How to Best Represent Proteins in Machine Learning-based Prediction of Drug/Compound-Target Interactions

Heval Atas[1,2], Tunca Doğan[1,3,4*]

[1] Biological Data Science Laboratory, Department of Computer Engineering, Hacettepe University, Ankara, Turkey

[2] Department of Health Informatics, Graduate School of Informatics, METU, Ankara, Turkey

[3] Institute of Informatics, Hacettepe University, Ankara, Turkey

[4] Department of Bioinformatics, Graduate School of Health Sciences, Hacettepe University, Ankara, Turkey

* To whom correspondence should be addressed, e-mail: tuncadogan@gmail.com

Abstract

The identification of drug/compound-target interactions (DTIs) constitutes the basis of drug discovery, for which computational predictive approaches have been applied. As a relatively new data-driven paradigm, proteochemometric (PCM) modeling utilizes both protein and compound properties as a pair at the input level and processes them via statistical/machine learning. The representation of input samples (i.e., proteins and their ligands) in the form of quantitative feature vectors is crucial for the extraction of interaction-related properties during the artificial learning and subsequent prediction of DTIs. Lately, the representation learning approach, in which input samples are automatically featurized via training and applying a machine/deep learning model, has been utilized in biomedical sciences. In this study, we performed a comprehensive investigation of different computational approaches/techniques for data preparation and protein featurization, including both conventional approaches and the novel learned embeddings, with the aim of achieving better data representations and more successful learning in PCM-based DTI prediction. For this, we first constructed realistic and challenging benchmark datasets on small, medium, and large scales to be used as reliable gold standards for specific DTI modeling tasks. We developed and applied a network analysis-based splitting strategy to divide datasets into structurally different training and test folds. Using these datasets together with various featurization methods, we trained and tested DTI prediction models and evaluated their performance from different angles. Our main findings can be summarized under 3 items: (i) random splitting of the dataset into train and test folds leads to near-complete data memorization and produce highly over-optimistic results, as a result, it should be avoided; (ii) learned protein sequence embeddings works well in DTI prediction, even though no information related to protein structures, interactions or biochemical properties is utilized during the training of these models; and (iii) PCM models tends to learn from compound features and leave out protein features, mostly due to the natural bias in DTI data. We hope this study will aid researchers in designing robust and high-performing data-driven DTI prediction systems that have real-world translational value in drug discovery.

1. Introduction

Drug discovery is a long-term and costly process that involves the identification of bioactive compounds as drug candidates via screening experiments. Although the advancements in high-throughput screening technology allow the scanning of thousands of compounds simultaneously, it is still not possible to fully analyze a certain portion of the target and compound spaces due to the excessive number of possible protein-compound combinations. This situation led to the emergence of computational approaches, such as the virtual screening (VS), for the *in silico* prediction of unknown drug-target interactions (DTIs) to aid screening experiments [1]. Conventional ligand-based (e.g., QSAR modelling) and structure-based (e.g., molecular docking) VS approaches aim to predict interactions between of a set of compounds and a predefined target protein. Ligand-based approaches mainly achieve this by utilizing molecular property-based compound similarities [2], while structure-based approaches employ 3-D structures of targets and compounds [3].

As a novel data-centric approach in this area, the proteochemometric (PCM) modelling aims to predict bioactivities by incorporating both compound and target features, usually via readily available molecular notation (e.g., SMILES) and amino acid sequence data, without requiring hard to obtain 3-D structures and dynamic information [4]. PCM can predict bioactivity relationships between large sets of compounds and targets under a single system using statistical/data-driven modeling techniques such as machine learning. This characteristic of PCM also allows the identification of off-target affects -a significant limitation of conventional VS approaches [4,5]-, which is especially important for drug repurposing and side-effect identification. To construct a machine learning-based PCM model, first, input compounds and/or target proteins are converted into quantitative feature vectors, so called “representations”, based on their molecular properties (i.e., descriptors). These vectors are then processed together with the magnitude of the bioactivity/interaction between these compounds and targets (i.e., labels), via machine learning algorithms, during the process of supervised model training [6]. For the automated artificial learning of DTIs to be successful, input feature vectors should comprise information about the interaction-related properties of compounds and targets. The better the input data is represented, the better the model can learn and generalize the shared properties among the dataset. Therefore, featurization of the input samples is crucial to construct models with high predictive performance.

There are different types of featurization approaches used for the representation of compounds and proteins. Due to the abundance of ligand-based DTI prediction methods, compound representations are extensively studied in the literature [7–9]. Therefore, this study focuses on protein representation techniques, which is a rapidly developing area lately. Sequence-based protein representations, which utilize amino acid sequences as input (rather than protein structures) are widely preferred in protein associated predictive tasks since complete 3-D structure information is not

