Predicting protein-membrane interfaces of peripheral membrane proteins using ensemble machine learning

Alexios Chatzigoulas[1,2,*] and Zoe Cournia[1,*]

[1]Biomedical Research Foundation, Academy of Athens, 4 Soranou Ephessiou, 11527 Athens, Greece, [2]Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, 15784 Athens, Greece

*To whom correspondence should be addressed.

Abstract

Motivation: Abnormal protein-membrane attachment is involved in deregulated cellular pathways and in disease. Therefore, the possibility to modulate protein-membrane interactions represents a new promising therapeutic strategy for peripheral membrane proteins that have been considered so far undruggable. A major obstacle in this drug design strategy is that the membrane binding domains of peripheral membrane proteins are usually not known. The development of fast and efficient algorithms predicting the protein-membrane interface would shed light into the accessibility of membrane-protein interfaces by drug-like molecules.

Results: Herein, we describe an ensemble machine learning methodology and algorithm for predicting membrane-penetrating residues. We utilize available experimental data in the literature for training 21 machine learning classifiers and a voting classifier. Evaluation of the ensemble classifier accuracy produced a macro-averaged F1 score = 0.92 and an MCC = 0.84 for predicting correctly membrane-penetrating residues on unknown proteins of an independent test set.

Availability and implementation: The python code for predicting protein-membrane interfaces of peripheral membrane proteins is available at https://github.com/zoecournia/DREAMM.

Contact: zcournia@bioacademy.gr

Supplementary information: Supplementary data are available.

1 Introduction

Membrane proteins are topologically divided in transmembrane proteins that are permanently attached in the interior of the membrane, peripheral membrane proteins that associate non-covalently with the surface of the membrane, and lipid-anchored proteins that attach to the membrane with a covalent bond (Boes *et al.*, 2021). Peripheral membrane proteins are essential in cellular processes such as transporting substances across the cell membrane, activating proteins and enzymes, regulating signal transduction, and other functions (Boes *et al.*, 2021; Monje-Galvan and Klauda, 2016). Abnormal protein-membrane attachment due to membrane-binding domain mutations and peripheral membrane protein over- or under activation are involved in deregulated cellular pathways and in disease (Boes *et al.*, 2021; Costeira-Paulo *et al.*, 2018; Hobbs *et al.*, 2016; Lashuel *et al.*, 2013; Mirsaeidi *et al.*, 2016; Mirza and Zahid, 2018; Segers *et al.*, 2007). Hence, the possibility to modulate protein-membrane interactions represents a promising therapeutic strategy for many disease indications and in particular for targeting membrane proteins that have been considered undruggable such as the membrane-anchored KRAS protein, which is implicated in over 30% of cancer types (Cox *et al.*, 2014; Kessler *et al.*, 2019), α-synuclein, which is a main pathological hallmark of Parkinson's disease (de Oliveira and Silva, 2019; Hijaz and Volpicelli-Daley, 2020), and lipid kinases such as PI3Kα, which is the most frequently mutated kinase and present in a variety of tumors (Yang *et al.*, 2019) with one of its hotspot mutations, H1047R, acting on altering the protein's association with the cell membrane (Cournia and Chatzigoulas, 2020; Gabelli *et al.*, 2010; Gkeka *et al.*, 2014; Gkeka *et al.*, 2015).

The feasibility of targeting protein-membrane interfaces is supported by the fact that peripheral membrane proteins contain a membrane-binding domain with cavities that could be potentially targeted by small molecules (Segers *et al.*, 2007; Sudhahar *et al.*, 2008). The literature reports the feasibility of targeting the protein-membrane interface indicating that therapeutic targets binding transiently to the membrane can be targeted with small molecules, and that inhibitors of protein-membrane interactions may be identified (Chen *et al.*, 2015; Li and Buck, 2020; Liu *et al.*, 2010; Nawrotek *et al.*, 2019; Nicolaes *et al.*, 2014; Segers *et al.*, 2007; Spiegel *et al.*, 2004). However, these examples are only limited compared to the overall drug design efforts of the community indicating that the accessibility of protein-membrane interfaces by small molecules has been so far unexplored possibly due to the complexity of the interface, the limited protein-membrane structural information, and the absence of tools and workflows to automate the drug design process at the protein-membrane interface. Moreover, protein-membrane interaction sites of peripheral membrane proteins are commonly undiscovered; hence, the first step into modulating the protein-membrane interface is their identification.

Several efforts towards the design of tools that detect protein-membrane regions, domains, and lipid-binding sites have appeared (Bhardwaj *et al.*, 2006; Nastou *et al.*, 2016; Scott *et al.*, 2006; Sharikov *et al.*, 2008); however, these are mainly applied to directly to 1D protein sequences without considering the protein structural information and in many cases the web links are outdated (Bhardwaj *et al.*, 2006; Scott *et al.*, 2006;

Sharikov *et al.*, 2008). To our knowledge, only two methodologies, which predict these interaction sites from the 3D protein structure, are currently publically available: the Positioning of Proteins in Membrane (PPM) (Lomize *et al.*, 2006; Lomize *et al.*, 2011) and the Membrane Optimal Docking Area (MODA) (Kufareva *et al.*, 2014). PPM combines an anisotropic solvent representation of the lipid bilayer, an all atom representation of a solute, and a universal solvation model, calculating rotational and translational positions of transmembrane and peripheral membrane proteins in membranes (Lomize *et al.*, 2006; Lomize *et al.*, 2011). MODA is based on the protein-protein interface predictor PIER (Kufareva *et al.*, 2007), which builds a set of evenly distributed points at 5 Å from one another and from the protein surface, defining each patch as the set of all protein surface atoms. Then, it calculates a score based on atom solvent-accessible surface area (SASA) and atom type specific weights, and transfers the patch membrane propensity scores to surface residues, thereby predicting which residues contact the cell membranes.

Herein, we present an automated prediction algorithm using ensemble machine learning, which identifies membrane-binding interfaces with high accuracy (macro-averaged F_1 score = 0.92 and an MCC = 0.84) taking as input the 3D peripheral membrane protein coordinates and demonstrates better accuracy than existing methods.

2 Methods

2.1 Data Preparation

To construct the dataset, we use 54 peripheral membrane with known 3D structures and experimentally known membrane-penetrating residues, retrieved from extensive literature search. For the dataset generation, protein structures were prepared by deleting unwanted chains and co-crystallized solvent atoms, adding missing side chain atoms, and converting non-standard amino acids to their standard equivalents using HTMD (Doerr et al., 2016). In case of NMR-resolved structures, the first model of the NMR ensemble was kept. Then, the dataset was split in a training set (~85% of the dataset, Table S1) and a test set (~15% of the dataset, Table S2). Finally, a dataset of 12.805 residues consisting of the training set samples and a dataset of 2.177 residues consisting of the test set were assembled. These samples were labeled in two categories, the membrane-penetrating and the non-penetrating residues, leading to a highly imbalanced classification problem (supervised learning), where the membrane-penetrating residues comprise ~1.3% of the total samples in the training set, which is addressed in section 2.4.

