (R) and aspartic acid (D).

Similarly to the structural case, for all residues we have that $P_{D|B,R} > P_{D|R} > P_{D|E,R}$, highlighting that for all residue types, SRVs are more frequently associated to disease when occurring in buried positions than in exposed ones. The tendency is remarkable for the majority of residues, already identified from HVAR3D-2.0 and including asparagine (N), lysine (K), and histidine (H). The analysis on HVARSEQ highlights a difference between $P_{D|B,R}$ and $P_{D|E,R}$ for tryptophan (W) and cysteine (C). However, this discrepancy can be due to prediction errors on these two less abundant (rare) residues in the database. Similarly, to what described for HVAR3D-2.0 (**Figure 3**), the frequency

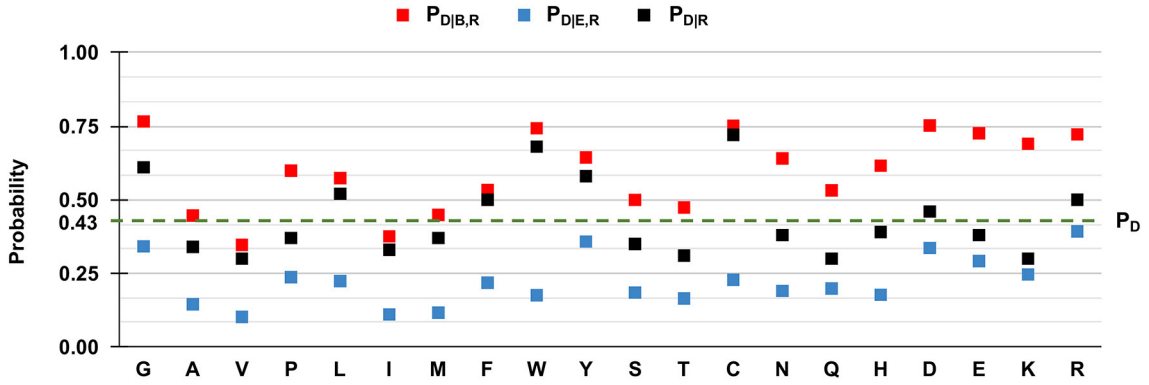


FIGURE 5 | Frequency of disease-related SRVs, depending on the wild type residue and the exposure state in HVARSEQ. Here, buried and exposure states of each residue position have been predicted using the method described in Section Analyzing distributions of variated wild-type residues in the sequence database. $\mathbf{P_D}$: the probability of a wild-type residue (position) to be disease associated in the HVARSEQ dataset [see Equation (1)]. $\mathbf{P_{D|R}}$: the conditional probability of being disease related when variated, given a wild-type residue [see Equation (2)]. $\mathbf{P_{D|B,R}}$: the conditional probability of a wild-type residue to be disease related upon variation when buried [see Equation (3)]. $\mathbf{P_{D|E,R}}$: the conditional probability of a wild-type residue R to be disease related upon variation when exposed [see Equation (4)].

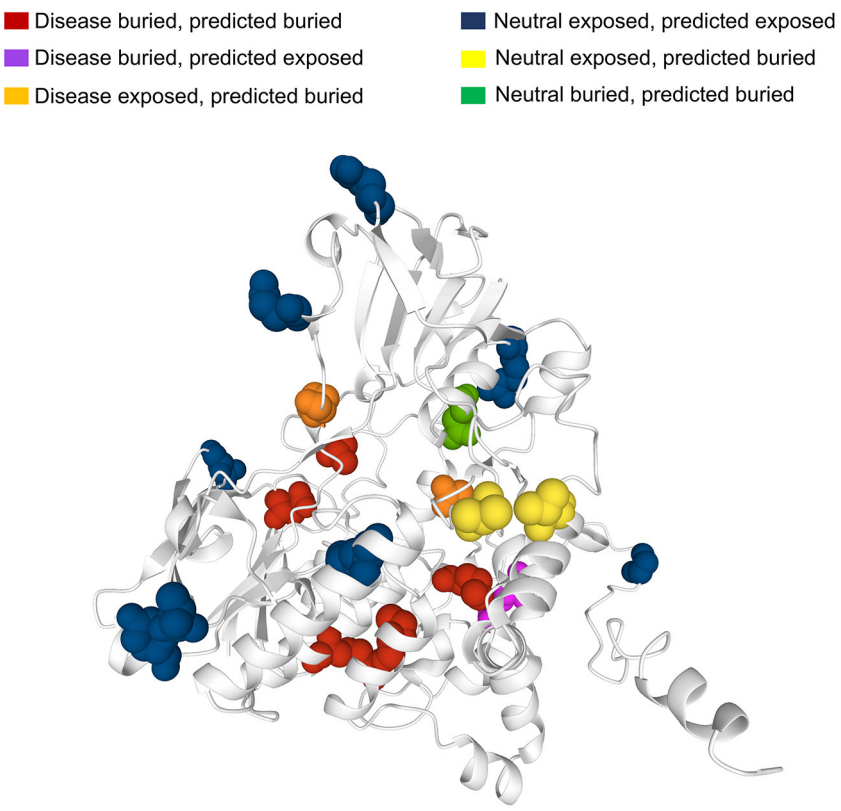


FIGURE 6 | Mapping SASA predictions on a protein model. The model is that of human Dimethylaniline monooxygenase 3 (UniProtKB: P31513) derived from the SWISS-MODEL Repository. Solvent exposure is computed from the available 3D protein model using DSSP. Variation (SVR) positions are highlighted using the spacefill view. In red, buried positions associated to disease-related SRVs and correctly predicted as buried by our method. In magenta, buried disease-related positions wrongly predicted as exposed. In orange, exposed disease-related positions wrongly predicted as buried. In blue, exposed neutral SRV positions correctly predicted as exposed. In yellow, exposed neutral positions wrongly predicted as buried. In green, buried neutral positions correctly predicted as buried.

of disease-related SRVs occurring at valine (V) and isoleucine (I) residues is lower than the baseline, independently of the exposure state.

Case Study

Many human protein sequences, without any associated three-dimensional (3D) structure, are endowed with models that can be derived from the SWISS-MODEL Repository[3], directly linked to the protein UniProtKB file. For sake of curiosity, we took advantage of an example to show the 3D location of our sequence-based prediction. In particular, in **Figure 6** we show the model of the human Dimethylaniline monooxygenase 3 protein (UniProtKB: P31513)[4]. This protein has 19 SRVs in HVARSEQ, eight of which are disease-related and 11 are neutral. Disease-related SRVs are all associated to Trimethylaminuria (OMIM:602079)[5], a disease condition resulting from the abnormal presence of large amounts of volatile and malodorous trimethylamine within the body. In **Figure 6**, we map all the solvent exposure predictions for all SRV positions into the 3D model.

It is evident that the vast majority of disease-related SRVs (6 out of 8) are in buried positions. Of these, five are correctly predicted as buried by our method (in red) while only one is wrongly predicted as exposed (in magenta). Neutral SRVs are mostly exposed (10 out of 11): eight of these are correctly predicted in exposed regions (in blue).

Results illustrate the general trend of what we observed in the structural data set and are consistent with the accuracy of the prediction method.

CONCLUSION AND PERSPECTIVE

In this paper, we focus on the solvent accessible surface area, a property of protein residues, firstly described and computed in several biophysical studies, to which Cyrus Chothia contributed (Chothia, 1976). The property, which nowadays can be computed with machine learning based methods, is here exploited in relation to another important problem: the annotation of variations in human proteins as disease related or not. We took advantage of an ample set of human protein structures to observe that indeed disease related variations occur more frequently in buried regions of the proteins than in solvent accessible surfaces. In turn, neutral polymorphisms are characterized by a more frequent solvent exposure. We then proved that with a deep learning method performing at the state of art, the tendency is observable also in the majority of all the wild-type residues undergoing variations that are presently listed in HUMSAVAR. We suggest that the solvent accessible surface area of wild type residues is a distinguished property to be included among those necessary to annotate pathogenic from non-pathogenic variations.

[3]https://swissmodel.expasy.org/repository
[4]https://www.uniprot.org/uniprot/P31513.
[5]https://www.omim.org/entry/60207

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

RC, PM, and CS: conceptualization and writing. RC, PM, CS, and MM: methodology. MM and CS: software. CS, MM, and PM: data curation and visualization. RC and PM: supervision. All authors contributed to the article and approved the submitted version.

FUNDING

The work was supported by the PRIN2017 grant (project 2017483NH8_002), delivered to CS from the Italian Ministry of University and Research.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fmolb.2020.626363/full#supplementary-material

REFERENCES

Ali, S., Hassan, M. D., Islam, A., and Ahmad, F. (2014). A review of methods available to estimate solvent-accessible surface areas of soluble proteins in the folded and unfolded states. *Curr. Protein Pept. Sci.* 15, 456–476. doi: 10.2174/1389203715666140327114232

Baldi, P. (2018). Deep learning in biomedical data science. *Annu. Rev. Biomed. Data Sci.* 1, 181–205. doi: 10.1146/annurev-biodatasci-080917-013343

Berman, H. M. (2000). The protein data bank. *Nucleic Acids Res.* 28, 235–242. doi: 10.1093/nar/28.1.235

Casadio, R., Vassura, M., Tiwari, S., Fariselli, P., and Martelli, P. L. (2011). Correlating disease related mutations to their effect on protein stability: a large-scale analysis of the human proteome. *Hum. Mutat.* 32, 1161–1170. doi: 10.1002/humu.21555

Chen, H., and Zhou, H.-X. (2005). Prediction of solvent accessibility and sites of deleterious mutations from protein sequence. *Nucleic Acids Res.* 33, 3193–3199. doi: 10.1093/nar/gki633

Chothia, C. (1976). The nature of the accessible and buried surfaces in proteins. *J. Mol. Biol.* 105, 1–12. doi: 10.1016/0022-2836(76)90191-1

Drozdetskiy, A., Cole, C., Procter, J., and Barton, G. J. (2015). JPred4: a protein secondary structure prediction server. *Nucleic Acids Res.* 43, W389–394. doi: 10.1093/nar/gkv332

Fan, C., Liu, D., Huang, R., Chen, Z., and Deng, L. (2016). PredRSA: a gradient boosted regression trees approach for predicting protein solvent accessibility. *BMC Bioinform.* 17:S8. doi: 10.1186/s12859-015-0851-2

Graves, A., and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* 18, 602–610. doi: 10.1016/j.neunet.2005.06.042

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Kabsch, W., and Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22, 2577–2637. doi: 10.1002/bip.360221211

Kaleel, M., Torrisi, M., Mooney, C., and Pollastri, G. (2019). PaleAle 5.0: prediction of protein relative solvent accessibility by deep learning. *Amino Acids* 51, 1289–1296. doi: 10.1007/s00726-019-02767-6

Klausen, M. S., Jespersen, M. C., Nielsen, H., Jensen, K. K., Jurtz, V. I., Sønderby, C. K., et al. (2019). NetSurfP-2.0: improved prediction of protein structural features by integrated deep learning. *Proteins Struct. Funct. Bioinforma.* 87, 520–527. doi: 10.1002/prot.25674

Lee, B., and Richards, F. M. (1971). The interpretation of protein structures: estimation of static accessibility. *J. Mol. Biol.* 55, 379–400. doi: 10.1016/0022-2836(71)90324-X

Ma, J., and Wang, S. (2015). AcconPred: predicting solvent accessibility and contact number simultaneously by a multitask learning framework under the conditional neural fields model. *BioMed Res. Int.* 2015:678764. doi: 10.1155/2015/678764

Martelli, P. L., Fariselli, P., Savojardo, C., Babbi, G., Aggazio, F., and Casadio, R. (2016). Large scale analysis of protein stability in OMIM disease related human protein variants. *BMC Genomics* 17:397. doi: 10.1186/s12864-016-2726-y

Miller, S., Lesk, A. M., Janin, J., and Chothia, C. (1987). The accessible surface area and stability of oligomeric proteins. *Nature* 328, 834–836. doi: 10.1038/328834a0

Mirdita, M., von den Driesch, L., Galiez, C., Martin, M. J., Söding, J., and Steinegger, M. (2017). Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic Acids Res.* 45, D170–D176. doi: 10.1093/nar/gkw1081

Mucchielli-Giorgi, M. H., Hazout, S., and Tufféry, P. (1999). PredAcc: prediction of solvent accessibility. *Bioinformatics* 15, 176–177. doi: 10.1093/bioinformatics/15.2.176

Pollastri, G., Baldi, P., Fariselli, P., and Casadio, R. (2002). Prediction of coordination number and relative solvent accessibility in proteins. *Proteins Struct. Funct. Genet.* 47, 142–153. doi: 10.1002/prot.10069

Rost, B., and Sander, C. (1994). Conservation and prediction of solvent accessibility in protein families. *Proteins Struct. Funct. Bioinforma.* 20, 216–226. doi: 10.1002/prot.340200303

Savojardo, C., Babbi, G., Martelli, P., and Casadio, R. (2019). Functional and structural features of disease-related protein variants. *Int. J. Mol. Sci.* 20:1530. doi: 10.3390/ijms20071530

Savojardo, C., Martelli, P. L., and Casadio, R. (2020). Protein–protein interaction methods and protein phase separation. *Annu. Rev. Biomed. Data Sci.* 3, 89–112. doi: 10.1146/annurev-biodatasci-011720-104428

Shrake, A., and Rupley, J. A. (1973). Environment and exposure to solvent of protein atoms. *Lysozyme and insulin. J. Mol. Biol.* 79, 351–371. doi: 10.1016/0022-2836(73)90011-9

Steinegger, M., Meier, M., Mirdita, M., Vöhringer, H., Haunsberger, S. J., and Söding, J. (2019). HH-suite3 for fast remote homology detection and deep protein annotation. *BMC Bioinform.* 20:473. doi: 10.1186/s12859-019-3019-7

Thompson, M. J., and Goldstein, R. A. (1996). Predicting solvent accessibility: higher accuracy using Bayesian statistics and optimized residue substitution classes. *Proteins* 25, 38–47. doi: 10.1002/(SICI)1097-0134(199605)25:1<38::AID-PROT4>3.0.CO;2-G

Tien, M. Z., Meyer, A. G., Sydykova, D. K., Spielman, S. J., and Wilke, C. O. (2013). Maximum allowed solvent accessibilites of residues in proteins. *PLoS ONE* 8:e80635. doi: 10.1371/journal.pone.0080635

Wu, W., Wang, Z., Cong, P., and Li, T. (2017). Accurate prediction of protein relative solvent accessibility using a balanced model. *BioData Min.* 10:1. doi: 10.1186/s13040-016-0121-5

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Computational and Structural Biotechnology Journal 19 (2021) 5791–5799

COMPUTATIONAL AND STRUCTURAL BIOTECHNOLOGY JOURNAL

journal homepage: www.elsevier.com/locate/csbj

DeepREx-WS: A web server for characterising protein–solvent interaction starting from sequence

Matteo Manfredi [a], Castrense Savojardo [a], Pier Luigi Martelli [a,*], Rita Casadio [a,b]

[a] Biocomputing Group, Department of Pharmacy and Biotechnology, University of Bologna, Bologna, Italy
[b] Institute of Biomembranes, Bioenergetics and Molecular Biotechnologies (IBIOM), Italian National Research Council (CNR), Bari, Italy

ARTICLE INFO

Article history:
Received 5 August 2021
Received in revised form 7 October 2021
Accepted 7 October 2021
Available online 13 October 2021

Keywords:
Residue solvent accessibility
Deep Learning
Protein flexibility
Protein disorder
Surface engineering

ABSTRACT

Protein–solvent interaction provides important features for protein surface engineering when the structure is absent or partially solved. Presently, we can integrate the notion of solvent exposed/buried residues with that of their flexibility and intrinsic disorder to highlight regions where mutations may increase or decrease protein stability in order to modify proteins for biotechnological reasons, while preserving their functional integrity. Here we describe a web server, which provides the unique possibility of integrating knowledge of solvent and non-solvent exposure with that of residue conservation, flexibility and disorder of a protein sequence, for a better understanding of which regions are relevant for protein integrity. The core of the webserver is DeepREx, a novel deep learning-based tool that classifies each residue in the sequence as buried or exposed. DeepREx is trained on a high-quality, non-redundant dataset derived from the Protein Data Bank comprising 2332 monomeric protein chains and benchmarked on a blind test set including 200 protein sequences unrelated with the training set. Results show that DeepREx performs at the state-of-the-art in the field. In turn, the Web Server, DeepREx-WS, supplements the predictions of DeepREx with features that allow a better characterisation of exposed and buried regions: i) residue conservation derived from multiple sequence alignment; ii) local sequence hydrophobicity; iii) residue flexibility computed with MEDUSA; iv) a predictor of secondary structure; v) the presence of disordered regions as derived from MobiDB-Lite3.0. The web server allows browsing, selecting and intersecting the different features. We demonstrate a possible application of the DeepREx-WS for assisting the identification of residues to be variated in protein surface engineering processes.

1. Introduction

Knowledge of the exposure of a residue in the context of a folded protein allows defining the protein folding core and identifying residues that interact with the solvent and other molecules in physiological or artificial environments [1]. Solvent exposure is routinely measured by residue Solvent Accessible Surface Area (SASA) or its Relative Solvent Accessibility (RSA), in which the maximum surface area for each residue type is the normalizing factor [2–4]. Residues in any protein can be therefore classified as buried or exposed by defining a threshold on the RSA value, routinely set equal to 20%. Programs like DSSP [5] or PSAIA [6] estimate RSA starting from the Protein Data Bank (PDB) coordinates of a protein structure. When the three-dimensional structure of a protein is not or partially available, computational methods can predict solvent exposure from the protein sequence.

Different prediction tools, mainly based on machine-learning approaches, provide RSA estimation, classifying residues into buried or exposed [7–9]. Finer-grained predictions into three or four classes of solvent exposure are possible [10]. Recently, solvent exposure is computed with deep-learning approaches [10,11].

New developments in the protein structure prediction field led to the release of AlphaFold2 [12], a very powerful deep-learning based tool for the ab-initio prediction of protein three-dimensional (3D) structure from sequence. AlphaFold2 optimally scored in the most recent edition of the Critical Assessment of Structure Prediction (CASP, predictioncenter.org), although the accuracy is not uniform across all CASP target categories and still limited on difficult targets (e.g., the free-modelling ones). Despite the success of AlphaFold2, the availability of sequence-based predictors of protein features, like solvent exposure, are still important for many reasons. Accurate predictions of protein features

* Corresponding author.
E-mail address: pierluigi.martelli@unibo.it (P.L. Martelli).

https://doi.org/10.1016/j.csbj.2021.10.016

can be useful to validate models generated with AlphaFold2 (or with others ab-initio methods), particularly in those regions where the models are expected to be low quality. Moreover, predictions of solvent exposure can be helpful also in the perspective of being integrated into end-to-end deep-learning methods, even during the learning phase, to guide and refine the training process. Tools like AlphaFold2 are very demanding in terms of computational resources, whereas simple predictors of protein structural features can be easily adopted in the presence of time/resource constraints for the preliminary structural/functional characterization of large datasets of proteins. This allows the quick identification of interesting cases on which focusing the attention and, possibly, applying more sophisticated (and computationally demanding) approaches.