available for many proteins and/or proteoforms. Additionally, computational intensity of protein structured-based models is usually high. Considering algorithmic approaches, protein representations can be grouped as conventional/classical descriptors and learned embeddings. Conventional descriptors are constructed (and subsequently employed for predictive tasks) using model-driven approaches that apply predefined rules and/or statistical calculations on various molecular properties including physicochemical [10–12], geometrical [13,14] and topological [12] characteristics of amino acids, sequence composition [11,15], semantic similarities [16], functional characteristics/properties [17–20], and evolutionary relationships [13,21] of proteins. Learned protein embeddings are generated via data-driven approaches that project protein sequences into high-dimensional vector spaces in the form of continuous feature representations using statistical techniques and machine/deep learning algorithms. These protein representation learning (PRL) methods usually borrow their modeling concepts from the field of natural language processing (NLP), where amino acids in a sequence are treated like words in a sentence/document. Due to this reason, many PRL methods are also called “protein language models”. These models usually utilize raw protein sequences within unsupervised learning, without any prior knowledge about their physical, chemical, or biological attributes [22]. Even though they are trained solely on the arrangement of amino acids in the sequence, these models are still found to be successful in automatically extracting physicochemical properties [23] and functional characteristics of proteins [24]. PRL methods have a wide range of applications including prediction of secondary structure [24–27], ligand-target protein interaction [28–30], splice junction prediction [31], family classification [23], protein function [32], remote homology detection [27,33], and protein engineering/design [24,27]. For evaluating the effectiveness of different types of protein representations in different areas of protein informatics, carefully designed benchmark studies are required. In contrary to compound representation studies, only a few studies are available for benchmarking of protein representations on tasks including protein family prediction [11], bioactivity modelling [34,35], and predicting biological properties for protein (re)design [36]. Also, these studies mainly compare conventional alignment-dependent descriptors rather than evaluating more diverse types of featurization. As a result, there is no study that evaluates cutting-edge protein language models, and compares them against well-known conventional descriptors in the context of drug-target interactions and drug discovery/repurposing.

Computational drug discovery studies mostly approached the subject from a chemo-centric point of view, only using compound attributes as input features (e.g., QSAR studies). These studies often overlook protein features and treat target proteins just as labels for input compounds during the modeling. The PCM modelling approach handles this situation by utilizing both protein and compound spaces at the input level. Although PCM modelling has shown promising results when compared to QSAR studies [34,35], it is still far from overcoming the DTI prediction problem. One of the reasons behind this is that the mechanism of learning is not well-understood in PCM, unlike ligand-based modeling in which the model predicts new ligands to the

target protein of interest based on the molecular similarities to its known ligands. In PCM, there are two factors, i.e., compound and protein features, and it is not clear to what degree the similarities between protein samples and between compound samples contribute to the learning process, and whether there is bias in-between. Another problem associated with data-driven DTI prediction in the literature is the reporting of overoptimistic performance results due to; *(i)* low coverage on compound and/or target spaces in terms of molecular and biological properties in training datasets (i.e., limited variance) which prevents the model from gaining the ability to generalize, and *(ii)* poorly planned and applied train/test dataset preparation (e.g., the random split) and model evaluation strategies. Most of the self-proclaimed high performing DTI prediction models in the literature cannot be translated well into real-world cases due to these non-realistic assessments. Recently, there have been efforts in terms of applying different dataset splitting strategies including temporal splitting [37], non-overlapped sampling [9,38], cluster-cross-validation [39], and scaffold-based splitting [40] to build robust models in more challenging scenarios. Temporal splitting strategy only considers a time-dependent data point separation. In non-overlapped sampling strategy, three different splitting settings are applied: warm start (common drugs and targets are present in both the training and test sets), cold start for drugs (drugs in the training set are unseen in the test set while common proteins are shared in these sets), cold start for proteins (proteins in the training set are not involved in the test set, but common drugs are allowed to be present in both sets) [41]. This strategy only differentiates samples in terms of identity, and does not take similarities between compound and/or proteins into account. Although cluster-cross-validation and scaffold-based splitting methods prevent the involvement of similar compounds in train and test sets, they do not take into consideration target similarities. These strategies are not valid for evaluating PCM-based models, in which there are three types of relationships to account for; *(i)* compound-target protein interactions, *(ii)* compound-compound similarities, and *(iii)* protein-protein similarities.

New computational approaches, evaluation strategies and datasets are required in order to address the aforementioned issues regarding data-centric evaluation and prediction of DTIs. In this study, we performed a rigorous benchmarking of various protein featurization techniques over realistic datasets constructed by PCM-suitable dataset preparation strategies, and evaluated the results from different perspectives, with the aim of contributing to the improvement of data-driven bioactivity modelling approaches in the field of drug discovery and repurposing.

One of the goals here is to identify feature types with better representation capabilities to be used for the automated prediction of DTIs. To achieve this, we built prediction models for various sequence-based protein representations using target feature-based and PCM modelling approaches. We employed widely used conventional protein descriptors by selecting those that reflect different aspects of proteins. In addition to the representations constructed by traditional approaches, we

also utilized the state-of-the-art protein representation learning methods (i.e., protein language models) in our benchmarks. The inclusion of learned representations is a remarkable novelty for protein benchmark studies.

Another goal of this study is the preparation of new challenging benchmark datasets with high coverage on both compound and protein spaces, suitable for specific DTI modeling tasks of interest, which can later be utilized in future studies. We carefully prepared small-, medium- and large-scale datasets by applying extensive filtering operations and a network-based splitting strategy to acquire realistic and well-balanced datasets. To our knowledge, this data splitting strategy which considers 3 types of relationships (i.e., drug-target interactions, protein-protein similarities, and compound-compound similarities), is proposed here for the first time. We used these datasets in our protein representation benchmarks.

The study is summarized in a schematic workflow in Figure 1. Firstly, we prepared datasets by applying filters specific to each data scale to avoid overoptimistic predictive performance results. Then, we split these datasets into train and test folds using different strategies for each data scale. For the construction of prediction models, we implemented target feature-based and PCM modelling approaches, and evaluated the performances of DTI prediction models that use different protein representations at the input level, using a variety of scoring metrics, and finally, discussed the results. All details regarding the construction of datasets, representations and DTI prediction models are provided in the Methods section. In the Results and Discussion section, we evaluated the effectiveness of different protein featurization techniques on different datasets, and discussed their strengths and weaknesses especially on the large-scale. As the first near comprehensive benchmark study of protein representation methods in the context of drug discovery and repurposing, we hope this study will aid researchers in choosing suitable representation approaches and techniques according to the specific modeling task at hand. Furthermore, our newly proposed challenging benchmark datasets can be used as reliable, reference/gold-standard datasets in further studies to design robust DTI prediction models with real-world translational value.