2.2 Feature extraction

Lipid interfaces of proteins possess specific chemical and topological properties, *i.e.*, amphipathic alpha-helices flanked by a flexible hinge or loop regions, being solvent-exposed or containing cationic patches around aromatic and aliphatic regions that anchor to the negatively-charged bilayers (Johnson and Cornell, 1999; Whited and Johs, 2015), which are regularly found in the inner leaflet of the plasma membrane (Lemmon, 2008). Therefore, two driving forces for protein-membrane association need to be considered: a) long range electrostatic interactions that drive protein-membrane proximity, b) hydrophobic interactions that facilitate protein anchoring to the hydrophobic fatty acid tails of the lipid bilayer (Cho and Stahelin, 2005; Whited and Johs, 2015).

Firstly, the ProtDCal tool was facilitated, which calculates 2788 thermodynamic, topographic, and property-based features (Ruiz-Blanco *et al.*, 2015), and then, based on the protein-membrane interactions, physico-chemical and biochemical features of the aforementioned membrane-associated properties were generated for each protein residue in the training set by employing computational methods in Python as described below. Protein-membrane regions mainly consist of amphipathic alpha-helices or hydrophobic loops, therefore, the Define Secondary Structure of Proteins (DSSP) program was utilized to define the secondary structure (Kabsch and Sander, 1983), parsing the PDB files with the Python package Biopython (Cock *et al.*, 2009; Hamelryck and Manderick, 2003). Additionally, DSSP measures geometrical properties, for example, backbone torsion angles, which were also kept as features. One-hot encoding, a technique that transforms each unique value in a categorical feature into a new binary feature, was applied on the amino acid and the secondary structure features to transform them from categorical to numerical features. Alongside the secondary structure, the solvent exposure is a significant property of the membrane-penetrating amino acids. The FreeSASA tool was utilized for the calculation of the SASA (Mitternacht, 2016), and the MSMS tool for calculating the residue and Cα depths (Sanner *et al.*, 1996). One advantage of FreeSASA lies in the separation of residue SASA into polar and non-polar SASA. Non-polar SASA combines solvent exposure with hydrophobicity, which is necessary for forming hydrophobic interactions with the inner leaflet of the plasma membrane. Therefore, the Wimley-White whole-residue interface and octanol hydrophobicity scales were additionally utilized as features (Wimley *et al.*, 1996; Wimley and White, 1996). To consider electrostatic interactions, PDB2PQR was utilized to calculate residue charges and protonation states using default parameters (Dolinsky *et al.*, 2007; Dolinsky *et al.*, 2004), and MDAnalysis for reading the resulting PQR file (Michaud-Agrawal *et al.*, 2011).

Moreover, additional properties (features), which could potentially be connected to the protein-membrane association, were sought. The sequence profiling tool HHblits was applied in order to calculate the conservation score (Remmert *et al.*, 2011), through HTMD (Doerr *et al.*, 2016), searching the Uniclust30 database (Mirdita *et al.*, 2017). To consider residue flexibility, the ProDy package was used to calculate squared fluctuations utilizing two different Elastic Network Models, the Gaussian Network Model and the Anisotropic Network Model (Bakan *et al.*, 2014; Bakan *et al.*, 2011). Finally, the feature space was enriched with the residue radius of gyration, which was calculated with MDAnalysis (Michaud-Agrawal *et al.*, 2011), the number of each amino acid type of neighboring amino acids, and their total number in a Cα – Cα distance of 7 Å.

To consider the surrounding amino acid properties of each residue, the mean values of the aforementioned features were calculated, for each residue and the residues at a 7 Å distance from the protein α carbon atoms (Cα – Cα). In this way, the 3D space is taken into consideration ensuring that the information of the surrounding residues is included in the feature space, for example, the mean hydrophobicity and charge from the neighboring residues, and the number of nearby lysine, arginine, and histidine residues, leading to 2880 features in total.

2.3 Feature and data selection

Training machine learning algorithms in datasets with redundant samples and features is computationally inefficient. Discarding redundant information is essential to reducing the data size and hence the computational effort as in this case the training set consists of 12,805 samples (residues) and 2,880 features. To reduce the sample size, and especially the sample size of the majority class, the solvent inaccessible residues of the proteins were removed because only interfacial residues penetrate the membrane, using a cutoff of 2.5 Å for the residue depth feature produced by MSMS (Sanner *et al.*, 1996), leading to 8,720 samples. Moreover, only residues that penetrate the hydrocarbon core of the membrane were retained, according to the octanol-interface scale (White, 2003), which is derived from the experimentally determined Wimley-White whole-residue interface and octanol hydrophobicity scales (Wimley *et al.*, 1996; Wimley and White, 1996), leaving out the A, S, N, G, E, D, K, R, and H amino acids, further reducing the sample size of the majority class. The C and Q amino acids were also removed, as only a few cases were present in the membrane-penetrating category. In the end, the training set was reduced to 3010 samples, reducing the required computational time to process the dataset. Moreover, the imbalanced classes problem was attenuated because the sample size of the majority class was reduced significantly, with the membrane-penetrating residues consisting the ~5.5% of the total samples.

Subsequently, features with zero standard deviation (all feature values are the same) were discarded in order to remove redundant features, further reducing the dataset to 2252 features. The Pearson pairwise correlation was measured leaving out features with more than 95% correlation; this led to 883 features. Utilizing two different tree-based machine learning algorithms, the extremely randomized trees method (Geurts *et al.*, 2006), using the scikit-learn Python package (Pedregosa *et al.*, 2011) and the leaf-wise gradient boosting decision tree algorithm LightGBM (Ke *et al.*, 2017), features that where less important than the 1% of the total feature importance in both algorithms were removed leading to a total of 727 features. The hyper-parameter space for these classifiers was fine-tuned by combining the randomized and grid search 5-fold cross validation approaches as explained in the “Ensemble machine learning methodology” section. As anticipated, the most important features were expressing hydrophobicity and solvent exposure (Figure S1, Table S3).