Computation of solvent exposure provides valuable information in different problems, which include defining constraints for *ab-initio* protein structure prediction tools, refining protein–protein interface predictors [13,14], and structurally and functionally characterizing sequence positions, which undergo pathogenic single-residue variations [15–17]. In biotechnological applications, knowledge of residue solvent exposure is of prominent importance. Rational surface engineering i.e., the chemical modification of key positions on the protein surface, is an effective tool for tailoring protein features to specific industrial and biotechnological demands [18,19] and references therein]. Applications of protein surface (re-)engineering include the improvement of protein solubility in different solvents [20,21], immobilization [22], and stabilization in aqueous or organic solvents [23,24]. In all these applications, computational prediction of protein solvent accessibility from sequence can provide constrains for screening the candidate sites to be considered for modifications when the experimental protein three-dimensional structure (or a validated structural model) is not available [19]. Other features, such as residue conservation in multiple sequence alignment, local protein flexibility, protein secondary structure and possibly the presence of intrinsically disordered regions can further reduce the search space, identifying residues not essential for protein function and/or located in external loops.

Here, we present DeepREx-WS, a web server providing a multi-dimensional characterization of exposed and buried positions of a protein starting from its residue sequence. A two-class prediction of protein solvent exposure is provided with a novel deep learning-based method, DeepREx. The new predictor described in this paper has been trained and tested on high-quality structures extracted from the PDB and performs at the state-of-the-art, when benchmarked against other methods available for the same task.

The server DeepREx-WS, for each position, supplements the exposure prediction of DeepREx with the Kyte-Doolittle hydrophobicity and residue conservation obtained from a multiple sequence alignment. Furthermore, three external resources, MEDUSA [25], PYTHIA [26] and MobiDB-Lite3.0 [27], are present to estimate, for each residue position, protein flexibility, protein secondary structure and the presence of intrinsically disordered regions, respectively.

We release DeepREx as both Python stand-alone program and Docker image.

2. Material and methods

2.1. DeepREx implementation

2.1.1. Datasets

DeepREx is trained and tested on a dataset extracted from the Protein Data Bank (PDB) [28] (accessed Oct 15, 2019), which includes 692,646 residues from 2532 non-redundant, monomeric proteins with an X-ray crystallographic structure at a resolution $\leq$ 2.5 Å and a coverage $\geq$ 70% of the corresponding UniProt sequence [29]. Mapping between PDB and UniProt sequences was retrieved with SIFTS [30]. Membrane proteins were excluded via a cross-check on the Orientations of Proteins in Membranes (OPM) database [31].

All proteins are declared by authors of the crystallography to be functional as monomers. The dataset was reduced by similarity, so that all protein sequences share $\leq$ 30% pairwise identity. The clustering and representative sequence selection have been performed using the MMseqs2 program [32]. Specifically, we used cluster mode 1 (single-linkage clustering) and 30% sequence identity threshold. No threshold has been set for coverage, allowing to cluster also sequences with very local sequence similarity. More details on the dataset collection are available in Supplementary Materials.

The absolute Solvent Accessible Surface Area (SASA) of each residue in the PDB file is computed using DSSP [5]. Relative Solvent Accessibility (RSA) values are then obtained dividing absolute SASA values by residue-specific maximal accessibility values, as extracted from the Sander and Rost scale [2]. Finally, each residue is classified as buried (B) if its RSA is $\leq$ 20%, and exposed (E) otherwise. This threshold divides the set of residues into two almost equally sized subsets, with 52% buried and 48% exposed residues and therefore provides a balanced dataset for training and testing.

The non-redundant dataset was then randomly split into a training set, comprising 2332 sequences, and a blind test set including 200 sequences. Proteins in the training set were further split into 10 equally sized sets for cross validation.

The blind test set includes 200 protein sequences (and their structures) from different organisms: 124 monomeric proteins from Bacteria, 56 from Eukaryotes, 15 from Archaea and 5 from Viruses. Moreover, these proteins cover a wide range of 3D SCOP/CATH [33,34] classes including 30 all-alpha proteins, 37 all-beta, 84 alpha/beta (a/b) and 16 alpha + beta (a + b) (32 proteins are unclassified). Overall, the 200 protein sequences contain 56,206 residues, 29,068 and 27,138 of which are buried and exposed, respectively, in the experimental 3D structure (for details, refer to Supplementary Table 1S).

Finally, we performed an additional comparative benchmark using 9 targets from the CASP14 experiment and previously used in literature for the evaluation of sequence-based prediction of protein features [26]. In particular, the chosen targets belong to the free modelling category i.e., no homologous sequences can be found for them and for this reason they are particularly challenging for structure prediction.

2.1.2. Input encoding

DeepREx is trained on 71 features, encoding for each position the protein sequence and information derived from Multiple Sequence Alignments (MSA).

MSA for each sequence in our dataset is generated with HHblits version 3 [35], setting two iterations and default parameters. Search is executed against the Uniclust30 database [36]. HHblits provides MSA and Hidden Markov Models (HMMs) adopted to guide the search of related sequences and from which we derived some of the features.

The 71-valued vector encoding each position *i* includes:

- The canonical residue one-hot encoding, representing primary-sequence information and accounting for 20 values.
- The protein sequence profile, computed from MSA and consisting of 21 values that account for the relative frequencies of each residue type (plus the gap) in the corresponding aligned position of the MSA.
- The HMM emission probabilities obtained from the match state in position *i* (20 values).

- The HMM transition probabilities (7 values), corresponding to all possible transitions between HMM states in position *i*.
- The 3 values of Neff_Match, Neff_Insertion and Neff_Deletion [35] computed by HHblits and encoding for the MSA local diversity around position *i*. These values provide the number of effective sequences (i.e., a sequence diversity estimation) for the subalignments comprising sequences having a match, an insertion and a deletion at position *i* of the full alignment, respectively.

2.1.3. The deep-learning architecture

Fig. 1 shows the deep architecture implemented in DeepREx.

Each sequence in the dataset is encoded as a *L*x71 matrix, where *L* is the protein length and 71 is the dimension of the encoding, as detailed in the previous section.

This input is firstly processed by three cascading Bidirectional Long-Short Term Memory (BLSTM) layers [37]. BLSTMs belong to the class of Long-Short Term Memory (LSTM) networks [38], a special recurrent neural network architecture well-suited for processing sequence data (e.g., protein sequences) and extracting relevant relations between elements of the sequence. Moreover, LSTMs have several advantages over traditional recurrent architectures in terms of stability of training and the proper handling of the vanishing gradient problem [39]. BLSTMs perform a double scanning of the input sequence, from left to right and vice versa, in order to better capture the sequential relations among sequence positions. Here, each BLSTM layer includes 32 activation units.

The output of the third recurrent layer is then provided as input to a time-distributed, fully-connected layer adopting a sigmoid activation function. This layer provides the final, binary classification of each residue in the sequence into buried or exposed classes. It computes a numerical output in the range [0,1] for each residue that can be interpreted as a probability for the residue to be exposed: all residues with $p \geq 0.5$ are predicted as exposed while those with $p < 0.5$ are classified as buried.

The method has been implemented with the Keras deep-learning Python library [40]. The total number of trainable parameters in the model is 76,353.

The output value *o* has been used to estimate the reliability index (RI) of the prediction:

$$RI = 2 \times |o - 0.5| \quad (1)$$

If *o* is close to 0.5 (uncertain classification), RI is close to 0. If *o* is close to 0 (strong classification in the buried class) or 1 (strong classification in the exposed class), RI is close to 1.

2.1.4. DeepREx training and evaluation

Training is performed by adopting a 10-fold cross-validation procedure, using 8 sets for training, one set for validation and early stopping (to avoid overfitting), and one for testing. Cross-validation results are reported as the average over performances computed on the testing sets. This training phase sets the optimal values of the architecture hyperparameters. Each model is trained for at most 1000 epochs. An early stopping procedure is adopted to reduce overfitting: the training procedure is stopped after 50 consecutive epochs when the error computed on the validation set does not decrease. The presence of sequences of variable length is handled using mini-batches of 64 sequences and zero-padding each sequence in the batch to the same length (i.e., the maximal length in the mini-batch). A masking layer, placed after the input layer, is used to ignore padded values. The ADAM optimizer [41] is adopted for gradient descent on the binary cross-entropy loss function. We run several complete cross-validations to select the optimal set of hyperparameters (number of activation units in LSTM layers, minibatch size, ADAM optimizer parameters). We chose the set of hyperparameters maximizing the performance of the method on the cross-validation validation sets.

Once the hyperparameters are fixed, the final DeepREx model for testing the blind set is obtained after training over the whole training set with the routinary procedure: 9/10 subsets are for

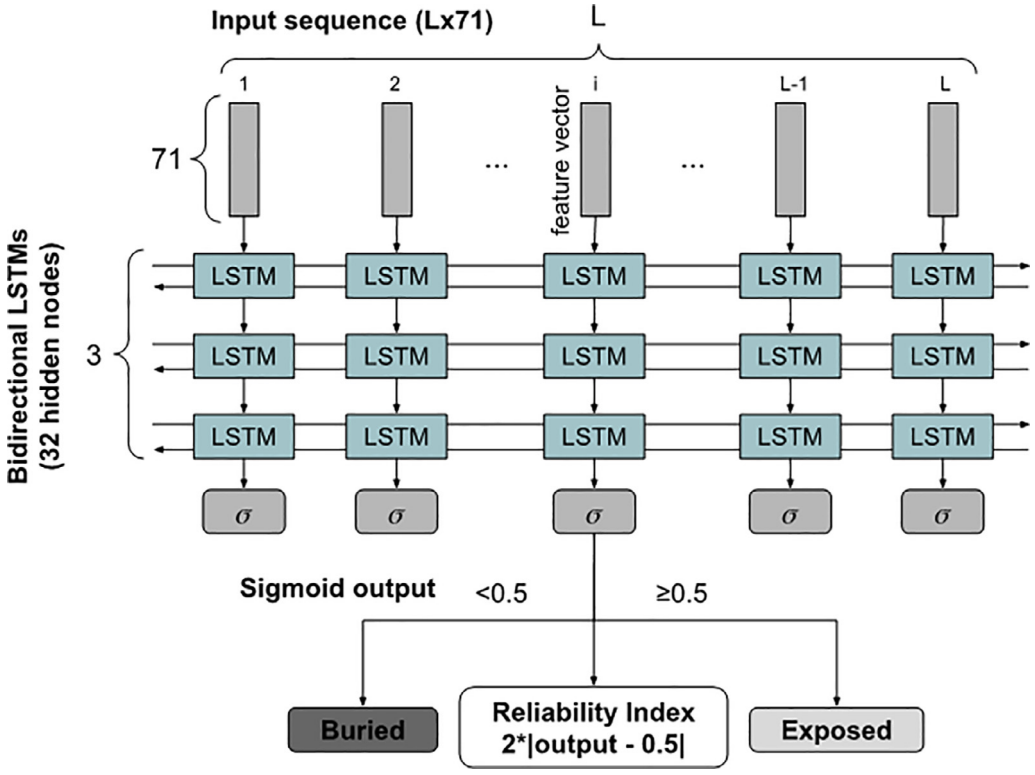


Fig. 1. Architecture of the deep neural network implemented in DeepREx to predict residue solvent exposure.

the actual training, while one random set among the 10 is adopted as validation set for early stopping. This final model is then tested on the 200 proteins of the blind test set and excluded from the training set to evaluate its performance.

2.1.5. Scoring indexes

The performance of the binary solvent accessibility classifiers is assessed with the following standard scores. Without loss of generality, we assume the exposed (E) and the buried (B) classes to be the positive and negative classes, respectively. In what follows, TP, TN, FP and FN are true positive, true negative, false positive and false negative predictions, respectively. The following scoring measures are computed:

- Accuracy (Q2), defined as:

$$Q_2 = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

- Precision:

$$Prec = \frac{TP}{TP + FP} \tag{4}$$

- Recall:

$$Rec = \frac{TP}{TP + FN} \tag{5}$$

- F1, the harmonic mean of the precision and recall, defined as:

$$F1 = \frac{2 * Prec * Rec}{Prec + Rec} \tag{6}$$

- Matthews Correlation Coefficient (MCC), defined as:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{7}$$

2.2. The Web Server (DeepREx-WS) implementation

DeepREx-WS integrates DeepREx predictions with external resources. We include predictions obtained with MEDUSA [25], estimating residue flexibility of the proteins across five classes (0 = rigid, 4 = flexible). MEDUSA is based on a deep convolutional neural network architecture processing an input comprising evolutionary information, derived from MSAs and residue physicochemical properties [25].

We provide secondary structure prediction by means of PYTHIA, a protein local conformation prediction tool [26]. Specifically, PYTHIA (released in 2021, [26]) can be easily integrated in our web server, being released as a docker container. Furthermore, it runs fast, and it takes multiple sequence alignments as inputs. It is designed to predict local conformation in terms of Protein Blocks (PB). Overall, 16 PDB classes (labelled with lower-case letters, from *a* to *p*) are provided by PYTHIA: PB labels *a*, *b*, *c*, *d*, *e* and *f* represent different beta-strand regions (*c* is for the core of strand, *a*, *b* and *d*, *e* for N- anc C-terminal caps, respectively), PB labels *g*, *h*, *i* and *j* are all representing random coils while labels *k*, *l*, *m*, *n*, *o* and *p* map into alpha-helices (*m* for the helix core, *k, l* and *n, o* for N- and C-terminal caps, respectively). Here we mapped PB to secondary structure as follows: *c* to beta-strand (E), *m* to alpha-helix (H) and the remaining labels to random coil (C).

We integrate intrinsically disordered regions as predicted with MobiDB-Lite3.0 [27], providing a binary prediction for each residue (disordered/structured). MobiDB-Lite3.0 computes a consensus derived from the outputs of eight different predictors of disordered regions and applies a filtering procedure to get rid of spurious disorder predictions. All the three methods have been downloaded and are executed in-house.

Finally, DeepREx-WS also includes for each residue a hydrophobicity index, computed by averaging the Kyte-Doolittle hydropathy scale [42] over a window of 5 residues, and a conservation index computed from the MSA with the following equation:

$$CI(i) = 1.0 - \left(-\frac{1}{\log(20)} \sum_{a=1}^{20} f_a(i) \times \log\left[f_a(i) \right] \right) \tag{2}$$

where $f_a(i)$ is the frequency of the residue type a in the position i of the MSA. The CI ranges between 0 (not conserved) and 1 (fully conserved). The MSA used for computing the CI is the same provided in input to the DeepREx predictor and built for the input sequence using HHblits as detailed in section 2.1.2. The CI is only computed for MSA positions having at most 70% of gaps in the aligned column. For position with more than 70% gaps a default conservation of 0 is reported.

The web server is implemented using the Python Django application server (version 2.2.5), Apache (version 2) and Postgresql (version 11). The user interface is designed using Bootstrap (version 4), DataTable (version 1.10.22), the neXtProt feature viewer (version 1.0, https://github.com/calipho-sib/feature-viewer) and custom JavaScript-based validators for input data.

3. Results

3.1. Performance of the solvent accessibility DeepREx prediction

3.1.1. Cross-validation and blind test performance

DeepREx performance is scored using a 10-fold cross-validation procedure on our training dataset comprising 2332 proteins sequences and a blind set with 200 protein sequences, compiled to be non-redundant with respect to our training dataset. Results are reported in Table 1. DeepREx is quite robust, achieving similar performances in the two validation procedures. Overall, our method discriminates buried from exposed residues with 82% accuracy, 82% F1 and 0.63 MCC.

We further compared DeepREx with two recent state-of-the-art tools, both based on deep-learning approaches: PaleAle5 [10] and NetSurfP-2.0 [11]. PaleAle5, predicts exposure into 4 classes: E (exposed), e (partially exposed), b (partially buried) and B (buried). The threshold used by PaleAle5 authors to separate exposed (either E or e) from buried (either b or B) residues is 25% RSA, very close to the threshold adopted in this work. NetSurfP-2.0 directly predicts

Table 1
DeepREx performance in a 10-fold cross-validation and on the blind test set.

Scoring index	Cross-validation	Blind test
Precision	0.820 ± 0.002	0.82
Recall	0.800 ± 0.001	0.80
F1	0.810 ± 0.001	0.82
Q2	0.810 ± 0.001	0.82
MCC	0.620 ± 0.002	0.63

For index definition see section 2.1.5.

RSA real values: in this case we used our 20% RSA threshold to transform these values into a binary classification.

Comparative results on the blind test and on the CASP14 dataset are reported in Table 2. We should remark that the blind test set may not be blind for the other methods. Remarkably, all methods achieve a similar performance on both testing sets. DeepREx reports the most balanced results in the blind test set, as shown by the close values of precision and recall. When tested on the CASP14 dataset comprising 9 free-modelling targets, performances of all methods drop to lower values. The 9 targets are difficult to predict since they do belong to the free-modelling CASP category, without or with very few homologous in the data base. Nonetheless, the three approaches seem to have very close performances, as highlighted by the only small differences in the MCC values.

The three methods (DeepREx, PaleAle5 and NetSurfP-2) are all based on similar neural network architectures involving LSTMs and/or convolutional layers. Among the three, DeepREx adopts the simplest architecture, with only three cascading BiLSTM layers. This ensures the lowest number of parameters for the resulting model without affecting prediction performances that are comparable among the three approaches.

Differently from the other two methods, our DeepREx predictor has been trained on functional monomeric protein chains. This allows to properly define solvent exposure in physiological conditions and to avoid the introduction of biases in solvent exposure computation due to conformational changes at the interfaces upon protein complex formation. However, training only on monomers does not limit the adoption of our model for predicting solvent exposure of multimeric protein chains. To prove this, we performed an additional experiment testing DeepREx on a set of 984 multimeric protein chains extracted from the PaleAle5 independent dataset [10]. In this test, we registered only a slight decrease in the accuracy. The performances of both methods are listed in Table 2S (Supplementary Materials). This suggests that the exclusion of multimeric chains from our training dataset has a very limited impact on the overall performance of DeepREx.

Finally, a reliability index (RI) can be associated to each prediction by applying Eq. (1). RI close to 0 indicates a prediction output close to 0.5 while RI close to 1 indicates that the output is close to 0 (buried) or 1 (exposed). We performed tests to assess whether the RI value can be adopted to discriminate accurate from poor predictions. Results are reported in Supplementary Table 3S and indicate that the higher the RI value the most accurate is the prediction. Notably, most predictions have RI values higher than 0.6. Predictions with low RI values (<0.2) mostly pertain to proteins with very few sequences in the corresponding MSA and, therefore, with a poor input information.