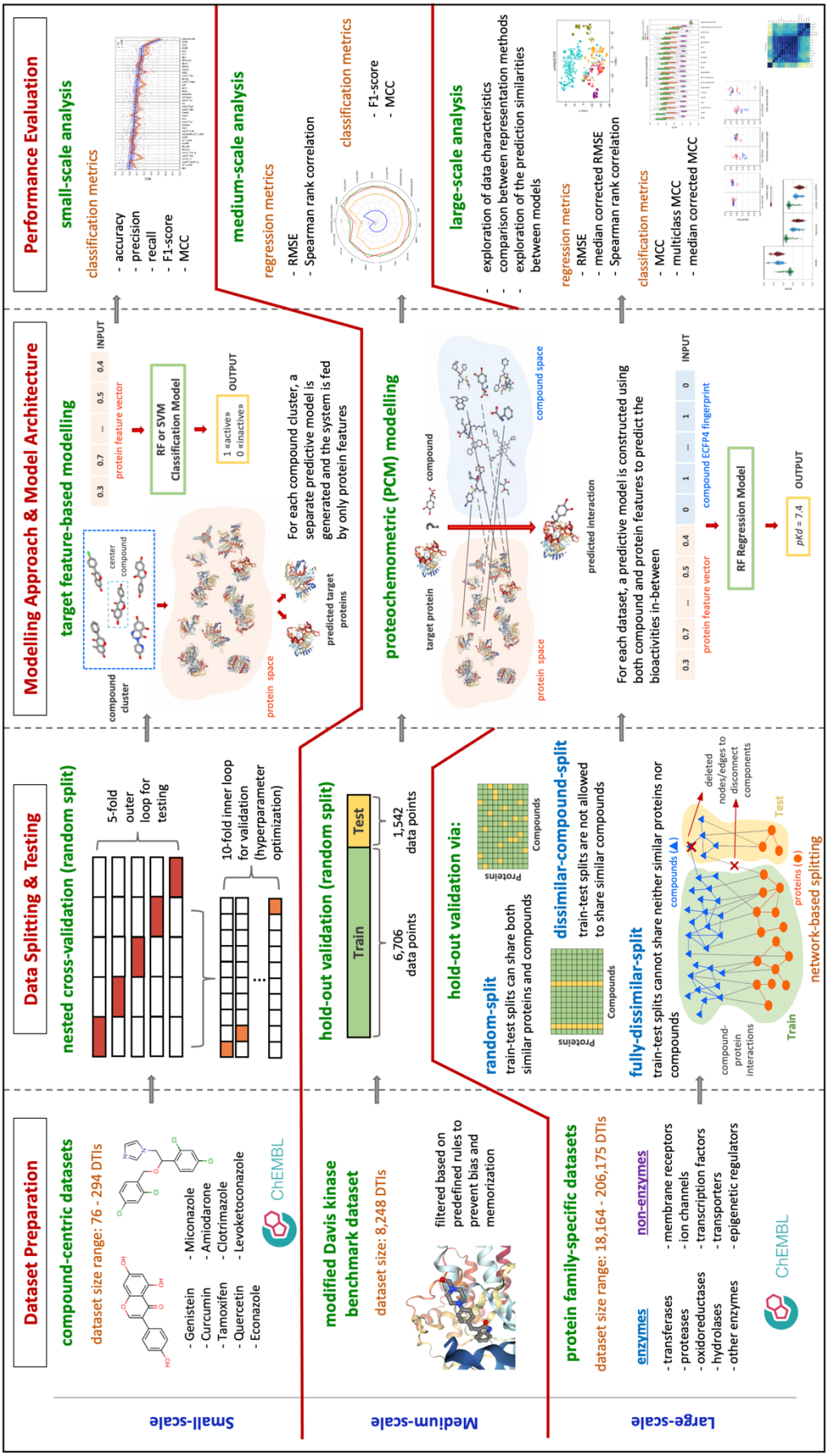

Figure 1. The schematic overview of the study.

2. Methods

In this section, we first explain the construction of benchmark datasets, with emphasis on the train/test data splitting strategies. Next, we explain featurization techniques used for the representation of proteins and compounds. Then, we summarize modelling approaches and algorithms employed for DTI prediction, along with additional explorative analysis such as the t-SNE projections. Finally, we mention performance evaluation metrics and the tools/libraries we used.

2.1. Dataset Construction and Splitting

In machine learning applications, two significant factors that affect the prediction and generalization abilities of models are the dataset content/size and the approach used in splitting data points to train/validation/test folds. We constructed and used three groups of datasets in different scales (i.e., small, medium, and large), each of which have distinctive characteristics, with the aim of observing how performances of different protein featurization methods change under these conditions.