2.4 Imbalanced classes problem

Reducing the data sample size improved the imbalanced classes problem but the increase of the membrane-penetrating class to almost five-fold still produced imbalanced classes. When the size of one class outnumbers the size of the other, machine learning algorithms will under-predict the infrequent class. To balance the two classes three techniques were utilized. The first one is to use weights on the samples, emphasizing the minority samples. The second is to over-sample the minority class using algorithms that generate synthetic samples based on the feature values of the minority class samples until both classes consist of equal number of samples. Finally, the third technique is to under-sample the majority class with sample selection methods until again both classes have equal number of samples. Notably, for a number of machine learning algorithms using weights is similar to over-sampling the minority class with duplicate samples. From the initial training set of 3,010 samples, two training sets of 5,686 samples using two different over-sampled techniques, two training sets of 334 samples using two different under-sampled techniques, and one more training set of 4,287 samples using a combination of over- and under-sampling methods were produced utilizing the imbalanced-learn Python toolbox (Lemaitre *et al.*, 2017). Using this procedure, six different training sets were produced: the initial training set using weights, two training sets using over-sampled techniques, two training sets using under-sampled techniques, and one training set using a combination of over- and under-sampling methods.

The first over-sampled training set was generated utilizing the Synthetic Minority Over-sampling Technique (SMOTE) technique, which synthesizes artificial new minority instances between existing real minority instances (Chawla *et al.*, 2002). The second over-sampled training set utilizing the Adaptive Synthetic (ADASYN) sampling algorithm is similar to SMOTE but attempts to infer which points in the minority class would be the most difficult for a model to learn and attempts to place a higher ratio of synthetic data close to these points (He *et al.*, 2008). For the undersampled training sets, the first one was generated using the Condensed Nearest Neighbor (CNN) method, which iteratively uses the 1 nearest neighbor rule to decide if a sample should be removed or not (Hart, 1968) and the second one was based on the Instance Hardness Threshold (IHT), which is a technique where a machine learning algorithm is trained on the training set and removes the samples with the lowest probabilities (Smith *et al.*, 2014). For the IHT method the scikit-learn gradient boosting classifier (Friedman, 2001) was utilized to estimate the instance hardness of the samples. Because over-sampling using SMOTE may lead to generation of noisy samples, a sixth training set was built, where SMOTE was followed by the Edited Nearest Neighbors undersampling method (SMOTEENN), which applies a nearest-neighbors algorithm to clean the training set (Tomek, 1976). By removing samples which do not agree enough with their neighborhood (the majority of the 5 closest neighbors belong to another class) the outliers that were generated from SMOTE were removed. These six training sets can be visualized in Figure S2 reduced to two dimensions.

2.5 Ensemble machine learning methodology

For each one of the six training sets, 21 machine learning classifiers were trained: 19 from the scikit-learn Python package (Pedregosa et al., 2011), the LightGBM classifier (Ke et al., 2017), and the XGBoost classifier (Chen and Guestrin, 2016). The hyper-parameters of these classifiers were optimized to discover the best parameters that fit the data for each classifier (Bergstra and Bengio, 2012; Claesen and De Moor, 2015). Specifically, for every training set and for every classifier the randomized search cross-validation technique was performed using 5 folds in a wide range of parameter values, training hundreds of thousands models. Subsequently, iteratively exhaustive searches were performed (grid search cross-validation with 5 folds) in a small range of parameter values in the vicinity of the best parameter space determined from the randomized search cross-validation, which led to a set of optimal parameters. Notably, in the over- and under-sampled datasets, over- and under-sampling was performed in each fold separately to avoid information leak in the validation fold.

To assess the performance of the classifier models for both the randomized and grid search cross-validation procedures, the macro-averaged harmonic mean of the precision and recall, F_1 score, was chosen as a metric. Recall expresses the amount of correct predicted true positives (Eq. 1), while precision expresses the predicted true positives that are actually true (Eq. 2). The general formula of F score is derived based on a positive real variable β, where β determines the importance of recall over precision (Eq. 3). When $\beta = 1$ (F_1 score), recall and precision are weighted equally (Eq. 4), when $\beta < 1$ more weight is given in precision and when $\beta > 1$ recall is favored.

$$recall = \frac{true\ positives}{true\ positives + fal\quad negatives} \quad (1) \quad precision = \frac{true\ positives}{true\ positives + fals\quad positives} \quad (2)$$

$$F_\beta = (1+\beta^2)\frac{precision * recall}{(\beta^2 * precision) + reca} \quad (3) \quad F_1 = 2 * \frac{precision * recall}{precision + recall} \quad (4)$$

Subsequently, for each training set the resulting predictions of the aforementioned classifiers were imputed as input to a meta-classifiers (second-level classifier). The voting classifier, which classifies a sample based on the majority voting of the first-level classifiers (Littlestone and Warmuth, 1994), and the stacking classifier, which trains a classifier on the output of the first-level classifiers in order to compute the final prediction (Wolpert, 1992), were employed using the Python library mlxtend (Raschka, 2018). In both meta-classifiers, all possible combinations of the first-level classifiers were examined to discover the best classifier combination. Every classifier combination was tested using the independent test set with known protein-membrane residues (Table S2) to measure the combination with the best performance. Subsequently, considering that not every residue in the dataset was experimentally tested, resulting in membrane-penetrating residues marked as non-penetrating, the best models were manually inspected in order to assess their false positives, and the final model was chosen based on F_2 score (see Results). Finally, the features were again inspected resulting in 167 redundant features that do not affect the behavior of the newly developed ensemble classifier and leading to 560 features, which accelerated the feature extraction process. A schematic representation of the above procedure is illustrated in Figure 1.

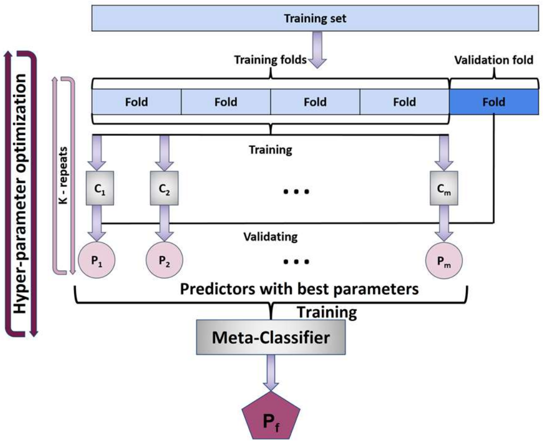

Fig. 1. For each of the six datasets, we optimized the hyper-parameter space of 21 classifiers using 5-fold cross-validation on the training set. The predictions of the models with the best F_1 score from these classifiers were provided as input to meta-classifiers. Given the F_2 score on the test set the best meta-classifier was kept as the final predictor.