3.2. The web server: DeepREx-WS

DeepREx-WS is available at https://deeprex.biocomp.unibo.it. The server input interface accepts a single sequence in FASTA format with length ranging between 50 and 5000 residues. Upon submission the user is redirected to the page where results will be available after job completion. This page automatically refreshes every 60 s and shows to the user the current status of the job (queued or running). The server also provides the user with a universal job identifier, which can be thereafter used to retrieve job results. The result page (Fig. 2) provides information about the job, including i) the identifier, ii) submission and completion time, iii) protein ID, iv) protein length and v) counts of buried and exposed predictions. After that, the output of the predictor is shown using an interactive viewer along the submitted protein sequence as well as in tabular format.

The following information is reported both in track and tabular form:

i) DeepREx output as two-class prediction of solvent exposure (E = exposed, B = buried).
ii) The RI associated to the DeepREx prediction.
iii) The Kyte-Doolittle hydropathy score [42], averaged over a window of five residues.
iv) The conservation index computed as in Equation (2).
v) The three-class prediction of secondary structure by PYTHIA [26].
vi) The five-class flexibility prediction provided by MEDUSA (0 = rigid, 4 = flexible) [25].
vii) The two-class prediction of intrinsically disordered regions provided by MobiDB-lite3.0 (S = structured, D = disordered) [27].

The feature viewer allows to navigate the sequence, visualizing the different predicted features along it. The user can zoom to specific regions and export a picture of the current visualization in PNG format.

Tabular data can be sorted according to any one of the reported outputs. Moreover, users can activate and combine filters for residue type, exposed or buried positions, reliability index, conservation index, flexibility level, secondary structure and disordered regions.

All results can be downloaded in Tab-Separated Values (TSV) format. If one or more filters are active, the downloaded TSV will report only results for selected residues.

3.3. DeepREx-WS output features

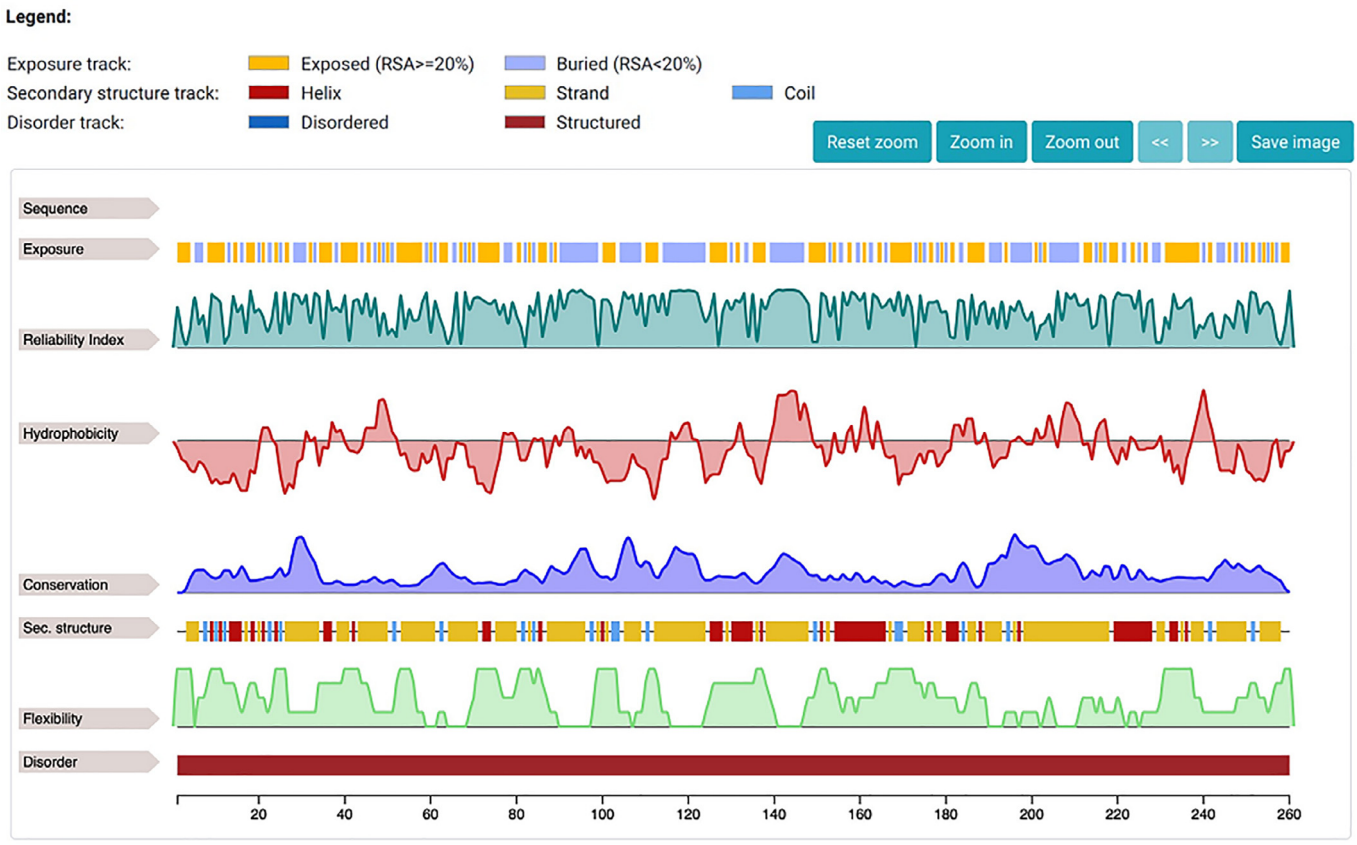
In this section we analyze the relation between solvent exposure and other features included in the DeepREx-WS output, comprising, as detailed above, hydrophobicity (Kyte-Doolittle), conservation index from MSA, flexibility (MEDUSA [25]), secondary structure (PYTHIA [26]) and disorder (MobiDB-Lite3.0 [27]).

All the correlation analyses (except for protein disorder) were performed on the 200 protein sequences included in our blind test (Table 3). Overall, the 200 proteins contain 56,206 residues, 29,068 and 27,138 of which are buried and exposed, respectively, in their experimental 3D structure. On this set DeepREx performs quite well, achieving a prediction accuracy of 82% and a MCC of 0.63

Table 2
Comparison of DeepREx and other protein solvent accessibility predictors on the blind test set and CASP14 targets.

Method	Dataset	Precision	Recall	F1	Q2	MCC
DeepREx	BlindTest	0.82	0.80	0.82	0.82	0.63
PaleAle5.0 [10]	BlindTest	0.78	0.85	0.82	0.82	0.65
NetSurfP2.0 [11]	BlindTest	0.92	0.77	0.82	0.83	0.66
DeepREx	CASP14	0.87	0.76	0.81	0.79	0.57
PaleAle5.0 [10]	CASP14	0.90	0.72	0.80	0.78	0.58
NetSurfP2.0 [11]	CASP14	0.81	0.89	0.85	0.81	0.59

For index definition see section 2.1.5.



Position	Residue	Exposure	Reliability Index	Hydrophobicity	Conservation	Flexibility	Disorder	Secondary structure
1	M	Exposed	0.7	-0.42	0.0	4	S	-
2	S	Exposed	0.28	-1.06	0.0	4	S	-
3	H	Exposed	0.05	-1.24	0.04369	4	S	E
4	H	Exposed	0.33	-1.7	0.16277	4	S	E
5	W	Buried	0.86	-1.8	0.24434	0	S	E
6	G	Buried	0.41	-1.24	0.25717	2	S	E
7	Y	Buried	0.55	-1.38	0.25732	2	S	C
8	G	Exposed	0.1	-1.84	0.21627	3	S	C
9	K	Exposed	0.92	-2.46	0.17615	4	S	H
10	H	Exposed	0.86	-2.28	0.16197	4	S	C
11	N	Exposed	0.78	-2.52	0.18599	4	S	H
12	G	Exposed	0.08	-2.44	0.2511	3	S	C
13	P	Buried	0.22	-2.44	0.25001	3	S	H
14	E	Exposed	0.89	-1.92	0.1766	3	S	H
15	H	Exposed	0.62	-2.48	0.21121	2	S	H
16	W	Buried	0.73	-2.94	0.29127	1	S	H
17	H	Exposed	0.26	-2.94	0.21777	2	S	E

Fig. 2. A screenshot of the DeepREx-WS result page.

(Table 2). The 200 proteins have a negligible disorder content according to MobiDB (less than 1%).

For the evaluation of the correlation between exposure and disorder we collected a dataset of 88 human proteins extracted from the DisProt database [43] and endowed with a disorder content ranging from 10% to 30%. We only compute correlation with respect to predicted exposure, since for disordered regions which, by definition, lack PDB structures, we cannot compute real solvent accessibility.

For what concerns secondary structure predictions, we report three different correlations between exposure and alpha-helix, beta-strand and coil predicted content, respectively.

Table 3
Pairwise Pearson's Correlation Coefficients (PCC) between predicted solvent exposure and the other features.

Feature	PCC with real solvent exposure [(a)]	PCC with predicted solvent exposure [(a)]
Flexibility (MEDUSA [25])	0.56±0.06	0.58±0.08
Alpha-helix (PYTHIA [26])	−0.10±0.10	−0.11±0.11
Beta-strand (PYTHIA [26])	−0.20±0.10	−0.21±0.10
Coil (PYTHIA [26])	0.24±0.08	0.25±0.08
Conservation from MSA	−0.37±0.11	−0.39±0.11
Hydrophobicity (Kyte-Doolittle [42])	−0.23±0.09	−0.24±0.10
Disorder (MobiDB-Lite3.0 [27]) [(b)]	–	0.27±0.11

[(a)] Average PCC computed per-protein and associated Standard Deviation values.
[(b)] Correlation computed on 88 proteins from DisProt [43] with disorder content ranging from 10% to 30%.

All correlation results are shown in Table 3 and are calculated per protein and then averaged.

Residue flexibility as predicted by MEDUSA well correlates with both real and predicted solvent exposure values (in Table 3, first line, average PCCs are 0.58 and 0.56, respectively). This can be partially explained by considering that MEDUSA adopts crystallographic B-factors as proxies for residue flexibility, and that these values tend to be higher at the protein surface. However, the correlation is not perfect, suggesting that the two features (i.e., residue solvent accessibility and flexibility) provide complementary information which can be profitably merged for a better understanding of residue structural properties from sequence.

Average correlation coefficients between exposure and helix and strand motifs are negative and close to 0, considering the significant deviations from the mean (in Table 3, second and third lines, respectively). This may indicate that exposed residues (both real and predicted) are not preferentially placed in helix or strand regions. Correlations with coils are slightly positive (in Table 3, fourth line), suggesting a weak propensity of exposed residues for coil regions.

Exposed residues (either real or predicted) tend to be localized in non-conserved positions, as highlighted by moderate anti-correlation reported in Table 3 between predicted and real solvent accessibility and conservation index (fifth line, average PCCs are −0.39 and −0.37, respectively). Moreover, as expected, solvent exposure anti-correlates with respect to hydrophobicity (in Table 3, sixth line, average PCCs are −0.24 and −0.23). Again, these results suggest that solvent accessibility cannot be completely explained by conservation or residue hydrophobicity alone, justifying the integration/combination of the different features for residue structural/functional characterization.

Finally, a modest correlation (PCC = 0.26) of exposure is also observed with protein disorder on a dataset of 88 proteins extracted from DisProt [43]. This may indicate a slight propensity of disordered regions for exposed positions.

Although the size of our protein sets is limited, the results presented in this section suggest that protein solvent exposure positively correlates with protein flexibility and negatively correlates with hydrophobicity and conservation. In general, all these features provide complementary information on residues and can be then combined to characterize proteins from a structural and functional point of view. This can be useful in many contexts such as protein surface engineering, where one looks for residues placed at the protein surface to be selected as candidate for site-specific mutagenesis. Routinely, selected positions are exposed residues characterized by low conservation indexes (in order to avoid functionally important sites) and placed in flexible loops. Starting from protein sequence, the combination of predicted exposure, flexibility and conservation can be helpful to reduce the search space in protein surface engineering. For instance, in our dataset of 200 proteins, selecting residues predicted as exposed, having a low conservation index (residue conservation lower than the median for each protein) and flexible (MEDUSA value ≥ 3) we obtain 12,068 residues, representing 21% of the total number of residues. This allows to significantly restrict the search space of candidate positions for surface engineering particularly when 3D structure is lacking.

3.4. Case study: DeepREx-WS to assist surface engineering

In this section, we benchmark DeepREx-WS in the context of protein surface charge engineering with an example. Surface charge engineering is particularly important for the industrial use of biocatalyst. Recently, much attention has been focused on halophilic enzymes that can be adopted in hypersaline environments (e.g., brines, ionic liquids or ionic detergents) [21]. Putative enzymes for the use in high-salt conditions have been traditionally identified among those available in natural systems. An alternative approach consists in the induction of halotolerance into an existing biocatalyst possessing the required features in terms of catalytic activity. Following this trend, in a recent study [21], authors considered the bovine carbonic anhydrase II (bCAII, UniProtKB: P00921) for the rational design of halotolerance by protein surface engineering. Specifically, in order to enhance bCAII halotolerance, authors adopted one of the possible mechanisms present in natural halophilic enzymes: the increase of the abundance of acidic residues in the protein surface. By this, 18 positions were identified and mutated into negative residues, after a rational choice procedure based on the available PDB bCAII structure (1V9E). The selection of positions to be mutated is not exhaustive and integrates considerations on solvent accessibility and/or side-chain steric bulks, and on the residue conservation in a multiple sequence alignment generated using 50 homologous sequences. The availability of the three-dimensional structure provides a large amount of information. However, what if the structure is not available as it is for many proteins? DeepREx-WS can assist the choice of residues to be mutated without the help of the structure. We submitted the 260-residue long sequence of the bCAII to the server and filtered the results to select possible positions for mutation into negative residues (Glutamic or Aspartic acid). Remarkably, the exposure prediction reaches a high MCC value (0.81). Mimicking the rational procedure described in [21] and considering the DeepREx-WS output for the whole protein sequence, we can select residues predicted as exposed, obtaining 139 positions, 112 of which are different from Glutamic or Aspartic acid, and then reducing the search space to 43% of the protein residues. All the 18 positions from [21] are included in this set. If we add a filter on protein conservation, selecting only lowly conserved residues (CI lower than the median on the protein equal to 0.2), we can further restrict to 78 possible target positions (30% of the sequence). Out of the 18 positions considered in [21], 13 are included in the set of 78 positions selected. Five out of 18 positions are not retained in our selection. Two of them (G8 and N24) have a conservation index (0.22) only slightly higher than the threshold used here (0.20). The remaining 3 positions (N62, N252 and Q254) are weakly variable in the MSA used in [21] and their selection in the study does not take into consideration conservation.

If exposed positions are intersected with most flexible ones (MEDUSA score equal to 3 or 4), 66 positions are selected, corresponding to 25% of the sequence. This set contains 12 out of 18 positions selected in [21]. Out of the 6 not included positions, 3 are predicted with a medium flexibility level (MEDUSA score equal to 2) and 3 are predicted with limited flexibility (MEDUSA score 1). Remarkably, none of them are predicted as rigid (MEDUSA score 0).

In Table 4 we report the complete output of DeepREx-WS for the 18 positions of interest reported in [20]. Interestingly, all the

Table 4
Analysis of relevant positions of the bovine carbonic anhydrase II protein (UniProtKB:P00921) reported in [21] with the DeepREx-WSs.

Pos	Res	SE[(a)]	RI[(b)]	HP[(c)]	CI[(d)]	Flexibility[(e)]	Disorder[(f)]
8	G	E	0.10	−1.84	0.22	3	S
18	K	E	0.98	−1.74	0.13	3	S
24	N	E	0.95	−0.22	0.22	4	S
36	K	E	0.98	−0.42	0.10	3	S
39	V	E	0.62	0.64	0.12	3	S
50	V	E	0.23	1.62	0.13	1	S
57	R	E	0.48	−1.72	0.09	1	S
62	N	E	0.54	−1.28	0.32	1	S
74	Q	E	0.76	−3.04	0.12	4	S
85	T	E	0.95	0.08	0.16	4	S
136	Q	E	0.11	−2.06	0.11	4	S
169	K	E	0.96	−2.56	0.1	4	S
177	N	E	0.78	−0.6	0.13	3	S
186	N	E	0.91	1.34	0.14	3	S
220	Q	E	0.95	−1.34	0.18	2	S
238	L	E	0.17	0.88	0.16	2	S
252	N	E	0.94	−2.32	0.28	3	S
254	Q	E	0.75	−2.36	0.25	2	S

(a) SE = Solvent Exposure, as predicted by DeepREx. E = Exposed, B = Buried.
(b) RI = DeepREx Reliability Index, as defined in Eq. (1).
(c) HP = Kyte-Doolittle Hydrophobicity [42].
(d) CI = Conservation Index, computed as in Eq. (2).
(e) Flexibility value, as predicted by Medusa [25]. It goes from 0 (rigid) to 4 (highly flexible).
(f) Disorder annotation as retrieved from MobiDB-Lite3.0 [27]. S = Structured, D = Disordered.

positions are correctly predicted as exposed, most of them with high reliability. Moreover, they are all characterized by a low conservation index (between 0.09 and 0.32), while most of them (12 out of 18) are predicted as localized in flexible regions (MEDUSA ≥ 3). Altogether, these features are in line with those required by the rational design performed in [21] and show that the DeepREx-WS prediction can reconstruct them starting from the protein sequence alone.

4. Conclusion

In this paper, we develop DeepREx, a novel deep-learning based tool for annotating residue solvent exposure into two classes (buried and exposed). DeepREx performance is evaluated on a blind dataset comprising 200 proteins and on a selected set of difficult targets from CASP14. Results show that DeepREx is competitive with other tools at the state-of-the-art. The method is made available as a web server (DeepREx-WS) and as a standalone tool, including a containerized version. This makes DeepREx well-suited for applications on large datasets and for easy integration into higher-level workflows. The web server which integrates the predictor of solvent accessibility (DeepREx-WS) is implemented to allow the intersection of DeepREx outputs with other protein features such as residue flexibility, conservation, hydrophobicity and inclusion in intrinsically disordered regions. Our results on 200 proteins indicate that solvent accessibility well correlates with flexibility and negatively correlates with conservation and hydrophobicity. Disorder is apparently negligible for this analysis. Furthermore, with the example of the bovine carbonic anhydrase II [21] and comparing with residue selection done directly on the protein structure, we confirm that the integration of the server outputs can profitably allow a primary selection of candidate positions for surface residue modification starting from the protein sequence alone. We propose our web server to highlight likely positions in protein sequence for surface engineering and as a valuable alternative when protein structure is not or partially available.