2.1.1. Small-scale: Compound-centric datasets

Here, the aim is to construct datasets of target proteins to be used in DTI prediction models in which the input is protein feature vectors, and the task is to classify them to their ligands. Each dataset is composed of target proteins of a specific drug/compound as reported in the ChEMBL (v23) database [42] in terms of experimentally measured bioactivities. Bioactivity data points with pChEMBL values (i.e., -log(IC50, EC50, Ki, Kd, Potency, …)) >5 (i.e., <10 uM) are placed in the positives (actives) dataset, and instances with pChEMBL <= 5 are placed in the negatives (inactives) dataset. In most cases, sizes of these compound centric training datasets were too small to construct robust prediction models. In order to overcome this problem, we first selected compounds with the highest number of active and inactive bioactivity data points, which we called “center compounds". Afterward, we constructed clusters by calculating pairwise molecular similarities between center compounds and all other compounds in the ChEMBL database using ECFP4 fingerprints and the Tanimoto coefficient. Compounds that are similar to a center compound (Tanimoto similarity >= 0.3 as used in previous studies such as [43]) are placed in the cluster of the corresponding center compound and their bioactivity information (i.e., active and inactive targets) are incorporated into the center compound’s dataset. Therefore, nine independent compound centric datasets were constructed, in which dataset sizes (i.e., the number of targets) range from 76 to 294. Information about these datasets, including statistics such as cluster sizes, active and inactive number of targets, are summarized in electronic supplementary information (ESI) Table S1. In order to balance the number of active and inactive targets in each dataset (initially, the number of inactive targets were considerably low), new proteins which are less than 50% similar to positive targets and less than

80% similar to negative targets already existing in the dataset were selected from ChEMBL and added to the negative target proteins list of the corresponding dataset. Due to the small size of datasets, separating a hold-out test fold was not feasible. Therefore, a nested cross-validation approach (with 10-fold inner loop in validation and 5-fold outer loop in testing) was applied during model evaluation. These datasets are used in small-scale target feature-based analysis described in section 3.2.

2.1.2. Medium-scale: Davis kinase dataset

We employed the Davis kinase dataset [44] for performing benchmark analysis on medium-scale, which is a commonly used benchmark set for DTI prediction. The original train-test instances of the Davis dataset are taken from the study by Ozturk et al. [45], in which a random splitting strategy was applied. This dataset includes ~30,000 DTI data points; however, bioactivity values of ~20,000 of them is 10 uM (i.e., pKd = 5), which causes a bias in predictive models. These are the data points in which a bioactivity was not observed, even when the maximum dose of 10 uM is used during the experiment. In the output, the highest dose (i.e., 10 uM) is written as the respective bioactivity value for these data points. In order to prevent bias, we removed these instances from both train and test datasets. For the train set, three additional filters were applied to avoid data memorization. All bioactivities of a compound or target are discarded if the compound or target:

1. only contains active or inactive data points based on the threshold pKd = 6.2, which is the median bioactivity value of the dataset,
2. has an active-to-inactive ratio > 4 or < 1/4 considering its bioactivity data points,
3. has a bioactivity distribution with standard deviation < 0.3, which means bioactivity values vary within a narrow range.

A successful machine learning model is expected to learn general principles from data rather than memorizing it. The instances fulfilling the conditions above may not contribute to the learning process, as they can be easily predictable regardless of the algorithm or feature set, since they have very similar outcomes. We removed these instances from the dataset, or else, the model would perform well just by memorizing the outcome of these cases. After this filtering operation, the modified Davis (mDavis) dataset is composed of 6,706 train and 1,542 test data points. This dataset is used in medium-scale PCM-based analysis described in section 3.3.

2.1.3. Large-scale: Protein family-specific datasets

With the aim of constructing large-scale gold standard datasets, we applied rigorous filtering operations on the reported bioactivities of different protein families including membrane receptors, ion channels, transporters, transcription factors, epigenetic regulators, and enzymes with five subgroups (i.e., transferases, proteases, hydrolases, oxidoreductases, and other enzymes). Protein family information is

annotated on the basis of ChEMBL [42] protein classification, excluding classes such as secreted proteins, other categories, and unclassified proteins that have inadequate number of data points.

For enzymes, subclasses belonging to the same main category were merged based on their EC number annotations. The merged enzyme classes and their corresponding EC numbers are given in ESI Table S2. Bioactivity data of these families are retrieved from ChEMBL (v24) database. Only the bioactivity data points with target type: “single protein”, standard relation: “=”, pChEMBL value: “not null”, and assay type: “B” (binding assay) are included in the dataset. Dataset statistics of each protein family are provided in ESI Table S3.

For each protein family-based dataset, three types of train-test folds were extracted based on different dataset splitting strategies, in which we considered molecular similarities in-between compounds and proteins. For this, we binarized pairwise similarity measurements as “similar” and “non-similar”. UniRef50 clusters [46] were used for generating protein similarity matrices, where proteins in the same cluster were accepted as similar to each other, which is equivalent to a threshold of 50% sequence similarity. Otherwise, proteins were considered dissimilar to each other. For compounds, Tanimoto coefficient-based pairwise similarities were calculated using compound ECFP4 fingerprints and the RDKit library [47]. Compound pairs with a Tanimoto score >= 0.5 were accepted as similar to each other. Otherwise, compounds were considered dissimilar to each other.

random-split dataset

This dataset is constructed by applying the random splitting strategy, so that similar compounds and proteins are allowed to be involved in both train and test sets. Random splitting is one of the most widely used dataset split strategies in machine learning applications; however, it eases the prediction task due to the presence of similar instances in both train and test sets. Thus, models display overoptimistic performance results. Considering our randomly split family-specific datasets, at least 95% of proteins and 60% of compounds in test sets are found to be similar (i.e., Tanimoto score >= 0.5 for compounds, and being in the same UniRef50 cluster for proteins) to the ones in their respective train sets.

dissimilar-compound-split dataset

This dataset is constructed by applying a splitting strategy that only considers compound similarities for distributing bioactivity data points into train-test splits. Compounds in train and test splits are dissimilar to each other in terms of Tanimoto score with less than 0.5 as explained above. Therefore, similar compounds are not allowed to take part in both train and test splits. This strategy makes the prediction task more difficult compared to the random-split dataset and partly prevents the

model from learning bioactivities fully over compound fingerprints, which contributes to fair benchmarking of protein descriptors.