3 Results

The first-level classifiers providing the most accurate results were those trained in the initial dataset using weights. For these classifiers and initial dataset, several second-level classifiers exhibited better performance than the individual classifiers in terms of F_1 score, precision/recall area under the curve (PR AUC), Matthews correlation coefficient (MCC), and other metric scores. The Receiver Operating Characteristics (ROC) AUC, which is regularly used in literature, was not considered as it may be misleading for highly imbalanced classification problems such as our case (Davis and Goadrich, 2006; Saito and Rehmsmeier, 2015). Then, results from the top second-level classifiers were subject to manual inspection and the best was selected according to the F_2 score to emphasize on recall. Although, seemingly, it is natural to prioritize on precision as in our case false positives are more critical than false negatives, manual inspection of the false positive results of the top second-level meta-classifiers indicated that these could actually be true positive membrane-penetrating residues as they are adjacent to amino acid residues that are membrane-penetrating or aligned with them to adjacent loops (see below). Finally, the best performing second-level classifier was the voting classifier for a combination consisting of five classifiers, the linear discriminant analysis, the logistic regression, the linear support vector classifier, the decision tree classifier, and the light gradient boosting machine. Various metric scores of the 21 first-level classifiers and the chosen second-level classifier for the initial dataset using weights can be viewed in Table S4.

The test set predictions can be viewed in Figure 2 and Table S5, where ~2/3 of the false positive residues are in fact correct predictions as they are located in the protein-membrane interface adjacent to true positives or on adjacent loops. For example, in retinoid isomerohydrolase, residue F262 is in a 4 Å distance from the experimentally confirmed membrane-penetrating residues; for the glycolipid transfer protein, residues I143 and Y153 are next to and aligned with W142. In other examples, i.e. the cholesterol-regulated Start protein 4, residue M196 although it is located in different loop, it is aligned with L124, as well as for the phosphatidylinositol transfer protein beta isoform, where M74 is in a different loop but aligned with the experimentally-confirmed membrane-penetrating residues W202 and W203; for the PH domain of the ceramide transfer protein, I37 and W40 are next to W33 and Y36, and F81 is aligned with them in an adjacent loop. Considering that these predictions are in fact located at the protein-membrane interface, they can be considered as true positives and the macro-averaged F_1 score increases from 0.86 to 0.92 and MCC from 0.71 to 0.84.

Moreover, the ensemble classifier was applied in the test set keeping all residue types. To reduce false positive non-hydrophobic amino acids, only those residues with center of mass (COM) distance of 14 Å from at least one of the predicted hydrophobic amino acids were kept (Table S6).

Furthermore, the performance of the classifier was compared to two computational tools that predict protein-membrane interfaces from 3D structures, the PPM web-server (Lomize *et al.*, 2012), which additionally predicts the orientation of proteins in membranes, and the MODA web-server (Kufareva *et al.*, 2014), in the test set without performing data selection (Table S6). Generally, the predictions of every tool was fairly accurate in predicting the protein-membrane regions with our classifier outperforming them in some cases. Specifically, for the retinoid isomerohydrolase homodimer, PPM falsely predicted the orientation (probably affected by the missing chains) and placed the protein in an orientation in which only one monomer was in contact with the membrane instead of both, while our classifier and MODA correctly predicted the protein-membrane regions in both chains (Figure S3). For the VSTx1 toxin, every tool predicted the protein-membrane region but falsely predicted W25. Moreover, our classifier falsely predicted as membrane-penetrating the residues near the N- and C-terminal and MODA falsely predicted the beta sheet on the opposite side of the protein-membrane interface and the C-terminal region (Figure S4). For cytotoxin 2, all tools performed the same with a few false positives for our classifier and MODA in the 41-46 region (Figure S5). Similarly, for sphingomyelinase C all tools recognized the experimentally-verified membrane-penetrating residues W284 and F285, with PPM and MODA recognizing residues in distant loops that are aligned with the experimentally-verified protein-membrane region suggesting a multiregional interaction with the membrane, which is probably true, according to the proposed membrane binding model (Figure S6). For the glycolipid transfer protein, our classifier and MODA provided similar results correctly identifying the membrane-penetrating α-helix, with MODA falsely predicting the C-terminus and our classifier residue Y81 to be membrane-penetrating residues. PPM also suggested the insertion of the membrane-penetrating α-helix, but with the addition of the P40-P44 region (Figure S7). Likewise for the cholesterol-regulated Start protein 4, all tools predicted correctly the experimentally-verified residue L124 with our classifier and PPM additionally predicting the 196-200 region, MODA falsely predicting the C-terminus, and our classifier falsely predicting residue W91 (Figure S8). Finally, for the PH domain of the ceramide transfer

Predicting protein-membrane interfaces

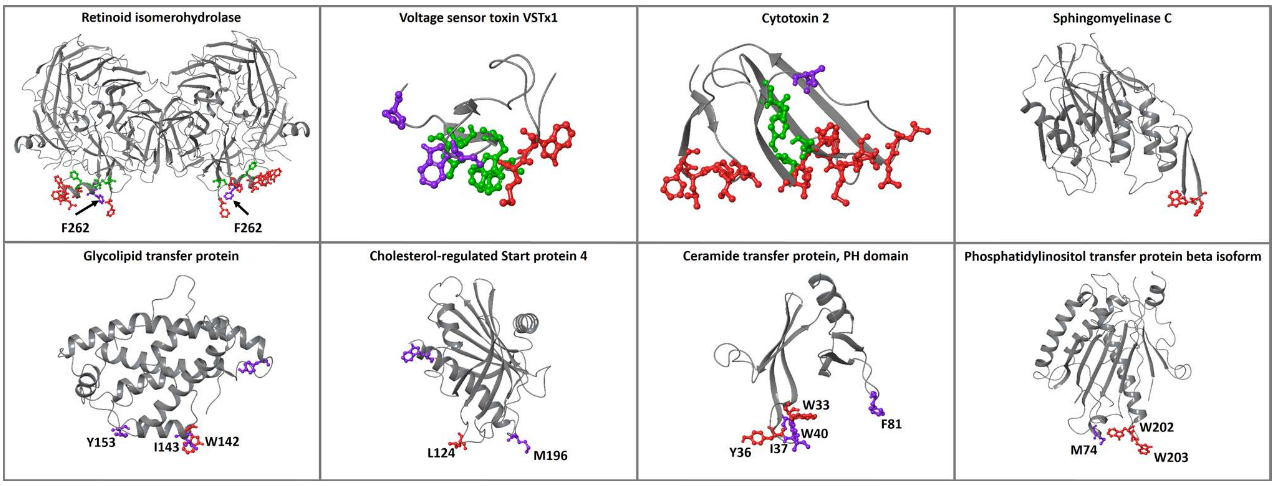
Retinoid isomerohydrolase
F262
F262
Voltage sensor toxin VSTx1
Cytotoxin 2
Sphingomyelinase C
Glycolipid transfer protein
Y153
I143
W142
Cholesterol-regulated Start protein 4
L124
M196
Ceramide transfer protein, PH domain
W33
W40
F81
Y36
I37
Phosphatidylinositol transfer protein beta isoform
W202
M74
W203

Fig. 2. Proteins of the test set. The experimental membrane-penetrating residues predicted from the ensemble classifier are depicted in red, the experimental membrane-penetrating residues not predicted from the classifier are depicted in green, and the residues predicted from the classifier that have not been experimentally-verified are depicted in purple.

protein and the phosphatidylinositol transfer protein beta isoform, the outcome was similar and correct for all tools (Figures S9-S10).