5. Data and method availability

The DeepREx web server and datasets are available at https://deeprex.biocomp.unibo.it.

The DeepREx standalone tool Python source code is available at https://github.com/BolognaBiocomp/deeprex. The program has been tested with Python version 3.8. External dependencies include the Biopython package (tested version 1.78), the Keras (tested version 2.4.3) deep-learning library as well as a working installation of the HHsuite (tested version 3.3.0) for multiple sequence alignment building.

DeepREx has been also released as a Docker container available at https://hub.docker.com/r/bolognabiocomp/deeprex. In both cases, the program takes in input: i) a FASTA file containing one or more sequences; ii) a valid sequence database for HHblits alignments; iii) a file name where an output TSV file will be written after termination.

Funding

The work was supported by the PRIN2017 grant (project 2017483NH8_002), delivered to CS from the Italian Ministry of University and Research.

CRediT authorship contribution statement

Matteo Manfredi: Methodology, Software, Validation. **Castrense Savojardo:** Conceptualization, Data curation. **Pier Luigi Martelli:** Conceptualization, Supervision. **Rita Casadio:** Conceptualization, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.csbj.2021.10.016.

References

[1] Miller S, Lesk AM, Janin J, Chothia C. The accessible surface area and stability of oligomeric proteins. Nature 1987;328(6133):834–6. https://doi.org/10.1038/328834a0.

[2] Rost B, Sander C. Conservation and prediction of solvent accessibility in protein families. Proteins 1994;20(3):216–26. https://doi.org/10.1002/prot.340200303.

[3] Tien MZ, Meyer AG, Sydykova DK, Spielman SJ, Wilke CO, Porollo A. Maximum allowed solvent accessibilites of residues in proteins. PLoS ONE 2013;8(11):e80635. https://doi.org/10.1371/journal.pone.0080635.

[4] Rose G, Geselowitz A, Lesser G, Lee R, Zehfus M. Hydrophobicity of amino acid residues in globular proteins. Science 1985;229(4716):834–8. https://doi.org/10.1126/science.4023714.

[5] Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers 1983;22(12):2577–637. https://doi.org/10.1002/bip.360221211.

[6] Mihel J, Šikić M, Tomić S, Jeren B, Vlahoviček K. PSAIA – Protein Structure and Interaction Analyzer. BMC Struct Biol 2008;8(1):21. https://doi.org/10.1186/1472-6807-8-21.

[7] Adamczak R, Porollo A, Meller J. Combining prediction of secondary structure and solvent accessibility in proteins. Proteins 2005;59(3):467–75. https://doi.org/10.1002/prot.20441.

[8] Magnan CN, Baldi P. SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. Bioinformatics 2014;30(18):2592–7. https://doi.org/10.1093/bioinformatics/btu352.

[9] Pollastri G, Martin AJ, Mooney C, Vullo A. Accurate prediction of protein secondary structure and solvent accessibility by consensus combiners of sequence and structure information. BMC Bioinf 2007;8(1):201. https://doi.org/10.1186/1471-2105-8-201.

[10] Kaleel M, Torrisi M, Mooney C, Pollastri G. PaleAle 5.0: prediction of protein relative solvent accessibility by deep learning. Amino Acids 2019;51(9):1289–96. https://doi.org/10.1007/s00726-019-02767-6.

[11] Klausen MS, Jespersen MC, Nielsen H, Jensen KK, Jurtz VI, Sønderby CK, et al. NetSurfP-2.0: improved prediction of protein structural features by integrated deep learning. Proteins Struct Funct Bioinf 2019;87(6):520–7. https://doi.org/10.1002/prot.v87.610.1002/prot.25674.

[12] Jumper J, Evans R, Pritzel A, Green T, Figurnov M, et al. (2021) Highly accurate protein structure prediction with AlphaFold. Nature, 596(7873), 583-589. doi: 10.1038/s41586-021-03819-2. Epub 2021 Jul 15. PMID: 34265844; PMCID: PMC8371605.

[13] Savojardo C, Fariselli P, Martelli PL, Casadio R. ISPRED4: interaction sites PREDiction in protein structures with a refining grammar model. Bioinformatics 2017;33(11):1656–63. https://doi.org/10.1093/bioinformatics/btx044.

[14] Porollo A, Meller J. Prediction-based fingerprints of protein-protein interactions. Proteins 2007;66(3):630–45. https://doi.org/10.1002/prot.21248.

[15] Savojardo C, Manfredi M, Martelli PL, Casadio R. Solvent accessibility of residues undergoing pathogenic variations in humans: from protein structures to protein sequences. Front Mol Biosci 2021;7:. https://doi.org/10.3389/fmolb.2020.626363626363.

[16] Martelli PL, Fariselli P, Savojardo C, Babbi G, Aggazio F, et al. Large scale analysis of protein stability in OMIM disease related human protein variants. BMC Genomics 2016;17(S2):397. https://doi.org/10.1186/s12864-016-2726-v.

[17] Savojardo C, Babbi G, Martelli PL, Casadio R. Functional and structural features of disease-related protein variants. IJMS 2019;20(7):1530. https://doi.org/10.3390/ijms20071530.

[18] Pedersen JN, Zhou Ye, Guo Z, Pérez B. Genetic and chemical approaches for surface charge engineering of enzymes and their applicability in biocatalysis: A review. Biotechnol Bioeng 2019;116(7):1795–812. https://doi.org/10.1002/bit.v116.710.1002/bit.26979.

[19] Shivange AV, Hoeffken HW, Haefner S, Schwaneberg U. Protein consensus-based surface engineering (ProCoS): a computer-assisted method for directed protein evolution. Biotechniques 2016;61(6):305–14. https://doi.org/10.2144/000114483.

[20] Simeonov P, Berger-Hoffmann R, Hoffmann R, Strater N, Zuchner T. Surface supercharged human enteropeptidase light chain shows improved solubility and refolding yield. Protein Eng Des Sel 2011;24(3):261–8. https://doi.org/10.1093/protein/gzq104.

[21] Warden AC, Williams M, Peat TS, Seabrook SA, Newman J, Dojchinov G, et al. Rational engineering of a mesohalophilic carbonic anhydrase to an extreme halotolerant biocatalyst. Nat Commun 2015;6(1). https://doi.org/10.1038/ncomms10278.

[22] Qi Y, Chilkoti A. Protein-polymer conjugation-moving beyond PEGylation. Curr Opin Chem Biol 2015;28:181–93. https://doi.org/10.1016/j.cbpa.2015.08.009.

[23] Turunen O, Vuorio M, Fenel F, Leisola M. Engineering of multiple arginines into the Ser/Thr surface of Trichoderma reesei endo-1,4-beta-xylanase II increases the thermotolerance and shifts the pH optimum towards alkaline pH. Protein Eng 2002;15:141–5. https://doi.org/10.1093/protein/15.2.141.

[24] Takagi H, Hirai K, Maeda Y, Matsuzawa H, Nakamori S. Engineering subtilisin E for enhanced stability and activity in polar organic solvents. J Biochem 2000;127:617–25. https://doi.org/10.1093/oxfordjournals.jbchem.a022649. PMID: 10739954.

[25] Meersche YV, Cretin G, de Brevern AG, Gelly J-C, Galochkina T. MEDUSA: prediction of protein flexibility from sequence ISSN 0022-2836. J Mol Biol 2021;166882. https://doi.org/10.1016/j.jmb.2021.166882.

[26] Cretin G, Galochkina T, de Brevern AG, Gelly JC. (2021) PYTHIA: Deep learning approach for local protein conformation prediction. Int J Mol Sci., 22(16), 8831. Published 2021 Aug 17. doi:10.3390/ijms22168831

[27] Necci M, Piovesan D, Clementel D, Dosztányi Z, Tosatto SCE (2020) MobiDB-lite 3.0: fast consensus annotation of intrinsic disorder flavors in proteins, Bioinformatics, https://doi.org/10.1093/bioinformatics/btaa1045

[28] Berman HM. The Protein Data Bank. Nucleic Acids Res 2000;28(1):235–42. https://doi.org/10.1093/nar/28.1.235.

[29] UniProt Consortium. UniProt: A worldwide hub of protein knowledge. Nucleic Acids Res 2019;47(D1):D506–15. https://doi.org/10.1093/nar/gky1049.

[30] Dana JM, Gutmanas A, Tyagi N, Qi G, O'Donovan C, et al. (2019) SIFTS: updated structure integration with function, taxonomy and sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins. Nucleic Acids Res, 47 (D1), D482–D489. https://doi.org/10.1093/nar/gky1114.

[31] Lomize MA, Pogozheva ID, Joo H, Mosberg HI, Lomize AL. OPM database and PPM web server: resources for positioning of proteins in membranes. Nucleic Acids Res 2012;40(D1):D370–6. https://doi.org/10.1093/nar/gkr703.

[32] Fox NK, Brenner SE, Chandonia JM. SCOPe: Structural Classification of Proteins–extended, integrating SCOP and ASTRAL data and classification of new structures. Nucleic Acids Res 2014;42(Database issue):D304–9. https://doi.org/10.1093/nar/gkt1240.

[33] Steinegger M, Söding J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. Nat Biotechnol 2017;35(11):1026–8. https://doi.org/10.1038/nbt.3988. Epub 2017 Oct 16 PMID: 29035372.

[34] Sillitoe I, Bordin N, Dawson N, Waman VP, Ashford P, Scholes HM, et al. CATH: increased structural coverage of functional space. Nucleic Acids Res, 2021;49(D1):D266-D273. doi: 10.1093/nar/gkaa1079.

[35] Steinegger M, Meier M, Mirdita M, Vöhringer H, Haunsberger SJ, et al. HH-suite3 for fast remote homology detection and deep protein annotation. BMC Bioinf 2019;20(1):473. https://doi.org/10.1186/s12859-019-3019-7.

[36] Mirdita M, von den Driesch L, Galiez C, Martin MJ, Söding J, et al. Uniclust databases of clustered and deeply annotated protein sequences and alignments. Nucleic Acids Res 2017;45(D1):D170–6. https://doi.org/10.1093/nar/gkw1081.

[37] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks 2005;18(5–6):602–10. https://doi.org/10.1016/j.neunet.2005.06.042.

[38] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.

[39] Pascanu R, Mikolov T, Bengio Y (2013) On the difficulty of training recurrent neural networks. arXiv:1211.5063 [cs].

[40] Chollet F (2015) Keras; GitHub.

[41] Kingma DP, Ba J (2017) Adam: A method for stochastic optimization. arXiv:1412.6980 [cs].

[42] Kyte J, Doolittle RF. A simple method for displaying the hydropathic character of a protein. J Mol Biol 1982;157(1):105–32.

[43] Hatos A, Hajdu-Soltész B, Monzon AM, Palopoli N, Álvarez L, et al. DisProt: intrinsic protein disorder annotation in 2020. Nucleic Acids Res 2020;48(D1):D269–76. https://doi.org/10.1093/nar/gkz975. PMID: 31713636; PMCID: PMC7145575.

Bioinformatics, 38(23), 2022, 5168–5174
https://doi.org/10.1093/bioinformatics/btac678
Advance Access Publication Date: 13 October 2022
Original Paper

Sequence analysis

E-SNPs&GO: embedding of protein sequence and function improves the annotation of human pathogenic variants

Matteo Manfredi†, Castrense Savojardo † , Pier Luigi Martelli * and Rita Casadio

Biocomputing Group, Department of Pharmacy and Biotechnology, University of Bologna, Bologna 40126, Italy

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

Associate Editor: Valentina Boeva

Received on May 16, 2022; revised on September 14, 2022; editorial decision on October 6, 2022; accepted on October 10, 2022

Abstract

Motivation: The advent of massive DNA sequencing technologies is producing a huge number of human single-nucleotide polymorphisms occurring in protein-coding regions and possibly changing their sequences. Discriminating harmful protein variations from neutral ones is one of the crucial challenges in precision medicine. Computational tools based on artificial intelligence provide models for protein sequence encoding, bypassing database searches for evolutionary information. We leverage the new encoding schemes for an efficient annotation of protein variants.

Results: E-SNPs&GO is a novel method that, given an input protein sequence and a single amino acid variation, can predict whether the variation is related to diseases or not. The proposed method adopts an input encoding completely based on protein language models and embedding techniques, specifically devised to encode protein sequences and GO functional annotations. We trained our model on a newly generated dataset of 101 146 human protein single amino acid variants in 13 661 proteins, derived from public resources. When tested on a blind set comprising 10 266 variants, our method well compares to recent approaches released in literature for the same task, reaching a Matthews Correlation Coefficient score of 0.72. We propose E-SNPs&GO as a suitable, efficient and accurate large-scale annotator of protein variant datasets.

Availability and implementation: The method is available as a webserver at https://esnpsandgo.biocomp.unibo.it. Datasets and predictions are available at https://esnpsandgo.biocomp.unibo.it/datasets.

Contact: pierluigi.martelli@unibo.it

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Single-nucleotide polymorphisms (SNPs) are major sources of human evolution. In many cases, these variations can be directly associated with the onset of genetic diseases. Specifically, SNPs occurring in protein-coding regions often lead to observable changes in the protein residue sequence. Single amino acid variations (SAVs) may have an impact at different levels, hampering protein structure, function, stability, localization and interaction with other proteins and/or nucleotides, hence setting the basis for the onset of pathologic conditions (Lappalainen and MacArthur, 2021; Vihinen, 2021 and references therein).

Public databases, such as HUMSAVAR (The UniProt Consortium, 2021) and ClinVar (Landrum *et al.*, 2018), store a compendium of known SAVs and provide, when available, information about the variant clinical significance. However, clear associations to diseases are still unknown for many SAVs, which substantially remain of Uncertain Significance (US). Therefore, SAV annotation is an issue, and effective computational tools are needed to provide large-scale annotation of uncharacterized human variation data.

In the past years, several computational approaches have been implemented, with the aim of annotating whether a protein variation is or not disease associated (Adzhubei *et al.*, 2010; Calabrese *et al.*, 2009; Carter *et al.*, 2013; Choi *et al.*, 2012; Jagadeesh *et al.*, 2016; Li *et al.*, 2009; Ng and Henikoff, 2001; Niroula *et al.*, 2015; Pejaver *et al.*, 2020; Raimondi *et al.*, 2017; Schwarz *et al.*, 2010; Yang *et al.*, 2022). Methods like SIFT (Ng and Henikoff, 2001) or PROVEAN (Choi *et al.*, 2012) are based on the conservation analysis in multiple sequence alignments. More complex approaches stand on different types of machine-learning frameworks. These

include neural networks (Pejaver *et al.*, 2020), random forests (Carter *et al.*, 2013; Li *et al.*, 2009; Niroula *et al.*, 2015; Raimondi *et al.*, 2017), gradient tree boosting (Jagadeesh *et al.*, 2016; Yang *et al.*, 2022), support vector machines (SVMs) (Calabrese *et al.*, 2009) and naive Bayes classifiers (Adzhubei *et al.*, 2010; Schwarz *et al.*, 2010). Each method is trained/tested on different datasets of SAVs, either extracted directly from public resources like HUMSAVAR (The UniProt Consortium, 2021) and/or ClinVar (Landrum *et al.*, 2018), or taking advantage of pre-compiled datasets of variations, like VariBench (Nair and Vihinen, 2013). Different types of descriptors extract salient features of the protein sequence and/or the local sequence context surrounding the variant position, including physicochemical properties, sequence profiles, conservation scores, predicted structural motifs and functional annotations. SNPs&GO (Calabrese *et al.*, 2009) firstly recognized the importance of functional annotations for the prediction of variant pathogenicity and introduced the LGO feature, a score of association between Gene Ontology (GO) (Ashburner *et al.*, 2000) annotations and the variant pathogenicity. The incorporation of the LGO feature significantly improved the prediction performance of SNPs&GO (Calabrese *et al.*, 2009).

Recent developments in the field of deep learning focus on the definition of new ways of representing protein sequences. Large-scale protein language models (PLMs) are inspired and derived from the natural language processing (NLP) field (Ofer *et al*, 2021). They learn numerical vector representations of protein sequences, containing important features that are reflected in the evolutionary conservation and in the sequence syntax (Bepler and Berger, 2021). These numerical vectors are then adopted to encode protein sequence and/or individual residues in place of canonical, hand-crafted features, such as physicochemical properties or evolutionary information. These distributed protein representations emerge from the application of learning models trained on large databases of sequence data (Bepler and Berger, 2021; Ofer *et al.*, 2021).

Successful PLMs are routinely trained on databases composed of hundreds of millions of unique sequences with hundreds of billions of residues. Training is computationally demanding, routinely requiring weeks or months of computations on high-performance Tensor Processing Units (TPUs) and/or Graphical Processing Units (GPUs) (Elnaggar *et al.*, 2021; Rives *et al.*, 2021). However, the advantage is that most of the computational cost is concentrated on the training phase, and once models are trained they can be adopted to embed new sequences with limited resources in terms of time, memory and computational power.

Embeddings obtained with language models have been recently employed for many different applications with great success, including the prediction of protein function and localization (Littmann *et al.*, 2021; Stärk *et al.*, 2021; Teufel *et al.* 2022), of protein contact maps (Singh *et al.*, 2022) and binding sites (Mahbub and Bayzid, 2022).

Several pre-trained language models currently exist in the literature (Alley *et al.*, 2019; Asgari and Mofrad, 2015; Elnaggar *et al.*, 2021; Heinzinger *et al.*, 2019; Rives *et al.*, 2021; Strodthoff *et al.*, 2020), mainly differing in their specific architectures [autoregressive, bidirectional, masked; see for review Bepler and Berger (2021)] and in the datasets adopted for training.

Not limited to the encoding of protein sequence data, embedding techniques are also applied to model the relationships existing within more complex structures, such as graphs, networks, or biological ontologies (Edera *et al.*, 2022; Grover and Leskovec, 2016; Kandathil *et al.*, 2022; Perozzi *et al.*, 2014; Zhong *et al.*, 2019).

In this article, we attempt to fully exploit the power of language models and embeddings for the prediction of variant pathogenicity from the human protein sequence. On the methodological side, two major contributions can be highlighted. Firstly, we adopt two different and complementary embedding procedures, ProtT5 (Elnaggar *et al.*, 2021) and ESM-1v (Meier *et al.*, 2021), to directly encode an input variation without introducing any hand-crafted feature as previously done. Secondly, leveraging the idea introduced in SNPs&GO (Calabrese *et al.*, 2009), we explore a new way of encoding functional annotations by adopting a model called Anc2Vec (Edera *et al.*, 2022), specifically designed for the embedding of GO terms (Ashburner *et al.*, 2000).