fully-dissimilar-split dataset

This dataset is constructed using a network-based splitting strategy to separate bioactivity data points (i.e., compound-target pairs) in disconnected components. Later, each component is either used in training or test splits so that, neither compounds nor proteins are similar to each other between training-test splits. This dataset is extremely challenging for DTI prediction methods. However, this approach is crucial to evaluate a DTI prediction model's ability to accurately predict new targets and/or new ligands which are truly novel (i.e., there is no bioactivity information for these compounds and target proteins in the source databases, also, there are no compounds and target proteins significantly similar to these compounds and targets in these bioactivity databases), as this is one of the most important promised advantages of the PCM modeling approach. The steps of the network-based splitting process are provided below:

1) Protein-protein and compound-compound pairwise similarity matrices were obtained independently for each protein family, based on protein family membership information and those proteins' interacting compound information from ChEMBL bioactivity data points. Similarity values were binarized according to the procedure explained above (i.e., 50% sequence/molecular similarity threshold for both protein and compounds).

2) A heterogeneous network was constructed for each protein family by merging similarity matrices and bioactivity data using the NetworkX Python library [48], where nodes represent proteins and compounds, and edges represent protein-protein or compound-compound similarities, and protein-compound bioactivity relationships. Here, it is ensured that any two components that are disconnected from each other in the network do not share any similarity at all, as a result, all bioactivity data points in a particular component can be placed in the training fold, while the ones in another component can be placed in the test fold. As a result, bioactivity data points (i.e., compound-target pairs) in training and test folds are always guaranteed to be dissimilar from each other. In practice, the problem was that nearly all nodes in the network formed a giant connected component, which means that it was not possible to distribute data points to training and test folds over disconnected components.

3) In order to overcome this issue, we preferred to discard some of the nodes (e.g., compounds) and edges (e.g., bioactivity data points) from the dataset to subdivide the giant connected components into smaller pieces. Instead of removing nodes and edges randomly, which may cause the loss of a high number of data points, we employed the Louvain heuristic algorithm [49] to detect communities in the giant component. This algorithm computes the partition of graph nodes by maximizing the network modularity. By discarding edges (or in some cases, discarding nodes if the

edge of interest is between two compounds) between different communities, the number of disconnected components was increased. Finally, bioactivity data points in each component were assigned either to training or test sets in a way that the ratio of the number of training fold data points to the test fold could be held within reasonable values, which still varied between different protein families (i.e., from the minimum of 8.70 to a maximum of 23.97).

Discarded data points of the fully-dissimilar-split dataset were also excluded from training-test folds of random-split and dissimilar-compound-split datasets for keeping instances of three sets exactly the same to yield fully comparable results between different split-based datasets. The sizes of these datasets after discarded data points range from 18,164 to 206,175 depending on the protein family. Detailed statistics are provided in ESI Table S3 and S4. These datasets are used in large-scale PCM-based analysis described in section 3.4.

2.2. Representation of Proteins and Compounds

We converted proteins and compounds into fixed-length numerical feature vectors to be used in our DTI prediction models as input samples. The following section describes different featurization approaches used in this study.

2.2.1. Protein representations

On the basis of algorithmic approaches used, we divided this subsection into two categories as conventional protein descriptors and learned protein embeddings. To construct both types of representations, protein amino acid sequences were used as input. These representation methods are explained below in terms of their molecular and technical aspects.

Conventional descriptors

This category comprises methods that employ model-driven approaches. This is achieved by transforming various molecular properties of proteins, such as sequence composition, evolutionary relationships, functional characteristics, or physicochemical properties of amino acids, into fixed-length numerical feature vectors with the implementation of predefined rules or statistical calculations. Hence, they convert protein sequences into a quantitative and machine-readable format that stores the relevant molecular information.

Eleven conventional protein descriptor sets (including the randomly generated one) used in this study are briefly explained below. Names, descriptions, and feature vector dimensions of these descriptors are given in Table 1.

- *apaac (amphiphilic pseudo amino acid composition)* represents the amino acid composition of protein sequences without losing the residue order effect by using

sequence-order factors. These factors are computed from correlation functions of hydrophobicity and hydrophilicity indices of amino acids. Therefore, apaac keeps the distribution of amphiphilic amino acids along the protein chain. It was proposed by Chou in 2005 and used for the prediction of enzyme subfamily classes [10].