The performance of the ensemble classifier was tested with additional protein use cases with known membrane-penetrating regions (Figure 3), and the results were compared with PPM and MODA (Table S7). For the cases of cholesterol oxidase, cytochrome P450 3A4, monoglyceride lipase MGLL, L-amino acid deaminase, and intestinal fatty acid binding protein, all tools correctly identified the protein-membrane regions (Figures S11, S12, S14, S19, and S20). For the 9-cis-epoxycarotenoid dioxygenase 1, chloroplastic, all tools predicted the protein-membrane regions, with the exception of our classifier, which predicted the insertion of one of the two parallel amphipathic helices, instead of both (Figure S13). Similarly, for the dihydroorotate dehydrogenase, all tools predicted the protein-membrane regions, with our classifier falsely identifying the residue W362 and MODA the region 245-247 (Figure S15). For phosphatase PTEN, our classifier successfully identified the protein-membrane region 263-269 of the C2 domain and the region around the L42 residue of phosphatase in the same membrane plane. MODA also identified the same phosphatase region, however it falsely identified the opposite side of the C2 domain as a protein-membrane region. PPM also falsely identified the opposite side of the C2 domain suggesting an orientation, which is opposite to the actual membrane orientation (Figure S16). For (S)-mandelate dehydrogenase, the protein-membrane region was correctly identified by all tools, but our classifier and MODA also identified amino acids 53-56 to be membrane-

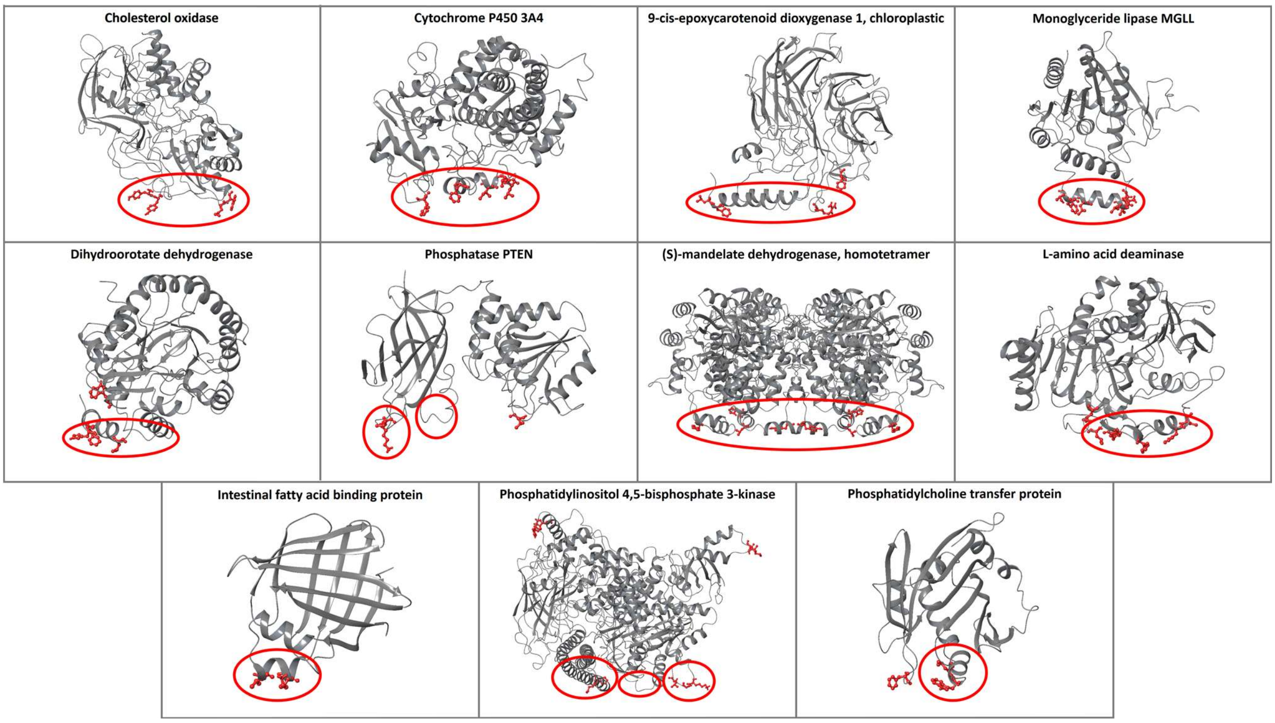
Cholesterol oxidase
Cytochrome P450 3A4
9-cis-epoxycarotenoid dioxygenase 1, chloroplastic
Monoglyceride lipase MGLL
Dihydroorotate dehydrogenase
Phosphatase PTEN
(S)-mandelate dehydrogenase, homotetramer
L-amino acid deaminase
Intestinal fatty acid binding protein
Phosphatidylinositol 4,5-bisphosphate 3-kinase
Phosphatidylcholine transfer protein

Fig 3. The predictions of the ensemble classifier for proteins with known protein-membrane interface regions. The membrane-penetrating residues predicted from the classifier are depicted in red, and the experimental membrane-penetrating regions are denoted with red circles.

penetrating (Figure S17); these residues are actually at the protein-protein interaction interface in the homotetramer of (S)-mandelate dehydrogenase and are not misclassified if we perform the predictions in the homotetramer biological assembly (Figure S18). For the phosphatidylinositol 4,5-bisphosphate 3-kinase alpha (PI3Kα), all tools predicted residues 232-233 as a protein-membrane interface, which is not in agreement with the experimental results, but is in fact the region where PI3Kα binds to K-RAS (Ras binding domain). Additionally, our classifier and MODA successfully identified the p110α 863-872 and the iSH2 512-525 regions, but falsely identified the 498-508 region, which links the C2 domain with the helical domain (Figure S21). The membrane orientation resulting from PPM is different from the one proposed through mutagenesis experiments (Gabelli *et al.*, 2010). Finally, for the phosphatidylcholine transfer protein, all tools provided the same results identifying the experimentally proven region 184-193 and an adjacent loop, while MODA additionally predicted loop 147-148 to be membrane binding, which is in the same plane with the other two membrane binding regions (Figure S22).