We trained an SVM using the above input encoding on a newly generated dataset of 101 146 human disease-related and benign variations obtained from the rational merging of data deposited in two databases, HUMSAVAR (The UniProt Consortium, 2021) and ClinVar (Landrum *et al.*, 2018). The method is tested on an independent, non-redundant blind set comprising 10 266 variations, adopting stringent homology reduction and evaluation procedures. Results obtained in a comparative benchmark and including one of the most recent and effective methods (Pejaver *et al.*, 2020), demonstrate that our model performs at the level or even better than the state-of-the-art (when available for comparison) reaching a Matthews Correlation Coefficient (MCC) of 0.72. Based on an input encoding derived solely from embedding models, our method is fast: this makes it suitable for large-scale annotation of human pathogenic variants.

We release our tool as a webserver at https://esnpsandgo.biocomp.unibo.it.

2 Materials and methods

2.1 Dataset

We obtained the dataset of SAVs by merging information extracted from two resources: HUMSAVAR (accessed on August 4, 2021), listing all missense variants annotated in human UniProt/SwissProt (The UniProt Consortium, 2021) entries, and ClinVar (accessed on March 29, 2021), the NCBI resource of relationships among human variations and disease phenotypes (Landrum *et al.*, 2018).

Both databases classify the effect of SAVs into different classes: Pathogenic or Likely Pathogenic (P/LP), Benign or Likely Benign (B/LB) and of US. We retained only P/LP SAVs clearly associated with the diseases catalogued in OMIM (Amberger *et al.*, 2019) or in MONDO (Shefchek *et al.*, 2020). We collected also all the B/LB variations and excluded SAVs labelled as US, somatic, or with contrasting annotations of the effect.

Overall, the dataset consists of 13 661 protein sequences endowed with 111 412 SAVs, including 43 895 P/LP SAVs in 3603 proteins and 67 517 B/LB SAVs in 13 229 proteins (Table 1, last row).

For all proteins in the dataset, we extracted GO (Ashburner *et al.*, 2000) annotations from the corresponding entry in UniProt. Overall, our dataset is annotated with 17 076 GO terms, including 11 476 Biological Process (BP), 3955 Molecular Function (MF) and 1645 Cellular Component (CC). The complete dataset is available at https://esnpsandgo.biocomp.unibo.it/datasets.

2.1.1 Cross-validation procedure and generation of the blind test set

To avoid biases between training and testing sets, we adopted a stringent clustering procedure to generate cross-validation sets. Firstly, we clustered protein sequences with the MMseqs2 program (Steinegger and Söding, 2017), by constraining a minimum sequence identity of 25% over a pairwise alignment coverage of at least 40%. We used a connected component clustering strategy so that if two proteins are clustered with a third one, they both end up in the same set. In this way, we limit sequence redundancy between training and testing sets, enabling a fair evaluation of the results. We selected 10% of the data to construct the blind test set for assessing the generalization performance of our approach and for benchmarking it

Table 1. The dataset of SAVs adopted in this study

Dataset	No. of pathogenic SAVs	No. of neutral SAVs	No. of proteins
Training set	39 812	61 334	12 347
Blind test set	4083	6183	1314
Total	43 895	67 517	13 661

with other popular methods available. The remaining 90% of the dataset was further split into 10 equally distributed subsets that were used in a 10-fold cross-validation procedure for optimizing the input encoding and for fixing the model hyperparameters. We also tried a 20–80% split (20% of the data for the blind test set and 80% for training with the 10-fold cross-validation procedure) and obtained a very similar performance. For this reason, we list results corresponding to the 10% blind test. When performing cross-validation, we took care of preserving the balancing of positive and negative examples in each subset (Supplementary Table S1).

It is worth noticing that the blind test can share similarity with proteins included in the training sets of the other benchmarked methods.

2.2 General overview of the approach

Figure 1 depicts the architecture of E-SNPs&GO, including three major blocks: an Input encoding, a Predictor and an Output. The input consists of a human protein sequence and a SAV occurring at a specific position along the sequence. In the input encoding phase, the sequence and its variant are embedded with two different procedures, ESM-1v (Meier *et al.*, 2021) and ProtT5 (Elnaggar *et al.*, 2021), generating for each sequence 1280 and 1024 features, respectively. In order to embed the functional protein annotation of the wild-type protein, we adopt Anc2Vec (Edera *et al.*, 2022), computing three sets of 200 features corresponding to the different subontologies.

In the predictor, the vector representation generated in the input encoding is then processed using a principal component analysis (PCA), which reduces the dimensionality of the input from 5208 features to 2400. The output feeds a SVM classifier performing the final labelling as Pathogenic (P/LP) or Benign (B/LB). A given input variant is predicted as pathogenic when the SVM output score ≥ 0, benign otherwise. A final calibration step allows to convert scores into probabilities for a variant to be pathogenic. Details of the methods included in E-SNPs&GO, are listed in the following sections.

2.3 Input encoding: embeddings of protein sequence, its variant and GO terms

2.3.1 Transformers for embedding of protein sequences and their variants

Several prominent language models and corresponding embedding generation schemes in NLP are available, and some of these have been adapted to protein sequences to perform specific prediction tasks (Bepler and Berger, 2021). Large-scale PLMs aim at learning a numerical vector representation that allows reconstructing the input sequence.

Among PLMs, transformer-based models (Vaswani *et al.*, 2017) aim to solve the problem of efficiently capturing long-distance interactions in the sequence. Transformers are architectures that include a self-attention mechanism to extract the context information from the whole sequence (Vaswani *et al.*, 2017). In general, a transformer language model builds on top of an encoder–decoder architecture. However, the different transformer-based PLMs only utilize either the encoder or the decoder part. In this respect, transformer-based PLMs can be classified in three different categories: (i) encoder-only models use only the encoder part of the transformer accessing the whole input sequence and are trained to reconstruct a somewhat corrupted version of the input (e.g. masking random positions along the sequence); (ii) decoder-only models (also called autoregressive models) use only the decoder part accessing, at each position, all the residues placed before the current one in the sequence and are usually trained to predict the next residue in the sequence; (iii) sequence-to-sequence models use both the encoder and the decoder and are trained to reconstruct a masked input sequence (Vaswani *et al.*, 2017).

The learned representation captures important features of the proteins, including physicochemical, structural, functional and evolutionary features (Bepler and Berger, 2021; Ofer *et al.*, 2021). By transfer learning, the embedded schemes are provided as input to Predictor block (Fig. 1).

In this article, we adopt two different protein embedding schemes, based on two different transformers models: ESM-1v (Meier *et al.*, 2021), an encoder-only model, and ProtT5 (Elnaggar

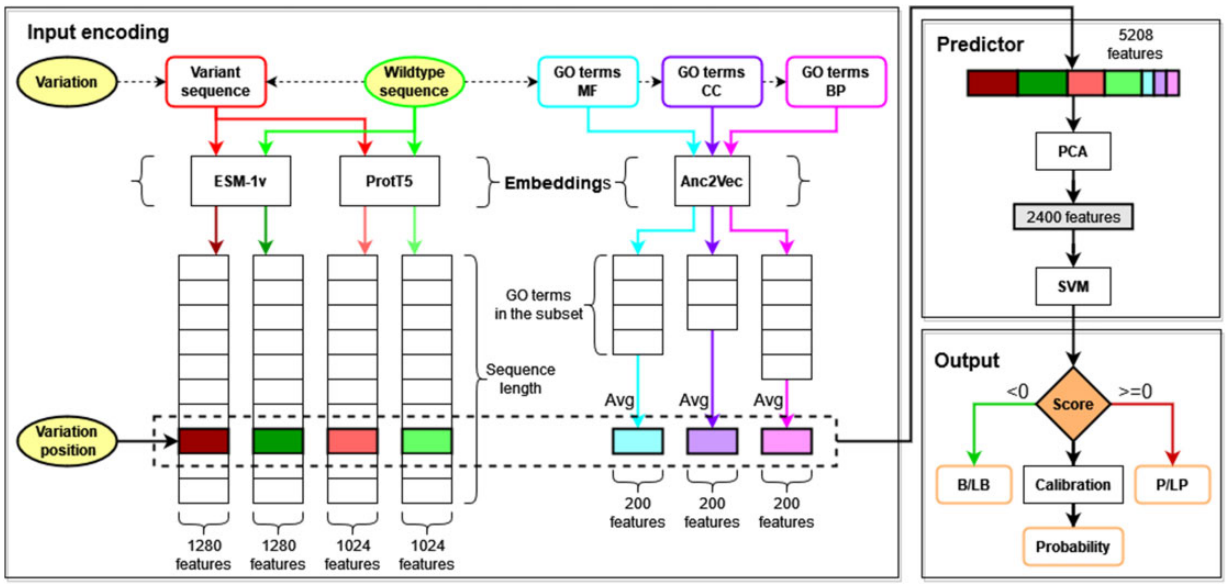


Fig. 1. General overview of the architecture of E-SNPs&GO. Inputs (wild-type sequence, variation and variation position) are in yellow. The architecture includes three major blocks: an Input encoding, a Predictor and an Output. During the Input encoding, three embedding models are adopted to generate vector representations. The wild-type sequence (green) and the variant sequence (red) are modelled with ESM-1v (Meier *et al.*, 2021) and ProtT5 (Elnaggar *et al.*, 2021). The GO functional annotations (blue MF, purple CC and pink BP) are modelled with Anc2Vec (Edera *et al.*, 2022). The vectors within the dashed box (marked with different colors), representing the variation position and the averaged (Avg) GO terms of the wild-type sequence, are then concatenated together to obtain a final representation consisting of $1280\times2+1024\times2+200\times3=5208$ features. This vector is fed to the Predictor, which includes a PCA to reduce the input dimensionality (from 5208 to 2400) and a SVM providing as a final output a binary classification into B/LB (negative class, Score <0) or P/LP (positive class, Score ≥ 0). We apply an Isotonic Regression (Calibration) to obtain a calibrated probability (A color version of this figure appears in the online version of this article.)

et al., 2021), a sequence-to-sequence model. The major difference stands in the volume of the sequence datasets used for generating the embedding schemes and in the adoption of different training procedures. ESM-1v was trained on a single run using a dataset of 98 million unique sequences extracted from UniRef90 (Suzek *et al.*, 2015). ESM-1v releases five models generated by training with five different random seeds (Meier *et al.*, 2021). Apparently, only a small difference in performance is obtained when the ensemble is compared to a single model (Meier *et al.*, 2021). Therefore, to reduce the computational cost, we adopted only one model (the first one). ProtT5 (version XL U50) was trained using a two-step procedure: in a first pass, training was performed using the large BFD database (Steinegger *et al.*, 2019; Steinegger and Söding, 2018), comprising the whole UniProt as well as protein sequences translated from multiple metagenomic sequencing projects, and consisting of about 2.1 billion unique sequences. In the second pass, a fine-tuning of the model was obtained using a smaller database derived from UniRef50 (Suzek *et al.*, 2015) and including 45 million unique sequences.

2.3.2 Embedding of biological ontologies

The concept of embedding can be generalized to any kind of data with different underlying structures, such as graphs or networks (Grover and Leskovec, 2016; Perozzi *et al.*, 2014). In particular, several embedding models have been defined to provide a numerical representation of nodes in ontologies (Chen *et al.*, 2021; Zhong *et al.*, 2019). Here, we adopt Anc2Vec (Edera *et al.*, 2022), a method that learns a vector representation for GO terms, by preserving ancestor relationships.

Because the embedding is not context-dependent, we precompute the vector representation for each possible GO term.

2.4 Predictor

2.4.1 Predictor input

For encoding variations, we firstly perform a full-sequence generation of embeddings using both the ESM-1v (Meier *et al.*, 2021) and the ProtT5 XL U50 (Elnaggar *et al.*, 2021) models. Given a protein sequence with L residues, this provides protein encodings of dimensions $L\times1280$ and $L\times1024$, respectively. Sequence embeddings are carried out independently on both the wild-type and the variant sequence.

For a variation at position i in a protein sequence, we compute a vector of 4608 features, including:

- 1280 features corresponding to ESM-1v embedding in position i of the variated sequence.
- 1280 features corresponding to ESM-1v embedding in position i of the wild-type sequence.
- 1024 features corresponding to ProtT5 (version XL U50) embedding in position i of the variated sequence.
- 1024 features corresponding to ProtT5 (version XL U50) embedding in position i of the wild-type sequence.

The ESM-1v embedding model constrains the maximal protein length (L) to 1024 residues. For this reason, variations occurring on longer sequences were encoded using a 201 long sequence window centred on the variant position.

After this step, we extract all the GO terms annotated in the UniProt entry of the wild-type protein carrying the variation. Potential term redundancy is removed by retaining only leaf terms. Terms from the three different GO sub-ontologies (MF, CC and BP) are processed independently. Each annotated GO term is then embedded as a vector of 200 features using the Anc2Vec model (Edera *et al.*, 2022). To obtain a single vector representation independent of the number of terms of a given protein, we average all the vector encodings (Fig. 1). Three final average vectors, one for each GO sub-ontology, are concatenated obtaining a protein function encoding of 600 components.

The final variation encoding comprises 5208 features, obtained by merging the local positional embedding (4608 features from ESM-1V + ProtT5 XL U50) described above and the Anc2Vec functional encoding (600 features). Eventually, we encode the different embeddings separately (see Section 3 and Table 2).

2.4.2 Model selection and implementation

The predictor includes two cascading components (Fig. 1): a PCA for reducing the dimensionality of the input features and a binary SVM with a Radial Basis Function (RBF) kernel, which performs the variant classification into pathogenic or not. We optimized the hyperparameters of both methods (such as the number of components of PCA, the SVM cost parameter C and the gamma coefficient of the RBF kernel) with a grid search procedure. A complete list of hyperparameters tested and their optimal values are available in Supplementary Table S2.

It is worth clarifying that, during both cross-validation and blind testing, the execution of the PCA step is always computed on the training set and then applied for projecting vectors of the testing set in the reduced space.

All methods are implemented in Python3 using the scikit-learn library (Pedregosa *et al.*, 2011). ESM-1v and ProtT5 embeddings are computed with the bio-embeddings package (Dallago *et al.*, 2021).

The complete machine-learning workflow is compliant with the DOME recommendation checklist (Walsh *et al.*, 2021), as reported in Supplementary Table S3.

2.5 Output

The SVM adopted for classification computes a decision function that represents the distance of the point mapping the input from the discrimination boundary. We use this value to estimate the reliability of the prediction, in terms of the probability of the input variation to be pathogenic (Fig. 1).

In a perfectly calibrated method, when a set of predictions scored with probability P is tested on real data, we expect that the fraction of true positives is exactly P. In this work, we adopt a procedure previously described (Benevenuta *et al.*, 2021) to obtain a calibrated probability that we provide in output alongside the predicted class. In particular, we fit an Isotonic Regression (Niculescu-Mizil and Caruana, 2005) in cross-validation and we use it to obtain a probability score on the blind test. Supplementary Figure S1 shows that E-SNPs&GO output probabilities are very close to being perfectly calibrated, more than other popular methods.

Keeping as a reference the probability of being P/PL, the probability score ($P_{\mathrm{P/PL}}$) gives an integer Reliability Index from 0 (random prediction) to 10 (certain prediction) using the formula:

$$\mathrm{RI} = \mathrm{round}\left(20 \times \left|P_{P/LP} - 0.5\right|\right). \tag{1}$$

2.6 Scoring indexes

We assess the performance with the following scores. P/LP variations are assumed to be the positive class, B/LB variations are the negative class. In what follows, TP, TN, FP and FN are true positive, true negative, false positive and false negative predictions, respectively.

We compute the following scoring measures:

- Accuracy (Q_2):

$$Q_2 = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}. \tag{2}$$

- Precision:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}. \tag{3}$$

Table 2. Performance of different embedding schemes

Input encoding	Q_2 (%)	Precision (%)	Recall (%)	*F*1-score (%)	ROC-AUC (%)	MCC
ESM-1v	82.4 (±1.5)	80.4 (±2.6)	77.0 (±2.8)	78.6 (±1.9)	81.6 (±1.5)	0.64 (±0.03)
ESM-1v+GO	83.3 (±1.4)	81.7 (±2.5)	78.1 (±2.7)	79.8 (±1.8)	82.6 (±1.4)	0.66 (±0.03)
ProtT5	83.0 (±1.3)	79.8 (±1.9)	80.0 (±2.8)	79.9 (±1.7)	82.6 (±1.4)	0.65 (±0.03)
ProtT5+GO	83.7 (±1.1)	81.8 (±1.9)	79.2 (±2.5)	80.5 (±1.5)	83.1 (±1.3)	0.67 (±0.02)
ESM-1v+ProtT5	83.6 (±1.4)	81.8 (±2.3)	78.6 (±2.9)	80.1 (±1.8)	82.9 (±1.5)	0.66 (±0.03)
ESM-1v+ProtT5+GO(-PCA)	83.1 (±0.8)	81.0 (±1.4)	78.0 (±1.5)	79.4 (±1.1)	82.8 (±0.8)	0.66 (±0.02)
ESM-1v+ProtT5+GO(+PCA)	85.1 (±0.9)	82.4 (±1.7)	79.1 (±1.7)	80.7 (±1.1)	84.1 (±0.9)	0.69 (±0.02)

Note: We adopted a 10-fold cross-validation on a training set comprising 101 146 human variations (Table 1) for testing the effect of different input encodings on the performances of the method. Standard deviation (between brackets) is computed over the 10 cross-validation sets and scoring indexes (defined in Section 2.6) are average values.

ESM-1v ($2\times1280=2560$ features).
ESM-1v + GO ($2\times1280+3\times200=3160$ features).
ProtT5 ($2\times1024=2048$ features).
ProtT5 + GO ($2\times1024+3\times200=2648$ features).
ESM-1v + ProtT5 ($2\times1280+2\times1024=4608$ features).
ESM-1v + ProtT5 + GO (−PCA) ($2\times1280+2\times1024+3\times200=5208$ features), no PCA used.
ESM-1v + ProtT5 + GO (+PCA) ($2\times1280+2\times1024+3\times200=5208$ features), PCA used to reduce dimensionality.

- Recall:

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}. \tag{4}$$

- *F*1-score, the harmonic mean of precision and recall:

$$F1 = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}. \tag{5}$$

- Area under the receiver operating characteristic curve (ROC-AUC).
- MCC:

$$\text{MCC} = \frac{\text{TP}\times\text{TN}-\text{FP}\times\text{FN}}{\sqrt{(\text{TP}+\text{FP})\times(\text{TP}+\text{FN})\times(\text{TN}+\text{FP})\times(\text{TN}+\text{FN})}}. \tag{6}$$

3 Results

3.1 Assessing the contribution of different input encodings

To select the optimal input encoding, we performed different experiments to test various combinations of input features. To this aim, we trained in cross-validation several independent SVM+PCA models using different input features and using the MCC to score and select the optimal model.