- *ctdd (distribution)* provides distribution patterns of amino acids in terms of the class they belong to considering a particular property. It utilizes 7 types of physicochemical properties including hydrophobicity with 7 different versions, normalized Van der Waals Volume, polarity, polarizability, charge, secondary structures, and solvent accessibility. Each property is divided into 3 classes and 20 amino acids are distributed into these classes based on their values for corresponding property (i.e., helix -EALMQKRH-, strand -VIYCWFT-, and coil -GNPSD- classes for secondary structure property). The distribution patterns are determined according to five different positions (residues) for the corresponding class, which are the first residue, and the residues exactly at the 25%, 50%, 75%, and 100% of the sequence. These positions are divided by the length of the whole protein sequence for the calculation of fractions of each class. This descriptor set was first proposed by Dubchak for protein fold recognition task [50].
- *ctriad (conjoint triad)* is based on the frequency of triple amino acid combinations in a protein sequence, where amino acids are first converted into a 7-letter reduced alphabet. These seven groups include {A,G,V}, {I,L,F,P}, {Y,M,T,S}, {H,N,Q,W}, {R,K}, {D,E}, and {C}. Amino acid groups are determined according to dipoles and volumes of the amino acid side chains. Ctriad was used for the first time to predict protein-protein interactions by Shen et al. [51].
- *dde (dipeptide deviation from expected mean)* is a kind of sequence composition descriptor set that relies on the deviation of dipeptide compositions from the expected means. Three parameters, i.e., dipeptide composition (*Dc*), theoretical mean (*Tm*), and theoretical variance (*Tv*), are computed for the construction of the dde descriptor set. Saravanan and Gautham proposed it in 2015 for the use of B-cell epitope prediction [15].
- *geary* utilizes the distribution of structural and physicochemical properties of amino acids along the sequence. It was first developed by Geary in 1954 [52] as a measure of spatial autocorrelation that uses the square-difference of property values. Li et al. served it as a protein descriptor set via the PROFEAT web server [53]. Also, Ong et. al. implemented it for the prediction of protein functional families [11].
- *k-sep_pssm (k-separated-bigrams-pssm)* is a column transformation-based descriptor set that computes the bigram transition probabilities between residues in terms of their positional distances from each other, which corresponds to the “k” value. The transition probabilities are calculated from transformations on position-specific scoring matrix (pssm) profiles of proteins. Pssm profiles represent evolutionary conservation of amino acids in a protein sequence, which are derived

from multiple sequence alignments of a set of protein sequences. This descriptor set was first proposed in the study of Saini et al. for improving protein fold recognition [21]. Wang et al. developed the POSSUM tool to calculate a set of PSSM-based descriptors including k-sep_pssm, and they utilized these descriptors for the prediction of bacterial secretion effector proteins [54].

- *pfam* represents domain profiles of proteins, according to protein domain annotations in the Pfam database [55], in the form of binary feature vectors. For each protein, it encodes the presence (1) and absence (0) of a unique list of domains presented in proteins in the corresponding dataset. This descriptor set was employed in the studies of Yamanishi et al. [18] and Liu et al. [56] with the purpose of predicting drug-target interactions.
- *qso (quasi-sequence order)* reflects the indirect effect of the protein sequence order by calculating coupling factors in terms of distances between contiguous residues in the sequence. The distances are determined using different amino acid distance matrices such as Schneider–Wrede distance matrix [57], which is derived from hydrophobicity, hydrophilicity, polarity, and side-chain volume properties of amino acids. It was first developed by Chou for the prediction of protein subcellular locations [58].
- *spmap (subsequence profile map)* is a feature space mapping method that represents sequence composition by calculating the distribution of fixed-length protein subsequence (length of 5 residues in the default version) clusters in a protein sequence. Subsequence clusters are generated using BLOSUM62 matrix-based similarities of all possible subsequences in the given protein set, extracted by the sliding windows approach. It was proposed by Sarac et al. for functional classification of proteins [59]. Later, spmap was used in GO term [60] and EC number [61] prediction methods. In this study, spmap-based feature vectors were generated using clusters of 5-residue subsequences of ChEMBL target proteins.
- *taap (total amino acid properties)* represents the total sum of corresponding residue values in a protein sequence for the selected properties from the AAIndex database [62]. It was first employed by Gromiha and Suwa for better discrimination of outer membrane proteins [63]. The properties used in our study are normalized average hydrophobicity scale, average flexibility indices, polarizability parameter, free energy of solution in water, residue accessible surface area in tripeptide, residue volume, steric parameter, relative mutability, hydrophilicity value and the side chain volume.
- *random200* is a descriptor set that constructs a feature vector (with the size of 200 x 1) for each different protein sequence, involving randomly generated continuous values ranging from 0 to 1. This representation was generated as a dummy protein representation for performance comparison against real protein representations (with the aim of checking whether bioactivities are learned completely using compound descriptors, in which the proteins descriptors would be useless).

iFeature stand-alone tool [64] was employed for the calculation of apaac, ctdd, ctriad, dde, geary and qso feature vectors. Protein domain annotations were retrieved from the Pfam database [55] for the construction of pfam feature vectors. Spmap feature vectors were calculated using the in-house algorithm explained above [59]. For k-sep_pssm and taap, POSSUM [54] and PROFEAT [65] web servers were used, respectively.

Learned embeddings

This category comprises protein representations that utilize data-driven approaches for the extraction of molecular information from protein sequences. Instead of using predefined rules and calculations as in model-driven conventional descriptors, learned representations are constructed via an artificial learning process, in which a model is trained on specific unsupervised/self-supervised tasks such as the prediction of the next amino acid in the sequence. For generating such protein representation models, deep neural network-based architectures and modeling approaches used in natural language processing (NLP) are preferred. During the training process, the model takes protein sequences as input, projects them into a high-dimensional vector space, and outputs in the form of fixed-length numerical feature vectors called “embeddings”. These numerical feature vectors can later be used for representing proteins in other predictive tasks (mostly supervised) such as DTI prediction.

Four protein representation learning methods used in this study are briefly explained below. Names, descriptions, and feature vector dimensions of these embeddings are given in Table 1.