Furthermore, the ensemble classifier was applied in the full structure of prothrombin (PDB: 5EDM (Pozzi et al., 2016)), comparing the results of identified membrane-protein interfaces with PPM and MODA (Figure 4). All tools predicted the GLA domain 3-5 region as a membrane contacting region, which is natural for the classifier as the GLA domain of prothrombin was in the training set. Additionally, the classifier predicted the residues Y93, W398, and V458 and MODA predicted the residues Y93, Y377, R379, and R484, suggesting an orientation parallel to the actual membrane orientation, opposed to the perpendicular suggested by PPM. Y93 is a key prothrombin residue, which is essential for stabilizing the closed form and protects the active site pocket of the protease domain (Chinnaraj et al., 2018). In the prothrombin closed form (PDB: 6BJR (Chinnaraj et al., 2018)), Y93 inserts its aromatic side chain into the binding pocket of the protease domain engaging W547 (W533 of 5EDM) and forms pi-pi interactions (Figure S23). The results provided by our classifier and MODA suggest that Y93 penetrates into the membrane, further indicating that when prothrombin engages the membrane the open form is favored with Y93 anchoring the membrane and opening the active site.

Finally, the ensemble classifier was also tested for the prediction of the protein-membrane interfaces of nine transmembrane enzymes described in Ref (Dufrisne et al., 2017), which include a soluble domain performing extracellular catalysis. In agreement with experimental results, our classifier predicted numerous residues that lie in the hydrophobic lipid bilayer core, along with membrane-interacting extracellular residues (Figure S24).

4 Discussion

Drug design of protein-membrane interfaces for peripheral membrane proteins has been so far neglected due to the complexity of the interface and the lack of a suitable workflows and simulation technology capable of implementing this drug design strategy. Furthermore, protein-membrane interaction regions of peripheral membrane proteins are commonly unknown, and only a few rational methodologies exist that predict these regions from the 3D protein structure. To assist in protein-membrane interface recognition, a novel ensemble machine learning classifier is described trained in experimental data retrieved from extensive literature search.

The ensemble classifier results are accurate in predicting correctly the membrane-penetrating residues in the test set, providing with a macro-averaged F_1 score = 0.92 and an MCC = 0.84. Additionally, in a different independent dataset with experimentally known protein-membrane regions, our classifier correctly identified membrane-penetrating residues in these regions with a few false positive predictions. In addition, comparative results demonstrated that our classifier performed similarly, and in some cases better, than the only two available web-servers that predict protein-membrane interaction sites from the 3D protein structure, PPM and MODA. Moreover, our classifier successfully predicted the membrane-penetrating residues and the residues that lie in the hydrophobic core of the lipid bilayer in transmembrane proteins containing a soluble catalytic domain.

Commonly, during development of computational tools several obstacles may emerge. In this case the first was the low number of the peripheral membrane proteins with experimentally known membrane-penetrating residues described in literature. The second and more crucial was the small number of residues that were tested experimentally in many of these proteins resulting in membrane-penetrating residues marked as non-penetrating, which in turn resulted in confusing the classifiers during the training process and making the selection of the best ensemble classifier strenuous. Moreover, the performance metrics, e.g. F scores, do not reflect the actual accuracy of the ensemble classifier, which is higher, and subsequently, a numerical comparison of our classifier, PPM, and MODA results is infeasible, and can be only assessed visually.

Manual inspection of false positive results revealed that several amino acids were located near the N- or C-termini, or near missing loops, probably because the area is more solvent exposed. Intriguingly, residues falsely predicted as membrane-penetrating are found to be implicated in protein-protein interactions. The assumption that protein-membrane interactions are similar to protein-protein interactions was also deduced by (Kufareva et al., 2014), who adapted their protein-protein interaction interface prediction PIER algorithm (Kufareva et al., 2007) in MODA. It

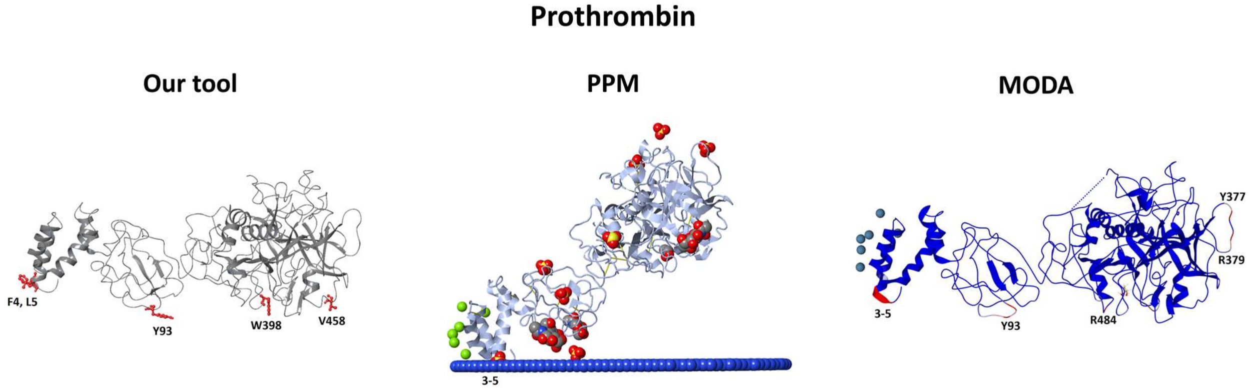

Fig 4. Comparison of the predictions provided from our classifier, PPM, and MODA for the open form of the prothrombin protein. Our classifier and MODA propose a parallel to the membrane orientation, suggesting the insertion of Y93 to the membrane, which in turn opens the active site of the protease domain (Figure S23).

should be noted that the current implementation predictions depend on structural information, therefore, in the case where in the 3D protein structure the membrane-penetrating residues are in the bulk of the protein and a conformational change is necessary to face them towards the membrane, or the protein is intrinsically disordered, the ensemble classifier would not be able to predict them.

Finally, it is noteworthy to mention that the membrane-penetrating residues are in many cases significant for the allosteric control of binding sites similar to prothrombin case. It is suggested that in the protein-membrane interfaces a binding pocket exists (Chen et al., 2015; Li and Buck, 2020; Liu et al., 2010; Nawrotek et al., 2019; Nicolaes et al., 2014; Segers et al., 2007; Spiegel et al., 2004). We strongly believe that these binding pockets could act allosterically being connected with the active site and could be able to modulate the open/closed form equilibrium (Chatzigoulas and Cournia, 2021; Cournia and Chatzigoulas, 2020) or even block protein function by disrupting the protein-membrane interactions.