GO terms provide global protein information. Their embedding does not consider the specific variant position. If the prediction is run considering only averaged embedded GO terms vector (Fig. 1), the predictor performance is very low (MCC = 0.27, data not shown). Different input encodings, corresponding to different predictors, perform differently (Table 2). The inclusion of GO embeddings in the final input is always beneficial, improving MCC by 2 or 3 percentage points in all cases (compare ESM-1v, ProtT5 and ESM-1v+ProtT5 with or without GO, respectively in Table 2). Considering the two protein sequence embeddings, ProtT5 outperforms ESM-1v both with and without the additional GO information. Most notably, the model trained on data from ProtT5 alone is the most balanced, reaching equal precision and recall. Finally, the concatenation of both sequence encodings and the GO embedding provides the best performance (MCC = 0.69), leading to an increase in precision without a corresponding decrease in recall.

Based on these results, we select the model trained with ESM-1v+ProtT5+GO as the optimal one.

3.2 Benchmark on the blind test set

We test our method adopting both a 10-fold cross-validation procedure and an independent blind test set constructed to be non-redundant with respect to the training dataset (see Section 2.1). Table 3 lists the results. E-SNPs&GO obtains similar results in cross-validation and blind test, making it very robust to generalization. Concerning individual indexes, our method seems to be slightly more precise than sensitive (compare Precision and Recall).

Table 3 includes also a comparative benchmark of our method with other state-of-the-art tools, including our SNPs&GO (Calabrese *et al.*, 2009), SIFT (Ng and Henikoff, 2001), PolyPhen-2 (Adzhubei *et al.*, 2010), PROVEAN (Choi *et al.*, 2012) and MutPred2 (Pejaver *et al.*, 2020), one of the most recent and best-performing approaches in the field. Methods are scored adopting our blind test set (Section 2.1), ensuring a fair evaluation of the performance of our method. However, this does not completely exclude the presence of biases in the evaluation of the other tools (with the exception of our SNPs&GO), since variations included in our blind test may be present in the respective training sets, leading to potential overestimation of their performance.

In Table 3, it appears that in this benchmark our method is performing at the state-of-the-art. Among tested approaches, PROVEAN, SIFT and PolyPhen-2, reporting MCCs of 0.57, 0.53 and 0.50, respectively, are scoring lower than our previous SNPs&GO (that achieves an MCC of 0.58). Our E-SNPs&GO and MutPred2, score with significantly higher MCC values of 0.72 and 0.71, respectively. Noticeably the embedding procedure seems to grasp all the properties extracted by an ensemble of different predictors of functional, structural and physicochemical properties, such as the one used by MutPred2 (including over 50 tools). Looking at individual scoring measures, MutPred2 appears more sensitive while our method reports a higher precision.

A detailed ablation study performed to evaluate the effect of the GO terms on the prediction scores (Supplementary Table S4), indicates that the CC sub-ontology slightly outperforms the others.

3.3 Prediction of variants of uncertain significance

We tested E-SNPs&GO on a dataset of 2588 proteins annotated with 9165 variants of uncertain significance (VUS) extracted from HUMSAVAR (accessed on May 12, 2022). Given that they are uncertain, we cannot assess our performances on this dataset. However, we can sample our predicted annotation in terms of probability and reliability [Equation (6)]. Setting as a reference the probability of being P/LP, Figure 2 shows the distribution of E-SNPs&GO predictions over the whole VUS set as a function of probability and reliability index. A total of 4537 variations are P/LP (pathogenicity probability ≥0.5), while 4628 are B/LB

Table 3. Benchmark of our and other top scoring methods available in literature

Input encoding		Q_2 (%)	Precision (%)	Recall (%)	*F*1-score (%)	ROC-AUC (%)	MCC
E-SNPs&GO[a]	Cross-validation	85.1 (±0.9)	82.4 (±1.7)	79.1 (±1.7)	80.7 (±1.1)	84.1 (±0.9)	0.69 (±0.018)
E-SNPs&GO[a]	Blind test set	86.8	85.7	80.1	82.8	85.6	0.72
SNPs&GO[a]	Blind test set	79.8	84.8	63.2	72.4	77.5	0.58
MutPred2.0[b]	Blind test set	85.6	78.6	87.7	82.9	85.9	0.71
PROVEAN[c]	Blind test set	78.2	68.7	83.0	75.2	79.0	0.57
SIFT[d]	Blind test set	74.4	62.7	88.0	73.2	76.7	0.53
PolyPhen-2[e]	Blind test set	72.3	60.6	89.5	72.2	75.1	0.50

Note: The benchmark is performed on a test set comprising 10 266 human variations (Table 1, 10% of the total number of SAVs) that is blind with respect to our training set. It could be redundant with respect to the training sets of other methods, leading to a possible overestimation of their performances. We also report our performances in cross-validation for comparison. We increased the size of the blind test set up to 20% of the number of SAVs and the E-SNPs&GO MCC score values were negligibly affected (0.5%, data not shown).

[a]E-SNPs&GO: this article; SNPs&GO (Calabrese *et al.*, 2009).

[b]MutPred2.0 (Pejaver *et al.*, 2020).

[c]PROVEAN (Choi *et al.*, 2012).

[d]SIFT (Ng and Henikoff, 2001).

[e]PolyPhen-2 (Adzhubei *et al.*, 2010).

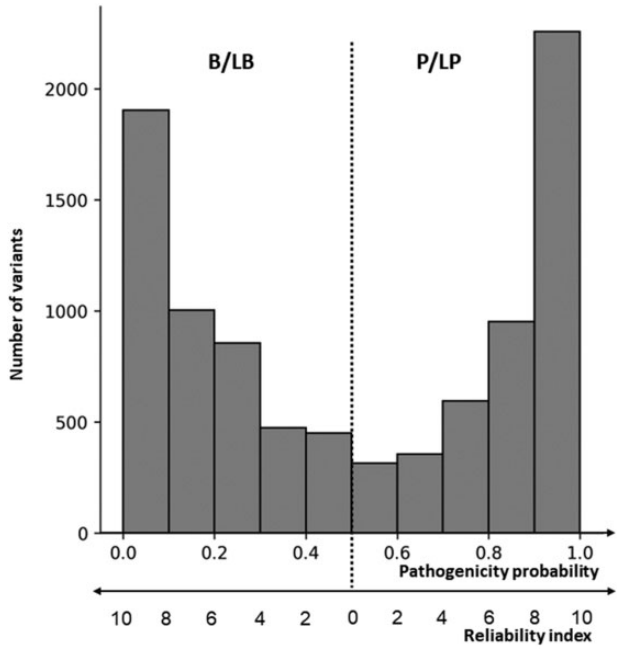


Fig. 2. Distribution of predicted pathogenicity probabilities for the dataset of VUS. The value 0.5 discriminates between B/LB and P/LP prediction. Probability values close to either 0 or 1 correspond to prediction with a high reliability index [Equation (1)]

(pathogenicity probability <0.5). The reliability index increases as the probability goes towards 1 or 0 for P/LP and B/LB predictions, respectively [Equation (6)]. In the dataset, 3210 P/LP and 2908 B/LB predictions score with a reliability [RI, Equation (6)] ≥6, accounting for the 67% of VUS. The remaining 33% is predicted with RI lower than 6. For further validation, VUS predictions are available at https://esnpsandgo.biocomp.unibo.it/datasets.

3.4 E-SNP&GO web server

E-SNPs&GO web server is available at https://esnpsandgo.biocomp.unibo.it. The server allows users to submit up to 1000 variations per single job. Upon job completion, the results can be visualized on the web page and downloaded in either a tab-separated or a JSON file.

We measured the average E-SNPs&GO runtime by submitting 100 different jobs each including 1000 variations randomly selected from the blind test set. In order to estimate the real execution time for the end user, this experiment was performed in the machine hosting the web server, equipped with one AMD EPYC 7301 CPU with 12 cores, 48 GB of RAM and no GPU available. On average, we obtain a running time of 12.4 ± 4.4 s per variation, when submitting the maximum allowed number of variations per job (1000 variations). This highlights a significant improvement over time-consuming approaches using canonical features such as evolutionary information extracted from multiple sequence alignments.

4 Conclusions

We introduce E-SNPs&GO, a method based on language models for annotating whether a single-nucleotide variation is or is not P/LP. We adopt two different protein embedding procedures based on transformers, ESM-1v (Meier *et al.*, 2021) and ProtT5 (Elnaggar *et al.*, 2021). Both embedding methods have been developed and tested on protein variant related problems, such as deep mutational scanning (Marquet *et al.*, 2021; Meier *et al.*, 2021). Here, we address the problem of annotating pathogenic versus benign variations. To this aim, we also add an embedding scheme for functional annotations of wild-type proteins, Anc2Vec (Edera *et al.*, 2022), a method that learns a vector representation for GO terms by preserving ancestor relationships. When benchmarked towards state-of-the-art methods available, E-SNPs&GO well compares to the recently developed MutPred2.0 (Pejaver *et al.*, 2020), which includes as input sequence features derived from some 50 predictors and outperforms previously published methods. Evidently, protein language models learn all the relevant information that can be eventually introduced as input by predictors addressing different tasks.

We prove that embedding models overpass the problem of having as input thousands of different features in order to collect all the relevant features for a reliable annotation of the human pathogenic variations.

Funding

This work was supported by PRIN 2017 project [2017483NH8 to C.S.] (Italian Ministry of University and Research).

Conflict of Interest: none declared.

Data availability

The data underlying this article are available in the article, in its online supplementary material and at the E-SNPs&GO web site: https://esnpsandgo.biocomp.unibo.it.

References

Adzhubei,I.A. *et al.* (2010) A method and server for predicting damaging missense mutations. *Nat. Methods*, **7**, 248–249.

Alley,E.C. *et al.* (2019) Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods*, **16**, 1315–1322.

Amberger,J.S. *et al.* (2019) OMIM.org: leveraging knowledge across phenotype–gene relationships. *Nucleic Acids Res.*, **47**, D1038–D1043.

Asgari,E. and Mofrad,M.R.K. (2015) Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLoS One*, **10**, e0141287.

Ashburner,M. *et al.* (2000) Gene ontology: tool for the unification of biology. *Nat. Genet.*, **25**, 25–29.

Benevenuta,S. *et al.* (2021) Calibrating variant-scoring methods for clinical decision making. *Bioinformatics*, **36**, 5709–5711.

Bepler,T. and Berger,B. (2021) Learning the protein language: evolution, structure, and function. *Cell Syst.*, **12**, 654–669.e3.

Calabrese,R. *et al.* (2009) Functional annotations improve the predictive score of human disease-related mutations in proteins. *Hum. Mutat.*, **30**, 1237–1244.

Carter,H. *et al.* (2013) Identifying Mendelian disease genes with the variant effect scoring tool. *BMC Genomics*, **14** (**Suppl. 3**), S3.

Chen,J. *et al.* (2021) OWL2Vec: embedding of OWL ontologies. *Mach. Learn.*, **110**, 1813–1845.

Choi,Y. *et al.* (2012) Predicting the functional effect of amino acid substitutions and indels. *PLoS One*, **7**, e46688.

Dallago,C. *et al.* (2021) Learned embeddings from deep learning to visualize and predict protein sets. *Curr. Protoc.*, **1**, e113.

Edera,A.A. *et al.* (2022) Anc2vec: embedding gene ontology terms by preserving ancestors relationships. *Brief. Bioinformatics*, **23**, bbac003.

Elnaggar,A. *et al.* (2021) ProtTrans: towards cracking the language of life's code through Self-Supervised deep learning and high performance computing. *IEEE Trans. Pattern Anal. Mach. Intell.*, **14**, 1.

Grover,A. and Leskovec,J. (2016) node2vec: scalable feature learning for networks. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016, San Francisco, CA, USA*, pp. 855–864.

Heinzinger,M. *et al.* (2019) Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics*, **20**, 723.

Jagadeesh,K.A. *et al.* (2016) M-CAP eliminates a majority of variants of uncertain significance in clinical exomes at high sensitivity. *Nat. Genet.*, **48**, 1581–1586.

Kandathil,S.M. *et al.* (2022) Ultrafast end-to-end protein structure prediction enables high-throughput exploration of uncharacterized proteins. *Proc. Natl. Acad. Sci. USA*, **119**, e2113348119.

Landrum,M.J. *et al.* (2018) ClinVar: improving access to variant interpretations and supporting evidence. *Nucleic Acids Res.*, **46**, D1062–D1067.

Lappalainen,T. and MacArthur,D.G. (2021) From variant to function in human disease genetics. *Science*, **373**, 1464–1468.

Li,B. *et al.* (2009) Automated inference of molecular mechanisms of disease from amino acid substitutions. *Bioinformatics*, **25**, 2744–2750.

Littmann,M. *et al.* (2021) Embeddings from deep learning transfer GO annotations beyond homology. *Sci. Rep.*, **11**, 1160.

Mahbub,S. and Bayzid,M.S. (2022) EGRET: edge aggregated graph attention networks and transfer learning improve protein–protein interaction site prediction. *Brief. Bioinformatics*, **23**, bbab578.

Marquet,C. *et al.* (2021) Embeddings from protein language models predict conservation and variant effects. *Hum. Genet.*, **141**, 1629–1647.

Meier,J. *et al.* (2021) Language models enable zero-shot prediction of the effects of mutations on protein function. In: Ranzato,M. *et al.* (eds) *Advances in Neural Information Processing Systems. Proceedings of NeurIPS 2021*, Vol. **34**. pp. 29287–29303.

Nair,P.S. and Vihinen,M. (2013) VariBench: a benchmark database for variations. *Hum. Mutat.*, **34**, 42–49.

Ng,P.C. and Henikoff,S. (2001) Predicting deleterious amino acid substitutions. *Genome Res.*, **11**, 863–874.

Niculescu-Mizil,A. and Caruana,R. (2005) Predicting good probabilities with supervised learning. In: *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05. Association for Computing Machinery, New York, NY, USA, pp. 625–632.

Niroula,A. *et al.* (2015) PON-P2: prediction method for fast and reliable identification of harmful variants. *PLoS One*, **10**, e0117380.

Ofer,D. *et al.* (2021) The language of proteins: NLP, machine learning & protein sequences. *Comput. Struct. Biotechnol. J.*, **19**, 1750–1758.

Pedregosa,F. *et al.* (2011) Scikit-learn: machine learning in python. *J. Mach. Learn. Res.*, **12**, 2825–2830.

Pejaver,V. *et al.* (2020) Inferring the molecular and phenotypic impact of amino acid variants with MutPred2. *Nat. Commun.*, **11**, 5918.

Perozzi,B. *et al.* (2014) DeepWalk: online learning of social representations. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA*, pp. 701–710.

Raimondi,D. *et al.* (2017) DEOGEN2: prediction and interactive visualization of single amino acid variant deleteriousness in human proteins. *Nucleic Acids Res.*, **45**, W201–W206.

Rives,A. *et al.* (2021) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. USA*, **118**, e2016239118.

Schwarz,J.M. *et al.* (2010) MutationTaster evaluates disease-causing potential of sequence alterations. *Nat. Methods*, **7**, 575–576.

Shefchek,K.A. *et al.* (2020) The monarch initiative in 2019: an integrative data and analytic platform connecting phenotypes to genotypes across species. *Nucleic Acids Res.*, **48**, D704–D715.

Singh,J. *et al.* (2022) SPOT-Contact-LM: improving single-sequence-based prediction of protein contact map using a transformer language model. *Bioinformatics*, **38**, 1888–1894.

Stärk,H. *et al.* (2021) Light attention predicts protein location from the language of life. *Bioinform. Adv.*, **1**, vbab035.

Steinegger,M. and Söding,J. (2017) MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.*, **35**, 1026–1028.

Steinegger,M. and Söding,J. (2018) Clustering huge protein sequence sets in linear time. *Nat. Commun.*, **9**, 2542.

Steinegger,M. *et al.* (2019) Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nat. Methods*, **16**, 603–606.

Strodthoff,N. *et al.* (2020) UDSMProt: universal deep sequence models for protein classification. *Bioinformatics*, **36**, 2401–2409.

Suzek,B.E. *et al.*; the UniProt Consortium. (2015) UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, **31**, 926–932.

Teufel,E. *et al.* (2022) SignalP 6.0 predicts all five types of signal peptides using protein language models. *Nat. Biotechnol.*, **40**, 1023–1025.

The UniProt Consortium (2021) UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Res.*, **49**, D480–D489.

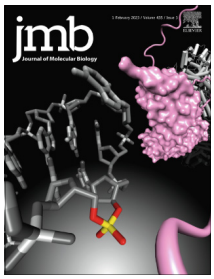
Vaswani,A. *et al.* (2017) Attention is all you need. In: *Proceedings of the 31st Annual Conference on Neural Information Processing Systems, NIPS 2017, Long Beach, CA, USA*, pp. 5999–6009

Vihinen,M. (2021) Functional effects of protein variants. *Biochimie*, **180**, 104–120.

Walsh,I. *et al.*; ELIXIR Machine Learning Focus Group. (2021) DOME: recommendations for supervised machine learning validation in biology. *Nat. Methods*, **18**, 1122–1127.

Yang,Y. *et al.* (2022) PON-All, amino acid substitution tolerance predictor for all organisms. *Front. Mol. Biosci.*, **9**, 867572.

Zhong,X. *et al.* (2019) GO2Vec: transforming GO terms and proteins to vector representations via graph embeddings. *BMC Genomics*, **20**, 918.



ISPRED-SEQ: Deep Neural Networks and Embeddings for Predicting Interaction Sites in Protein Sequences

Matteo Manfredi † **Castrense Savojardo** † **Pier Luigi Martelli** * **and Rita Casadio**

Biocomputing Group, *Dept. of Pharmacy and Biotechnology, University of Bologna, Italy*

Correspondence to Pier Luigi Martelli: *pierluigi.martelli@unibo.it (P.L. Martelli)*
https://doi.org/10.1016/j.jmb.2023.167963
Edited by Michael Sternberg

Abstract

The knowledge of protein–protein interaction sites (PPIs) is crucial for protein functional annotation. Here we address the problem focusing on the prediction of putative PPIs considering as input protein sequences. The issue is important given the huge volume of protein sequences compared to experimental and/or computed structures. Taking advantage of protein language models, recently developed, and Deep Neural networks, here we describe ISPRED-SEQ, which overpasses state-of-the-art predictors addressing the same problem. ISPRED-SEQ is freely available for testing at https://ispredws.biocomp.unibo.it.