- *unirep* is one of the best-known learned protein representations, which was developed in 2019 by Alley et al. using a variation of recurrent neural networks (RNN) called the multiplicative long-/short-term-memory (mLSTM) architecture [24]. Alley et al. trained the model on approximately 24 million protein amino acid sequences in the UniRef50 clusters of UniProt, with the objective of predicting the next amino acid in these sequences. They evaluated the representation capability of unirep on various tasks including the prediction of protein stability, semantic similarity, secondary structure, evolutionary and functional information. In our study, we constructed 1900- and 5700-dimensional unirep protein embeddings for each sequence. The model used for the construction of these embeddings is available at https://github.com/churchlab/UniRep.
- *transformer* is a powerful deep architecture that utilizes the attention mechanism in a way to allow the extraction of context without depending on the sequential order information in the input samples [66], and it is currently widely used in representation learning and generative modelling of textual data with large-scale parallelization during training. As part of the “Tasks Assessing Protein Embeddings (TAPE)” study, Rao et al. developed a transformer-based protein

representation learning model using the Bidirectional Encoder Representations from Transformers (BERT) algorithm [27]. This model was trained on approximately 32 million protein sequence fragments taken from the Pfam domain annotation database [55] for masked-token prediction. It was also tested on tasks such as secondary structure prediction, contact prediction, remote homology detection, fluorescence landscape prediction, and stability landscape prediction. For each sequence, the model returns two different versions of representation vectors: (i) average, and (ii) pooled, in 768-dimensions. The model is accessible at https://github.com/songlab-cal/tape.

- *protvec* was developed by Asgari and Mofrad [23] as one of the first models used in the construction of learned protein embeddings. It was trained on 546,790 sequences in the UniProtKB/Swiss-Prot database using the skip-gram modelling approach, in which, given a target residue, the model predicts the surrounding amino acids. In protvec, protein sequences were embedded into 100-D vectors of 3-gram subsequences as biological words (i.e., 3-length aa sequences). For characterizing biophysical and biochemical properties of sequences, these 3-grams were analyzed qualitatively and quantitatively in terms of mass, volume, van der Waals volume, polarity, hydrophobicity, and charge. Protein feature extraction capability of protvec was also evaluated in terms of protein family classification and disordered sequence characterization tasks by representing each protein sequence as the summation of 100-D vectors of its 3-grams. 100-D vector representation of 3-grams can be retrieved from http://dx.doi.org/10.7910/DVN/JMFHTN.
- *seqvec* utilizes bi-directional language model architecture of the method: “Embeddings from Language Models (ELMo)” for extracting features relevant to per-residue and per-protein prediction tasks. Heinzinger et al. developed the seqvec model by training on approximately 33 million UniRef50 sequences with the goal of predicting the next amino acid in the sequence [25]. The authors evaluated the success of seqvec on the prediction of secondary structures and regions with intrinsic disorder at the residue level, and subcellular localization prediction at the protein level. 1024-dimensional seqvec protein embeddings can be computed as described in https://github.com/rostlab/SeqVec.

2.2.2. Compound representations

For representing compounds, we employed the (circular) fingerprinting approach, and used Extended-Connectivity Fingerprints (ECFPs) as feature vectors. ECFPs are circular topological fingerprints that are widely used for molecular characterization, similarity searching, and structure-activity relationship modeling. ECFPs represent the presence of particular substructures by considering circular atom neighborhoods within a diameter range [67]. We constructed 1024-bit ECFP4 fingerprints (corresponding to a radius of 2) using RDKit [47], in which compound SMILES notations were given as input to the tool. As output, a fixed-length binary

fingerprint vector was generated for each compound by applying a hash function on its substructures. For the medium- and large-scale PCM models, we also generated 1024-bit random binary compound fingerprints to be used in dummy baseline models for evaluating the effect of compound data on DTI prediction performances. To be able to simulate ECFP4 fingerprints more realistically, we adjusted the frequency of 1 and 0 values to 0.1 and 0.9, respectively (similar to real feature vectors).

2.3. Modelling Approaches

In order to evaluate different protein featurization methods on DTI prediction, we utilized 2 different modelling approaches: (i) target feature-based modelling, and (ii) proteochemometric (PCM) modelling. Below, we summarized each approach together with the implementation details.

2.3.1. Target feature-based modelling

In target feature-based DTI prediction modelling (also known as "*in silico* target-fishing" or "reverse-screening based modelling"), we trained an independent DTI prediction model for each selected drug/compound cluster (Methods subsection 2.1.1). Representation/feature vectors of proteins that are reported to be interacting with the drug/compound of interest (in the ChEMBL database) are given as input to the model for training and test. Here, the model predicts whether a query protein could be the target of the corresponding compound of that model, via binary classification. For this, we binarized the source bioactivity data points (i.e., compound-target pairs) according to the magnitude of the interaction in-between, as pChEMBL>5 pairs are accepted as active (i.e., interacting), and pChEMBL<=5 pairs are accepted as inactive (i.e., non-interacting). Hence, the system input is solely composed of protein features, where compounds are just used as labels.

We generated separate models for each protein descriptor set using support vector machine (SVM) and random forests (RF) classifiers, as these are widely used and well-performing machine learning algorithms. The models are implemented with scikit-learn python library [68]. For SVM models, "rbf" kernel was applied with optimized C and gamma parameters within ranges of [1,10,100] and [0.001,0.01,0.1,1], respectively. RF models were constructed by adjusting the parameters as; number of trees: 200, and the maximum feature number: the square root of the total number of features. We employed "small-scale compound-centric" datasets explained in the Methods subsection 2.1.1 for the model training and performance evaluation. In the end, we trained and tested 1935 RF and 1935 SVM models (i.e., 43 protein descriptor sets -including random200- for 9 different drug/compound clusters, 5-fold outer loop in nested cross validation: 43*9*5).

2.3.2. Proteochemometric (PCM) modelling

We constructed PCM models for both medium-scale and large-scale datasets (please see Methods 2.1) using the RF regression algorithm, since we observed that RF models performed better than SVM models in the previous analysis of target feature-based modelling. For parameters, we adjusted the number of trees to 100 and maximum ratio of features to 0.33, corresponding one third of the total number of features.