Funding

This work was supported by the State Scholarships Foundation (IKY) [MIS-5000432 to A.C.]; the Hellenic Foundation for Research and Innovation (H.F.R.I.) [1780 to Z.C.]; and the National Infrastructures for Research and Technology S.A. (GRNET S.A.) [pr008033_gpu-Mem-Surf, pr008033_thin-Mem-Surf, and pr009022_gpu-Mem-Surf to A.C.].

Conflict of Interest: none declared.

References

Bakan, A. *et al.* (2014) Evol and ProDy for bridging protein sequence evolution and structural dynamics, *Bioinformatics*, **30**, 2681-2683.

Bakan, A. *et al.* (2011) ProDy: protein dynamics inferred from theory and experiments, *Bioinformatics*, **27**, 1575-1577.

Bergstra, J. and Bengio, Y. (2012) Random Search for Hyper-Parameter Optimization, *J. Mach. Learn. Res.*, **13**, 281-305.

Bhardwaj, N. *et al.* (2006) Structural bioinformatics prediction of membrane-binding proteins, *J. Mol. Biol.*, **359**, 486-495.

Boes, D.M. *et al.* (2021) Peripheral Membrane Proteins: Promising Therapeutic Targets across Domains of Life, *Membranes (Basel)*, **11**.

Chatzigoulas, A. and Cournia, Z. (2021) Rational design of allosteric modulators: Challenges and successes, *WIREs Comput. Mol. Sci.*, e1529.

Chawla, N.V. *et al.* (2002) SMOTE: Synthetic minority over-sampling technique, *J Artif Intell Res*, **16**, 321-357.

Chen, L. *et al.* (2015) Novel inhibitors induce large conformational changes of GAB1 pleckstrin homology domain and kill breast cancer cells, *PLoS Comput. Biol.*, **11**, e1004021.

Chen, T. and Guestrin, C. (2016) XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, San Francisco, California, USA, pp. 785-794.

Chinnaraj, M. *et al.* (2018) Structure of prothrombin in the closed form reveals new details on the mechanism of activation, *Sci Rep*, **8**, 2945.

Cho, W. and Stahelin, R.V. (2005) Membrane-protein interactions in cell signaling and membrane trafficking, *Annu. Rev. Biophys. Biomol. Struct.*, **34**, 119-151.

Claesen, M. and De Moor, B. (2015) Hyperparameter search in machine learning, *arXiv preprint arXiv:1502.02127*.

Cock, P.J. *et al.* (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics, *Bioinformatics*, **25**, 1422-1423.

Costeira-Paulo, J. *et al.* (2018) Lipids Shape the Electron Acceptor-Binding Site of the Peripheral Membrane Protein Dihydroorotate Dehydrogenase, *Cell chemical biology*, **25**, 309-317 e304.

Cournia, Z. and Chatzigoulas, A. (2020) Allostery in membrane proteins, *Curr. Opin. Struct. Biol.*, **62**, 197-204.

Cox, A.D. *et al.* (2014) Drugging the undruggable RAS: Mission possible?, *Nat. Rev. Drug Discov.*, **13**, 828-851.

Davis, J. and Goadrich, M. (2006) The relationship between Precision-Recall and ROC curves. *Proceedings of the 23rd international conference on Machine learning*. Association for Computing Machinery, Pittsburgh, Pennsylvania, USA, pp. 233–240.

de Oliveira, G.A.P. and Silva, J.L. (2019) Alpha-synuclein stepwise aggregation reveals features of an early onset mutation in Parkinson's disease, *Communications biology*, **2**, 374.

Doerr, S. *et al.* (2016) HTMD: High-Throughput Molecular Dynamics for Molecular Discovery, *J Chem Theory Comput*, **12**, 1845-1852.

Dolinsky, T.J. *et al.* (2007) PDB2PQR: expanding and upgrading automated preparation of biomolecular structures for molecular simulations, *Nucleic Acids Res.*, **35**, W522-525.

Dolinsky, T.J. *et al.* (2004) PDB2PQR: an automated pipeline for the setup of Poisson-Boltzmann electrostatics calculations, *Nucleic Acids Res.*, **32**, W665-667.

Dufrisne, M.B. *et al.* (2017) Structural basis for catalysis at the membrane-water interface, *Biochim Biophys Acta Mol Cell Biol Lipids*, **1862**, 1368-1385.

Friedman, J.H. (2001) Greedy Function Approximation: A Gradient Boosting Machine, *The Annals of Statistics*, **29**, 1189-1232.

Gabelli, S.B. *et al.* (2010) Structural effects of oncogenic PI3Kalpha mutations, *Curr. Top. Microbiol. Immunol.*, **347**, 43-53.

Geurts, P. *et al.* (2006) Extremely randomized trees, *Mach Learn*, **63**, 3-42.

Gkeka, P. *et al.* (2014) Investigating the structure and dynamics of the PIK3CA wild-type and H1047R oncogenic mutant, *PLoS Comput. Biol.*, **10**, e1003895.

Gkeka, P. *et al.* (2015) Exploring a non-ATP pocket for potential allosteric modulation of PI3Kalpha, *J. Phys. Chem. B*, **119**, 1002-1016.

Hamelryck, T. and Manderick, B. (2003) PDB file parser and structure class implemented in Python, *Bioinformatics*, **19**, 2308-2310.

Hart, P. (1968) The condensed nearest neighbor rule (Corresp.), *IEEE Trans. Inf. Theory*, **14**, 515-516.

He, H.B. *et al.* (2008) ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning, *IEEE IJCNN*, 1322-1328.

Hijaz, B.A. and Volpicelli-Daley, L.A. (2020) Initiation and propagation of alpha-synuclein aggregation in the nervous system, *Mol. Neurodegener.*, **15**, 19.

Hobbs, G.A. *et al.* (2016) RAS isoforms and mutations in cancer at a glance, *J. Cell Sci.*, **129**, 1287-1292.

Johnson, J.E. and Cornell, R.B. (1999) Amphitropic proteins: regulation by reversible membrane interactions (review), *Mol. Membr. Biol.*, **16**, 217-235.

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

Ke, G. *et al.* (2017) Lightgbm: A highly efficient gradient boosting decision tree, *Adv Neural Inf Process Sys*, 3146-3154.

Kessler, D. *et al.* (2019) Drugging an undruggable pocket on KRAS, *Proc. Natl. Acad. Sci. U. S. A.*, **116**, 15823-15829.

Kufareva, I. *et al.* (2007) PIER: protein interface recognition for structural proteomics, *Proteins*, **67**, 400-417.

Kufareva, I. *et al.* (2014) Discovery of novel membrane binding structures and functions, *Biochem. Cell Biol.*, **92**, 555-563.