Introduction

Proteins are key players in most biological processes. Proteins are social entities and interact with membranes, within themselves or with other proteins, and/or biomolecules (including nucleic acids) to accomplish their functions within the cell. Among all the different features that protein functional annotation requires, it is also important to determine the likelihood of protein–protein interaction. Therefore, effective computational tools for the prediction of protein–protein interactions are important to characterize protein function and to expand interactomes of different species.[1–3]

The identification of Protein-Protein Interaction (PPI) sites, namely protein residues involved in physical interactions within interacting proteins, can be addressed using two complementary approaches. On one hand, different biochemical and biophysical experimental methods (such as X-ray crystallography, nuclear magnetic resonance , alanine scanning mutagenesis and chemical cross-linking) can be applied to determine protein–protein interfaces at the atomic or residue level.[4] Although very accurate, the applicability of these methods to large-scale characterization of PPI is still hampered by economical and technical issues.

On the other hand, computational methods are cost-effective solutions to complement experimental approaches in identifying and characterizing PPI sites. Docking programs are the major class of computational tools to study PPIs [for review, see ref 2]. Very accurate models can be obtained through docking when the two interacting partners are known in advance.

However, when the interacting partner/s is/are not known, machine-learning approaches can compute PPI sites on unbound protein chains. Historically, these methods have been relying on several physicochemical features extracted from protein sequence and/or structure and they can discriminate between interacting and non-interacting residues.[2]

The most accurate approaches are based on information extracted from protein 3D structures. Very informative features include protein solvent accessibility, protrusion, depth indexes, secondary structures, B-factors, and general geometrical features.[5]

Prediction of PPI sites from protein sequence alone is still challenging and methods developed

Please cite tts article as: M. Manfredi, C. Savojardo, P.L. Martelli, et al., ISPRED-SEQ: Deep Neural Networks and Embeddings for Predicting Interaction Sites in Protein SequencesJournal of Molecular Biology, https://doi.org/10.1016/j.jmb.2023.167963

for this specific task are less performing than those based on 3D structures. Methods implemented so far for PPI prediction from protein sequence include in input evolutionary information, conservation scores and physical–chemical properties of amino acids (e.g., hydrophobicity, polarity, charge and/or conformational propensities). Additionally, structural features computed from protein sequence with specific classifiers, such as predicted solvent accessibility and secondary structure, are also included with the aim of filling the scoring gap with structure-based approaches. Several methods have been developed in the past and recent years,[2] based mainly on different types of machine learning, including shallow and deep neural networks.[6–15]

Recently, protein language models trained on large volumes of sequence datasets have been proven to be effective in providing protein/residue representations that are alternative and competitive with canonical hand-crafted features such as evolutionary information and physicochemical properties.[17–20] Representations/embeddings provided by these models have been successfully adopted in many prediction tasks.[21–25]

Here we present ISPRED-SEQ, a novel webserver based on a deep-learning model to predict PPI-sites from protein sequence encoded with an embedding procedure. The method stands on a deep architecture combining convolutional blocks and three cascading fully connected layers. ISPRED-SEQ is trained on a dataset of 6,066 protein chains derived from a dataset available in literature[14]. The main novelty of ISPRED-SEQ is the input generation, obtained using two state-of-the-art protein language models, ESM1-b[17] and ProtT5.[18]

We benchmark ISPRED-SEQ on four different independent test data derived from literature.[9,14–15,26–27] All proteins included in the training dataset have less than 25% sequence similarity with sequences in the testing sets, adopting a stringent homology-reduction procedure. Results show that ISPRED-SEQ performs at the state-of-the-art, reporting MCC scores higher than those obtained by other approaches in all the benchmarks performed.

The ISPRED-SEQ web server is freely accessible at https://ispredws.biocomp.unibo.it.

Materials and Methods

Datasets

Training dataset. For training the ISPRED-SEQ network we used a set of protein chains derived from a dataset available in literature[28] and already adopted, after some filtering steps, to train the DELPHI method.[14] The DELPHI dataset comprises 9,982 protein chain sequences extracted from the PDB and sharing no more than 25% pairwise sequence identity. Moreover, the sequences in the training set are also non-redundant (25% identity) with respect to all the sequences included in the independent test datasets (see next section). Starting from this set, we further restricted the number of protein sequences by filtering out all the chains (as in the correspondent UniProt file) having a coverage with the associated PDB structure/s less than 80%, in order to validate PPI annotation on structural experimental evidence. After this filtering step, we ended up with 6,066 protein sequences comprising 1,757,296 residues.

Annotation of PPI sites was then retrieved from the original data available from[28] and manually curated. Starting from the PDB structure of the complex, a residue of a given chain is defined in interaction if the distance between an atom of the residue and an atom of another residue in a different chain is below a given distance threshold, which routinely is set equal to the total sum of the van der Waals' radii of the two atoms plus 0.5 Å[28]. PPI annotations are available for the complete UniProt protein sequences after combining all interaction sites obtained from multiple protein complexes in which each protein is represented, adopting SIFTS[29] for the relative mapping of PDB and UniProt.[28] Overall, our dataset comprises 285,751 interaction sites, corresponding to about 16% of the whole set of residues.

We split the training dataset into 10 different subsets for performing the 10-fold cross validation procedure. Before splitting, we further clustered the sequences at 25% sequence identity and 40% alignment coverage using MMseqs2.[30] The cross-validation split was then performed by randomly distributing complete clusters (instead of individual sequences) among the different subsets. This step is required to capture residual local redundancies between pair of sequences that could have survived the first redundancy reduction performed during dataset construction.

Independent test datasets. To evaluate generalization performance of ISPRED-SEQ and to compare it with other state-of-the-art approaches we used four different independent test sets widely used in literature for comparative evaluation of tools.[9,14–15,26–27] Supplementary Table 1 provide an overview of all datasets used in this study.

The first dataset comprises 448 protein chains used in a review comparing different tools for protein interaction site prediction from sequence.[27] The aim of the authors was to collect data including not only protein–protein interaction sites, but also annotations for DNA, RNA and small-ligand binding sites. For this reason, the dataset was obtained starting from the BioLip database,[31] collecting nucleic-acid and ligand binding site annotations. For the set of proteins retrieved from BioLip, authors also extracted protein–protein interaction sites by

analyzing corresponding protein complexes available in the PDB. Protein interaction sites are identified using the same definition adopted for the training set (see above). Internal redundancy of the dataset was set to 25% pairwise sequence identity using the Blastclust tool.[32] We refer to this dataset to as the Dset448.

The second dataset used here is referred to as the Dset335 and it is a subset of the Dset448 introduced in[14] for sake of comparing the methods DELPHI and DLPred.[33] The 335 sequences included in the dataset are indeed selected such that they are non-redundant at 25% sequence identity with the DLPred training set, hence enabling a fair comparison with this method. We used Dset335 to also include DLPred in our benchmark.

The third and fourth datasets, referred to as HomoTE and HeteroTE, respectively, were introduced by Hou and coauthors[9,26]. Recently, these sets were also used for evaluating the performance of the PIPENN prediction tool.[15] HomoTE and HeteroTE include 479 and 48 protein chains from homomeric and heteromeric complexes, respectively. Interface residues are defined in HomoTE and HeteroTE using a slightly different definition based on the computation of Accessible Surface Area (ASA) before and after complex formation: interacting residues are those whose ASA value undergoes a change upon complex formation[26]. Nevertheless, as highlighted in literature,[34] this definition provides very similar or equal interaction interfaces as those based on inter-chain distances.

ISPRED-SEQ implementation

The ISPRED-SEQ general architecture is depicted in Figure 1. Starting from a protein sequence, ISPRED-SEQ input is constructed using two alternative protein language models: i) ESM1-b[17], an encoder-only transformer model trained on about 27 million sequences from UniRef50[35], and ii) ProtT5[18], a sequence-to-sequence model derived from the T5 architecture[36], trained on the large Big Fantastic Database (BFD)[37] comprising 2.1 billion sequences and fine-tuned on the UniRef50 database.

For each residue in the input sequence, ESM1-b and ProtT5 provide embeddings of dimension 1280 and 1024, respectively. These are then concatenated to form a single vector comprising 2304 components for each residue.

Since ESM1-b can only accept input sequences of length lower than 1022, all longer sequences are split into non-overlapping chunks of equal length. After this step, the sequence embedding is reconstructed by concatenating all the chunks.

The joint embedding (ESM1-b + ProtT5) is then processed using a four-layer network. The first layer is a 1-dimensional convolutional neural network with 2304 filters (the number of filters is set as to be equal to the input dimension) and a filter width of 31, corresponding to a window comprising 31 flanking residues and centered at each residue position. The positional output of the convolutional layers is processed by two dense, fully connected layers with 128 and 32 hidden units, respectively. The final output consists of a single unit with a sigmoid activation function. Each residue is classified as interaction site if the output value is greater or equal to 0.5, as not in interaction otherwise.

For sake of assessing the contribution of the input encoding, we also trained alternative models based on different types of inputs, including: the sequence one-hot encoding, providing 20 values per residue, the position-specific scoring matrix (PSSM), computed using two runs of HHblits[38] against the UniClust30 database[39] and providing 20 values per residues, ESM1-b embedding only (1280 values per residue) and ProtT5 embedding only (1024 values per residue). For all the models trained, we adopted the same architecture shown in Figure 1, and changing the number of convolutional filters to be equal to the input dimension (20 for one-hot and PSSMs, 1280 for ESM1-b and 1024 for ProtT5).

Training is performed using minibatches of 64 residues adopting an early stopping procedure that halts the training after 10 epochs without a decrease in the validation loss. The loss that we implemented is a binary cross-entropy and we adopted an Adam optimizer.[40]

To fix all the hyperparameters of the model we performed a grid search using a strict 10-fold cross validation. After that, we retrained the final model on the whole training dataset, and we evaluated it on the different benchmark sets.

Scoring measures

The following measures were used to score performance of the different methods:

- Accuracy (Q_2):

$$Q_2 = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

- Precision:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

- Recall:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

- F1-score, the harmonic mean of precision and recall:

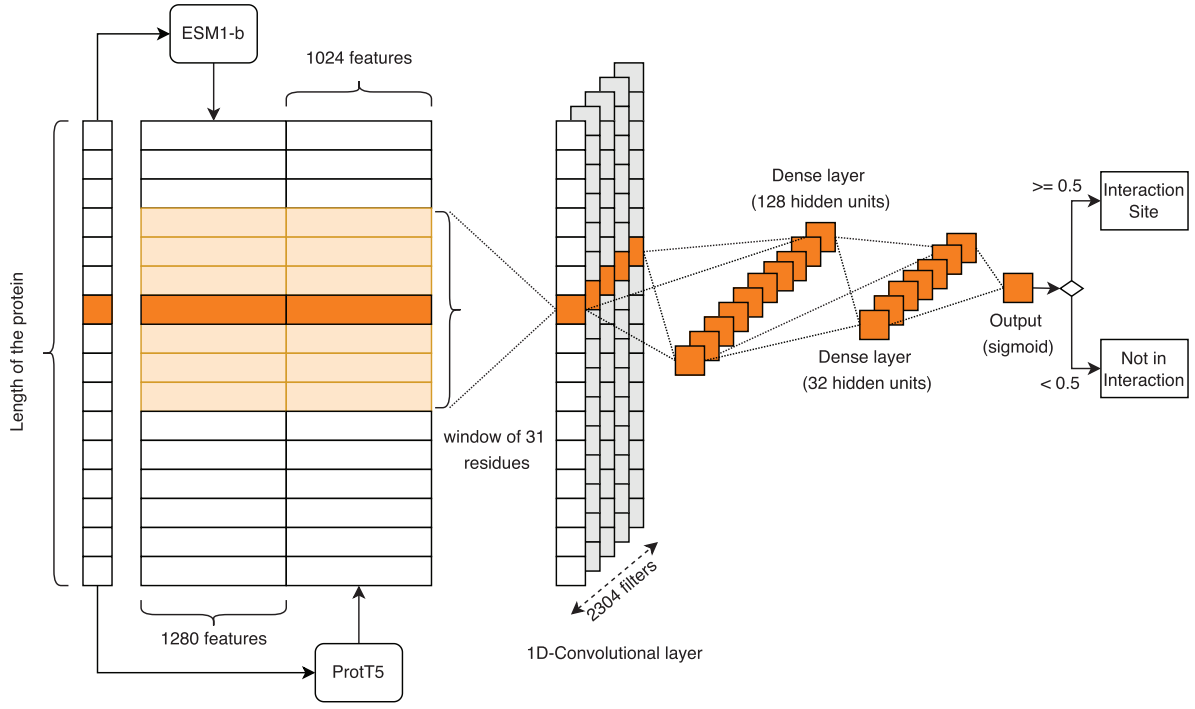


Figure 1. The ISPRED-SEQ deep network architecture. The input sequence is encoded using the two language models (ESM1-b[17] and ProtT5[18]), producing a joint embedding of 2304 features. These are processed using a 1D-Convolutional layer with 2304 filters of size 31. The convolutional output is then processed by two fully connected Dense layers with 128 and 32 hidden units, respectively. The final output is a single unit with sigmoid activation function: each residue is classified as Interaction Site when the output value is greater or equal to 0.5, non-interaction site otherwise (see Materials and Methods for details).

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

- Area Under the Receiver Operating Characteristic Curve (ROC-AUC).
- Matthews Correlation Coefficient (MCC):

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \quad (5)$$

Routinely, the probability value discriminating between positive and negative predictions is set to 0.5. For benchmarking on blind test sets ISPRED-SEQ towards other approaches[14–15,27], we adopted a methodological strategy previously described.[27] According to this procedure, for each of the different methods, a method-specific threshold is introduced to set the number of positive predictions equal to the number of real positive examples.[14–15,27] This procedure allows comparing different methods on the same number of predictions.[27] AUC values are however independent of this procedure.

Results

ISPRED-SEQ performance

For fine tuning ISPRED-SEQ, we tested the network architecture using a 10-fold cross-validation procedure to compare different input encodings. Specifically, we evaluated five different models trained on different inputs, including: i) the sequence one-hot encoding, ii) the sequence profile, iii) the ESM1-b embedding only, iv) the ProtT5 embedding only and v) the joint embedding obtained combining ESM1-b and ProtT5. Supplementary Table 2 lists the results.

Models incorporating canonical features (one-hot and sequence profiles) are both outperformed by embedding-based approaches. MCCs obtained with embedding-based approaches score with values above 0.30 and higher that the 0.14 value obtained with only the sequence profile as input (Supplementary Table 2). Data are shown in Supplementary Table 2, obtained adopting a cross validation procedure. This highlights the effectiveness of language model representations in the task of predicting PPI sites. The two different language models (ESM1-b and ProtT5) provide similar contributions individually achieving

comparable MCC scores (0.30 and 0.31, respectively). When combined, the value of MCC is 0.34 (adopting a threshold for positive predictions equal to 0.5), suggesting that the ESM1-b and ProtT5 are complementary, and their combination is advantageous for the problem at hand. This conclusion is further supported by data shown in Supplementary Table 3, where we can observe that predictions made using the two models disagree on roughly 25% of the data (on 14.9% ProtT5 is correct, on 9.3% ESM1-b is correct).

We compared ISPRED-SEQ with state-of-the-art tools, including DELPHI[14], PIPENN,[15] PITHIA[16], SCRIBER[11], SSRWF[8], CRFPPI[41] and LORIS.[42] Table 1 shows the results, and Supplementary Table 4 shows more details regarding the tools adopted for the comparison.

Performance of all methods, with the exclusion of ISPRED-SEQ, are extracted from literature[14–15]. Specifically, performance on Dset448 and Dset335 for DELPHI, SCRIBER, SSRWF, CRFPPI and LORIS are derived from[14], results of PIPENN in all datasets are taken from the original reference paper,[15] and results fro PITHIA are taken from.[16]

All the benchmarked methods provide numerical prediction scores representing the propensity of each input residue to be a PPI site. A threshold must be set to obtain a binary prediction. To compare ISPRED-SEQ performance with other state-of-the-art tools, we adopted the same strategy described in[14–15] and defined in[27] by which binary predictions are obtained using a different threshold for each method so that the number of positive predictions (FP + TP) is equal to the number of real positive examples (TP + FN), or equivalently FP = FN. For our ISPRED-SEQ, performance measures obtained using this strategy are labelled as "th⇒FP = FN" in Table. 1. A direct comparison with the state-of-the-art methods is therefore possible. For sake of completeness, we also show ISPRED-SEQ score obtained using the threshold of 0.5 on the output prediction score. This threshold assumes a probability meaning for the output of ISPRED-SEQ and it is the one adopted in the web server.

Regardless of the method adopted for choosing the threshold, Table 1 indicates that ISPRED-SEQ outperforms all the methods in all the considered datasets. In the Dset448 (the most recent and complete dataset released in literature so far[27]), ISPRED-SEQ achieves a MCC value of 0.39, seven percentage points higher than the one obtained by the second top-performing method, PITHIA.

In the Homo-TE dataset containing homomeric interfaces, ISPRED-SEQ reaches a MCC value of 0.46, again significantly higher than the one registered by PIPENN. Performance on the small Hetero-TE, containing only 48 chains, are lower. However, also in this case, ISPRED-SEQ

Table 1 Comparative benchmark on different independent test sets.

Method	Dataset	MCC	F1	Precision	Recall	Q2	AUC
ISPRED-SEQ (th = 0.5)°	Dset448	0.34	0.42	0.29	0.78	0.71	0.82
ISPRED-SEQ (th⇒FP = FN)°	Dset448	0.39	0.47	0.47	0.47	0.86	0.82
PITHIA[16] *	Dset448	0.32	0.41	0.41	0.41	0.84	0.78
DELPHI[14] †	Dset448	0.27	0.37	0.37	0.37	0.83	0.74
PIPENN[15] ‡	Dset448	0.25	0.39	0.39	0.39	0.79	0.73
SCRIBER[11] †	Dset448	0.23	0.33	0.33	0.33	0.82	0.72
SSWRF[8] †	Dset448	0.18	0.29	0.29	0.29	0.81	0.69
CRFPPI[41] †	Dset448	0.15	0.27	0.26	0.27	0.81	0.68
LORIS[42] †	Dset448	0.15	0.27	0.26	0.26	0.81	0.66
ISPRED-SEQ (th = 0.5)°	Dset335	0.33	0.40	0.27	0.77	0.72	0.82
ISPRED-SEQ (th⇒FP = FN)°	Dset335	0.39	0.46	0.46	0.46	0.87	0.82
PITHIA[16] *	Dset335	0.30	0.38	0.38	0.38	0.85	0.76
DELPHI[14] †	Dset335	0.28	0.36	0.36	0.36	0.85	0.75
SCRIBER[11] †	Dset335	0.23	0.32	0.32	0.32	0.84	0.72
DLPred[33] †	Dset335	0.21	0.31	0.31	0.31	0.84	0.72
ISPRED-SEQ (th = 0.5)°	Homo_TE	0.42	0.56	0.42	0.83	0.71	0.84
ISPRED-SEQ (th⇒FP = FN)°	Homo_TE	0.46	0.58	0.58	0.58	0.81	0.84
PIPENN[15] ‡	Homo_TE	0.34	0.49	0.49	0.49	0.77	0.77
ISPRED-SEQ (th = 0.5)°	Hetero_TE	0.20	0.27	0.17	0.68	0.65	0.72
ISPRED-SEQ (th⇒FP = FN)°	Hetero_TE	0.16	0.24	0.24	0.24	0.86	0.72
PIPENN[15] ‡	Hetero_TE	0.11	0.20	0.20	0.20	0.85	0.66

* Data taken from.[16]
† Data taken from.[14]
‡ Data taken from.[15]
° th, threshold value (see Materials and Methods). Performance of all methods different from ISPRED-SEQ are reported considering a prediction threshold that makes equal the numbers of false positive and false negative predictions.[27] Results of ISPRED-SEQ adopting the same strategy are reported (th⇒ FP = FN) as well as those obtained adopting a probability threshold equal to 0.5 (th = 0.5).

outperforms the other tested method (PIPENN) by 5 percentage points, considering the MCC value.