Lashuel, H.A. *et al.* (2013) The many faces of alpha-synuclein: from structure and toxicity to therapeutic target, *Nat. Rev. Neurosci.*, **14**, 38-48.

Lemaitre, G. *et al.* (2017) Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning, *J Mach Learn Res*, **18**, 559–563.

Lemmon, M.A. (2008) Membrane recognition by phospholipid-binding domains, *Nat. Rev. Mol. Cell Biol.*, **9**, 99-111.

Li, Z. and Buck, M. (2020) Computational Design of Myristoylated Cell-Penetrating Peptides Targeting Oncogenic K-Ras.G12D at the Effector-Binding Membrane Interface, *J. Chem. Inf. Model.*, **60**, 306-315.

Littlestone, N. and Warmuth, M.K. (1994) The Weighted Majority Algorithm, *Information and Computation*, **108**, 212-261.

Liu, Z. *et al.* (2010) Trp2313-His2315 of factor VIII C2 domain is involved in membrane binding: structure of a complex between the C2 domain and an inhibitor of membrane binding, *J. Biol. Chem.*, **285**, 8824-8829.

Lomize, A.L. *et al.* (2006) Positioning of proteins in membranes: a computational approach, *Protein Sci.*, **15**, 1318-1333.

Lomize, A.L. *et al.* (2011) Anisotropic solvent model of the lipid bilayer. 2. Energetics of insertion of small molecules, peptides, and proteins in membranes, *J. Chem. Inf. Model.*, **51**, 930-946.

Lomize, M.A. *et al.* (2012) OPM database and PPM web server: resources for positioning of proteins in membranes, *Nucleic Acids Res.*, **40**, D370-376.

Michaud-Agrawal, N. *et al.* (2011) MDAnalysis: a toolkit for the analysis of molecular dynamics simulations, *J. Comput. Chem.*, **32**, 2319-2327.

Mirdita, M. *et al.* (2017) Uniclust databases of clustered and deeply annotated protein sequences and alignments, *Nucleic Acids Res.*, **45**, D170-D176.

Mirsaeidi, M. *et al.* (2016) Annexins family: insights into their functions and potential role in pathogenesis of sarcoidosis, *J Transl Med*, **14**, 89.

Mirza, F.J. and Zahid, S. (2018) The Role of Synapsins in Neurological Disorders, *Neurosci Bull*, **34**, 349-358.

Mitternacht, S. (2016) FreeSASA: An open source C library for solvent accessible surface area calculations, *F1000Res*, **5**, 189.

Monje-Galvan, V. and Klauda, J.B. (2016) Peripheral membrane proteins: Tying the knot between experiment and computation, *Biochim. Biophys. Acta*, **1858**, 1584-1593.

Nastou, K.C. *et al.* (2016) MBPpred: Proteome-wide detection of membrane lipid-binding proteins using profile Hidden Markov Models, *Biochim. Biophys. Acta*, **1864**, 747-754.

Nawrotek, A. *et al.* (2019) PH-domain-binding inhibitors of nucleotide exchange factor BRAG2 disrupt Arf GTPase signaling, *Nat. Chem. Biol.*, **15**, 358-366.

Nicolaes, G.A. *et al.* (2014) Rational design of small molecules targeting the C2 domain of coagulation factor VIII, *Blood*, **123**, 113-120.

Pedregosa, F. *et al.* (2011) Scikit-learn: Machine Learning in Python, *J Mach Learn Res*, **12**, 2825-2830.

Pozzi, N. *et al.* (2016) How the Linker Connecting the Two Kringles Influences Activation and Conformational Plasticity of Prothrombin, *J. Biol. Chem.*, **291**, 6071-6082.

Raschka, S. (2018) MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack, *Journal of open source software*, **3**, 638.

Remmert, M. *et al.* (2011) HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment, *Nat. Methods*, **9**, 173-175.

Ruiz-Blanco, Y.B. *et al.* (2015) ProtDCal: A program to compute general-purpose-numerical descriptors for sequences and 3D-structures of proteins, *BMC Bioinf.*, **16**, 162.

Saito, T. and Rehmsmeier, M. (2015) The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets, *PLoS ONE*, **10**, e0118432.

Sanner, M.F. *et al.* (1996) Reduced surface: an efficient way to compute molecular surfaces, *Biopolymers*, **38**, 305-320.

Scott, D.L. *et al.* (2006) Protein-lipid interactions: correlation of a predictive algorithm for lipid-binding sites with three-dimensional structural data, *Theor. Biol. Med. Model.*, **3**, 17.

Segers, K. *et al.* (2007) Coagulation factor V and thrombophilia: background and mechanisms, *Thromb. Haemost.*, **98**, 530-542.

Segers, K. *et al.* (2007) Design of protein–membrane interaction inhibitors by virtual ligand screening, proof of concept with the C2 domain of factor V, *Proceedings of the National Academy of Sciences*, **104**, 12697-12702.

Sharikov, Y. *et al.* (2008) MAPAS: a tool for predicting membrane-contacting protein surfaces, *Nat. Methods*, **5**, 119.

Smith, M.R. *et al.* (2014) An instance level analysis of data complexity, *Machine Learning*, **95**, 225-256.

Spiegel, P.C. *et al.* (2004) Disruption of protein-membrane binding and identification of small-molecule inhibitors of coagulation factor VIII, *Chem. Biol.*, **11**, 1413-1422.

Sudhahar, C.G. *et al.* (2008) Cellular membranes and lipid-binding domains as attractive targets for drug development, *Curr. Drug Targets*, **9**, 603-613.

Tomek, I. (1976) An experiment with the edited nearest-neighbor rule, *IEEE Trans. Syst. Man Cybern.*, **SMC-6**, 448-452.

White, S.H. (2003) Translocons, thermodynamics, and the folding of membrane proteins, *FEBS Lett.*, **555**, 116-121.

Whited, A.M. and Johs, A. (2015) The interactions of peripheral membrane proteins with biological membranes, *Chem. Phys. Lipids*, **192**, 51-59.

Wimley, W.C. *et al.* (1996) Solvation energies of amino acid side chains and backbone in a family of host-guest pentapeptides, *Biochemistry*, **35**, 5109-5124.

Wimley, W.C. and White, S.H. (1996) Experimentally determined hydrophobicity scale for proteins at membrane interfaces, *Nat. Struct. Biol.*, **3**, 842-848.

Wolpert, D.H. (1992) Stacked Generalization, *Neural Networks*, **5**, 241-259.

Yang, J. *et al.* (2019) Targeting PI3K in cancer: mechanisms and advances in clinical trials, *Mol. Cancer*, **18**, 26.