Independently of the procedure adopted for evaluating the scoring indexes, ISPRED-SEQ overpasses the performance of all other methods. This is also evident when considering the AUC values reported in Table 1, totally independent of the strategy adopted for the other scoring indexes.

The ISPRED-SEQ web server

ISPRED-SEQ webserver is available at https://ispredws.biocomp.unibo.it/. The server input interface accepts a single protein sequence in FASTA format with length ranging between 50 and 5000 residues. Upon submission, the user is redirected to the page where results will be available after job completion. The page automatically refreshes every 60s and shows to the user the current status of the job (queued or running). The server also provides the user with a universal job identifier, which can be thereafter used to retrieve job results. The result page (Figure 2) provides information about the job, including i) the identifier, ii) submission and completion time, iii) protein ID, iv) protein length and v) counts of positive and negative predictions. After that, the output of the predictor is shown

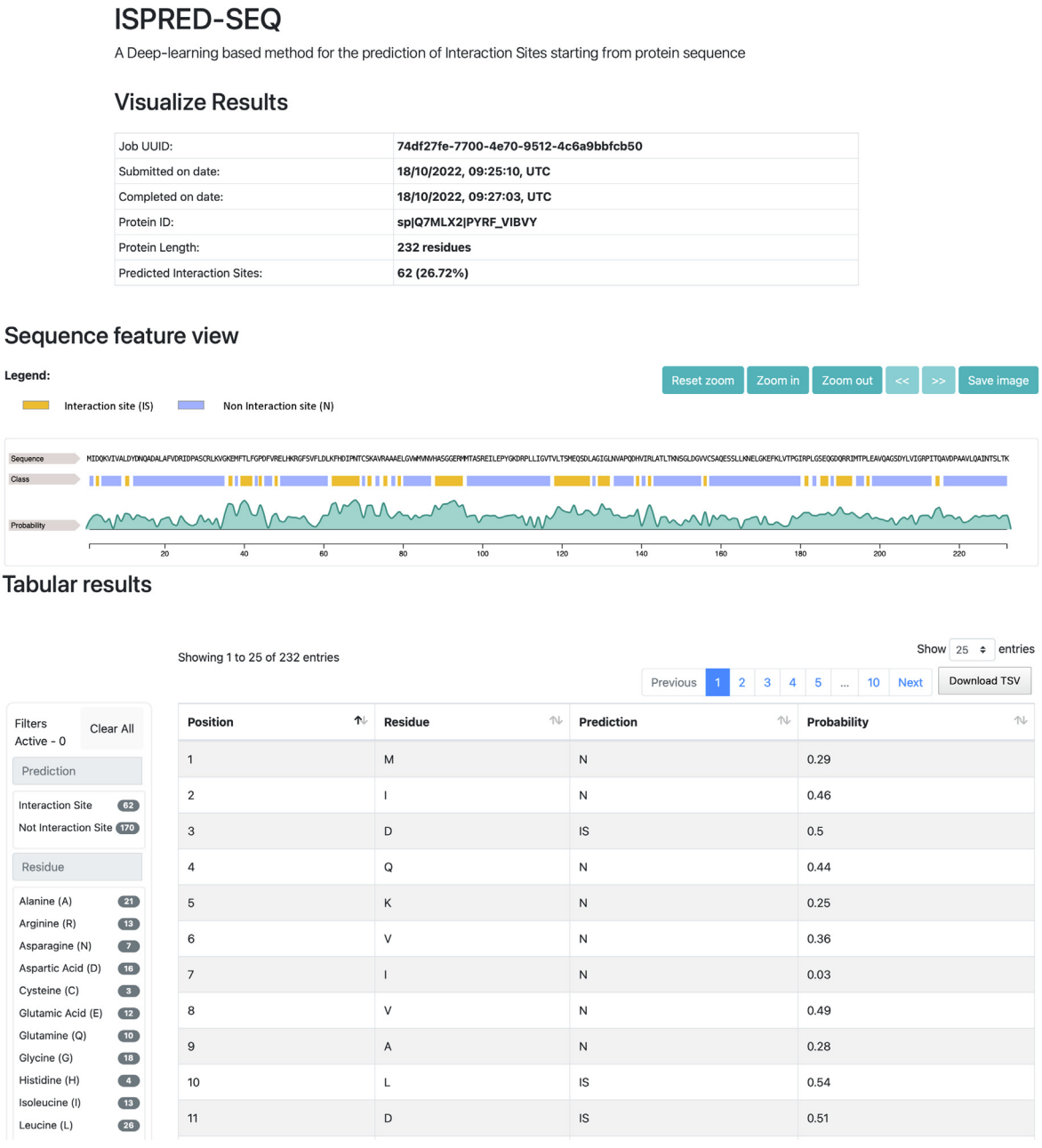


Figure 2. The ISPRED-SEQ web server output page.

using an interactive viewer. This allows to visualize the whole PPI-site probability profile computed for each residue during the procedure. The page highlights in yellow the predicted PPIs along the sequence. Results are as well summarized in tabular format (Figure 2).

Conclusions

In this paper we present ISPRED-SEQ, a novel method for the prediction of PPI sites from sequence. ISPRED-SEQ novelty is the adoption of input encodings based on embeddings generated by two state-of-the-art protein language models, ESM1-b and ProtT5. In our tests, residue representations based on embeddings outperform canonical feature descriptors such as one-hot encoding and sequence profiles. The scoring index values, although good, still need improvement. However, the major bias is due to the fact that still we do not have a complete picture of all the possible PPIs in a cell, as discussed before.[1–2]

We evaluated ISPRED-SEQ using several independent datasets released in literature and compared its performances against recently state-of-the-art approaches, also based on deep-learning algorithms. In all the tests performed, ISPRED-SEQ significantly outperforms top-scoring methods, reaching MCC scores of 0.39 on recent benchmark datasets containing more than 300 proteins.

We propose ISPRED-SEQ as a valuable tool for the characterization of protein interface residues starting from the protein primary sequence.

We released ISPRED-SEQ as a publicly accessible web server available at https://ispredws.biocomp.unibo.it.

CRediT authorship contribution statement

Matteo Manfredi: Data curation, Formal analysis, Software, Validation, Writing – original draft, Writing – review & editing. **Castrense Savojardo:** Conceptualization, Supervision, Formal analysis, Software, Validation, Writing – original draft, Writing – review & editing. **Pier Luigi Martelli:** Conceptualization, Supervision, Formal analysis, Validation, Writing – original draft, Writing – review & editing. **Rita Casadio:** Conceptualization, Supervision, Formal analysis, Validation, Writing – original draft, Writing – review & editing.

DATA AVAILABILITY

All data are avaiable at: https://ispredws.biocomp.unibo.it/sequence/about/download/

DECLARATION OF COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The work was supported by PRIN2017 grant (project 2017483NH8_002), delivered to CS by the Italian Ministry of University and Research. We acknowledge ELIXIR-IIB, the Italian node of the ELIXIR infrastructure.

Appendix A. Supplementary Data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jmb.2023.167963.

Received 21 November 2022;
Accepted 10 January 2023;
Available online xxxx

Keywords:
end-to-end models;
embedding;
deep neural networks;
PPI prediction;
protein sequence

† The authors equally contributed to the work.

References

1. Li, S., Wu, S., Wang, L., Li, F., Jiang, H., Bai, F., (2022). Recent advances in predicting protein–protein interactions with the aid of artificial intelligence algorithms. *Curr. Opin. Struct. Biol.* **73**, 102344. https://doi.org/10.1016/j.sbi.2022.102344.
2. Casadio, R., Martelli, P.L., Savojardo, C., (2022). Machine learning solutions for predicting protein–protein interactions. *WIREs Comput. Mol. Sci.*. https://doi.org/10.1002/wcms.1618.
3. Lyon, A.S., Peeples, W.B., Rosen, M.K., (2021). A framework for understanding the functions of biomolecular condensates across scales. *Nat. Rev. Mol. Cell Biol.* **22**, 215–235. https://doi.org/10.1038/s41580-020-00303-z.
4. Rodrigues, J.P.G.L.M., Karaca, E., Bonvin, A.M.J.J., (2015). Information-driven structural modelling of protein-protein interactions. *Methods Mol. Biol.* **1215**, 399–424. https://doi.org/10.1007/978-1-4939-1465-4_18.
5. Savojardo, C., Fariselli, P., Martelli, P.L., Casadio, R., (2017). ISPRED4: interaction sites PREDiction in protein structures with a refining grammar model. *Bioinformatics* **33**, 1656–1663. https://doi.org/10.1093/bioinformatics/btx044.

6. Ofran, Y., Rost, B., (2003). Predicted protein-protein interaction sites from local sequence information. *FEBS Lett.* **544**, 236–239.
7. Murakami, Y., Mizuguchi, K., (2010). Applying the Naïve Bayes classifier with kernel density estimation to the prediction of protein-protein interaction sites. *Bioinformatics* **26**, 1841–1848. https://doi.org/10.1093/bioinformatics/btq302.
8. Wei, Z.-S., Han, K., Yang, J.-Y., Shen, H.-B., Yu, D.-J., (2016). Protein-protein Interaction Sites Prediction by Ensembling SVM and Sample-weighted Random Forests. *Neurocomput.* **193**, 201–212. https://doi.org/10.1016/j.neucom.2016.02.022.
9. Hou, Q., De Geest, P.F.G., Vranken, W.F., Heringa, J., Feenstra, K.A., (2017). Seeing the trees through the forest: sequence-based homo- and heteromeric protein-protein interaction sites prediction using random forest. *Bioinformatics* **33**, 1479–1487. https://doi.org/10.1093/bioinformatics/btx005.
10. Hou, Q., De Geest, P.F.G., Griffioen, C.J., Abeln, S., Heringa, J., Feenstra, K.A., (2019). SeRenDIP: SEquential REmasteriNg to DerIve profiles for fast and accurate predictions of PPI interface positions. *Bioinformatics* **35**, 4794–4796. https://doi.org/10.1093/bioinformatics/btz428.
11. Zhang, J., Kurgan, L., (2019). SCRIBER: accurate and partner type-specific prediction of protein-binding residues from proteins sequences. *Bioinformatics* **35**, i343–i353. https://doi.org/10.1093/bioinformatics/btz324.
12. Qiu, J., Bernhofer, M., Heinzinger, M., Kemper, S., Norambuena, T., Melo, F., Rost, B., (2020). ProNA2020 predicts protein–DNA, protein–RNA, and protein–protein binding proteins and residues from sequence. *J. Mol. Biol.* **432**, 2428–2443. https://doi.org/10.1016/j.jmb.2020.02.026.
13. Zeng, M., Zhang, F., Wu, F.-X., Li, Y., Wang, J., Li, M., (2020). Protein–protein interaction site prediction through combining local and global features with deep neural networks. *Bioinformatics* **36**, 1114–1120. https://doi.org/10.1093/bioinformatics/btz699.
14. Li, Y., Golding, G.B., Ilie, L., (2021). DELPHI: accurate deep ensemble model for protein interaction sites prediction. *Bioinformatics* **37**, 896–904. https://doi.org/10.1093/bioinformatics/btaa750.
15. Stringer, B., de Ferrante, H., Abeln, S., Heringa, J., Feenstra, K.A., Haydarlou, R., (2022). PIPENN: protein interface prediction from sequence with an ensemble of neural nets. *Bioinformatics* **38**, 2111–2118. https://doi.org/10.1093/bioinformatics/btac071.
16. Hosseini, S., Ilie, L., (2022). PITHIA: Protein Interaction Site Prediction Using Multiple Sequence Alignments and Attention. *Int. J. Mol. Sci.* **23**, 12814. https://doi.org/10.3390/ijms232112814.
17. Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., et al., (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. U S A.* **118** https://doi.org/10.1073/pnas.2016239118. e2016239118.
18. Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., et al., (2021). ProtTrans: towards cracking the language of lifes code through self-supervised deep learning and high performance computing. *IEEE Trans. Pattern. Anal. Mach. Intell.* **PP** https://doi.org/10.1109/TPAMI.2021.3095381.
19. Heinzinger, M., Elnaggar, A., Wang, Y., Dallago, C., Nechaev, D., Matthes, F., Rost, B., (2019). Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinf.* **20**, 723. https://doi.org/10.1186/s12859-019-3220-8.
20. Bepler, T., Berger, B., (2021). Learning the protein language: Evolution, structure, and function. *Cell Syst.* **12**, 654–669.e3. https://doi.org/10.1016/j.cels.2021.05.017.
21. Stärk, H., Dallago, C., Heinzinger, M., Rost, B., (2021). Light attention predicts protein location from the language of life. *Bioinformat. Adv.* **1**, vbab035. https://doi.org/10.1093/bioadv/vbab035.
22. Littmann, M., Heinzinger, M., Dallago, C., Olenyi, T., Rost, B., (2021). Embeddings from deep learning transfer GO annotations beyond homology. *Sci. Rep.* **11**, 1160. https://doi.org/10.1038/s41598-020-80786-0.
23. Teufel, F., Almagro Armenteros, J.J., Johansen, A.R., Gíslason, M.H., Pihl, S.I., Tsirigos, K.D., Winther, O., Brunak, S., et al., (2022). SignalP 6.0 predicts all five types of signal peptides using protein language models. *Nat. Biotechnol.* **40**, 1023–1025. https://doi.org/10.1038/s41587-021-01156-3.
24. Mahbub, S., Bayzid, M.S., (2022). EGRET: edge aggregated graph attention networks and transfer learning improve protein–protein interaction site prediction. *Brief. Bioinform.* **23**, bbab578. https://doi.org/10.1093/bib/bbab578.
25. Singh, J., Litfin, T., Singh, J., Paliwal, K., Zhou, Y., (2022). SPOT-Contact-LM: improving single-sequence-based prediction of protein contact map using a transformer language model. *Bioinformatics* **38**, 1888–1894. https://doi.org/10.1093/bioinformatics/btac053.
26. Hou, Q., Dutilh, B.E., Huynen, M.A., Heringa, J., Feenstra, K.A., (2015). Sequence specificity between interacting and non-interacting homologs identifies interface residues – a homodimer and monomer use case. *BMC Bioinf.* **16**, 325. https://doi.org/10.1186/s12859-015-0758-y.
27. Zhang, J., Kurgan, L., (2018). Review and comparative assessment of sequence-based predictors of protein-binding residues. *Brief. Bioinform.* **19**, 821–837. https://doi.org/10.1093/bib/bbx022.
28. Zhang, J., Ma, Z., Kurgan, L., (2019). Comprehensive review and empirical analysis of hallmarks of DNA-, RNA- and protein-binding residues in protein chains. *Brief. Bioinform.* **20**, 1250–1268. https://doi.org/10.1093/bib/bbx168.
29. Dana, J.M., Gutmanas, A., Tyagi, N., Qi, G., O'Donovan, C., Martin, M., Velankar, S., (2019). SIFTS: updated Structure Integration with Function, Taxonomy and Sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins. *Nucleic Acids Res.* **47**, D482–D489. https://doi.org/10.1093/nar/gky1114.
30. Steinegger, M., Söding, J., (2017). MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.* **35**, 1026–1028. https://doi.org/10.1038/nbt.3988.
31. Yang, J., Roy, A., Zhang, Y., (2013). BioLiP: a semi-manually curated database for biologically relevant ligand–protein interactions. *Nucleic Acids Res.* **41**, D1096–D1103. https://doi.org/10.1093/nar/gks966.
32. Altschul, S., (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.

Nucleic Acids Res. **25**, 3389–3402. https://doi.org/10.1093/nar/25.17.3389.

33. Zhang, B., Li, J., Quan, L., Chen, Y., Lü, Q., (2019). Sequence-based prediction of protein-protein interaction sites by simplified long short-term memory network. *Neurocomputing* **357**, 86–100. https://doi.org/10.1016/j.neucom.2019.05.013.
34. Ezkurdia, I., Bartoli, L., Fariselli, P., Casadio, R., Valencia, A., Tress, M.L., (2009). Progress and challenges in predicting protein-protein interaction sites. *Brief. Bioinformatics.* **10**, 233–246. https://doi.org/10.1093/bib/bbp021.
35. Suzek, B.E., Wang, Y., Huang, H., McGarvey, P.B., Wu, C. H., (2015). UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* **31**, 926–932. https://doi.org/10.1093/bioinformatics/btu739.
36. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J., (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21** 140:5485–140:5551.
37. Steinegger, M., Mirdita, M., Söding, J., (2019). Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nat. Methods.* **16**, 603–606. https://doi.org/10.1038/s41592-019-0437-4.
38. Steinegger, M., Meier, M., Mirdita, M., Vöhringer, H., Haunsberger, S.J., Söding, J., (2019). HH-suite3 for fast remote homology detection and deep protein annotation. *BMC Bioinf.* **20**, 473. https://doi.org/10.1186/s12859-019-3019-7.
39. Mirdita, M., von den Driesch, L., Galiez, C., Martin, M.J., Söding, J., Steinegger, M., (2017). Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic Acids Res.* **45**, D170–D176. https://doi.org/10.1093/nar/gkw1081.
40. Kingma D. P. & Ba, J. (2017). Adam: A Method for Stochastic Optimization, ArXiv:1412.6980 [Cs]. http://arxiv.org/abs/1412.6980 (accessed October 19, 2020).
41. Wei, Z.-S., Yang, J.-Y., Shen, H.-B., Yu, D.-J., (2015). A cascade random forests algorithm for predicting protein-protein interaction sites. *IEEE Trans. Nanobiosci.* **14**, 746–760. https://doi.org/10.1109/TNB.2015.2475359.
42. Dhole, K., Singh, G., Pai, P.P., Mondal, S., (2014). Sequence-based prediction of protein-protein interaction sites with L1-logreg classifier. *J. Theor. Biol.* **348**, 47–54. https://doi.org/10.1016/j.jtbi.2014.01.028.