Here, the task is predicting the actual binding affinity values of the input compound-target pairs in terms of pKd/pChEMBL values. In order to reduce the number of conventional protein descriptors to a manageable level, we selected 10 out of the total 42 descriptor sets by removing the ones that have similar featurization approaches and similar performance scores in the previous analysis. Hence, we generated PCM models for 10 conventional protein descriptors, including apaac, ctdd, ctriad, dde, geary, k-sep_pssm, pfam, qso, spmap and taap, and 6 learned representations including protvec, seqvec, transformer-avg, transformer-pool, unirep1900, and unirep5700. These protein representations were concatenated with ECFP4 (1024 bits) fingerprint-based representations of compounds to construct the finalized input feature vectors of models. Furthermore, we generated two baseline models by concatenating random200 protein feature vectors with ECFP4 and random compound fingerprints, which are named “random200” and “random200_random-ecfp4” models, respectively. We also constructed two additional models on the large-scale dataset by training the system solely on the ECFP4 and random compound feature vectors, instead of concatenating with protein feature vectors, which we called “only-ecfp4” and “only-random-ecfp4” models. Thus, we trained and tested 18 PCM models on the medium-scale using the mDavis kinase dataset (i.e., models built on 10 conventional descriptors and 6 learned representations as well as 2 baseline models). For PCM-based models on large-scale, we trained and tested models on 10 protein family-specific datasets, each having 3 versions of train-test split folds with varying difficulty levels, which are explained in Methods subsection 2.1.3 in detail. Therefore, we constructed 600 PCM models on large-scale (i.e., 3 splits * 10 families * (10 conventional descriptors + 6 learned embeddings + 4 baseline models)).

2.4. t-SNE Projection of Protein Representations on Large-Scale Datasets

t-distributed stochastic neighbor embedding (t-SNE) is a non-linear dimensionality reduction technique that is especially used for the visualization of high dimensional datasets [69]. For exploratory analysis of protein family-specific large-scale datasets, we applied t-SNE on protein feature vectors of each protein representation method and colored nodes in different embeddings according to protein (sub)families and train-test fold data points. We implemented the algorithm using the scikit-learn [68] manifold module with default parameters (i.e., 2-D embedding space, perplexity=30,

and Euclidean distance metric), following an investigation of perplexity values in the range of 40 to 100.

Table 1. Properties of the selected protein descriptor sets and representations used in our benchmarks.

Name	Approach	Description	Dimension
apaac [10]	Model-driven (physico-chemistry)	amino acid composition regarding the sequence order correlated factors computed from hydrophobicity and hydrophilicity indices of a.a*	80
ctdd [50]	Model-driven (physico-chemistry)	chain length-based distribution of a.a for selected physicochemical properties	195
ctriad [51]	Model-driven (physico-chemistry)	triad frequency of residues classified on dipoles and volumes of aa side chains	343
dde [15]	Model-driven (sequence comp.**)	dipeptide composition deviation	400
geary [53]	Model-driven (physico-chemistry)	autocorrelation regarding the distribution of physicochemical properties of a.a	240
k-sep_pssm [21]	Model-driven (sequence homology)	column transformation-based position specific scoring matrix (pssm) profiles	400
pfam [18]	Model-driven (functional properties)	domain profiles	38-294 ***
qso [58]	Model-driven (physico-chemistry)	sequence order effect based on physicochemical distances between coupled residues	100
spmap [59]	Model-driven (sequence comp.*)	subsequence-based feature map	544
taap [63]	Model-driven (physico-chemistry)	summation of corresponding residue values for selected physicochemical properties	10
random 200	-	randomly generated continuous numbers between 0 and 1 with uniform distribution	200
protvec [23]	Data-driven (learned embedding)	sequence embedding utilizing skip-gram modelling approach	100
seqvec [25]	Data-driven (learned embedding)	sequence embedding based on bi-directional language model architecture “ELMo”	1024
transformer [27]	Data-driven (learned embedding)	transformer-architecture based embedding method that utilizes attention mechanism	768
unirep [24]	Data-driven (learned embedding)	sequence embedding based on mLSTM architecture as a variation of recurrent neural networks	1900 & 5700

* amino acids, ** compositon, *** size varies depending on the dataset, since pfam vectors only include the domains presented in the given protein dataset.

2.5. Model Performance Evaluation

The performance of target feature-based models was evaluated via accuracy, precision, recall, F1-score, and Matthews Correlation Coefficient (MCC) classification metrics with nested cross-validation (set as 10-fold inner loop for hyperparameter optimization and 5-fold outer loop for testing) on the small-scale dataset. F1-score is the harmonic mean of precision and recall; thus, it takes both false positive and false negative predictions into account. MCC incorporates all true and false predictions into the equation, and it is preferred over both accuracy and F1-score due to its robustness and reliability, especially in the cases of dataset imbalance [70].

For PCM-based regression models, performances were evaluated using Root Mean Square Error (RMSE) and Spearman rank correlation (r_s) metrics over the hold-out test set of mDavis dataset on medium-scale; and random-split, dissimilar-compound-split, and fully-dissimilar-split based test sets of each protein family-specific dataset on large-scale. RMSE computes the deviation of predictions from the actual values, and lower RMSE scores indicate better model performance. Spearman correlation evaluates the relationship between the ranks of the predicted and actual values. One of the problems related to regression-based prediction models is that, in some cases the distribution of predicted values has a shifted mean (i.e., the distribution and rank of predictions is in correlation with the true label values; however, the mean prediction value is either higher or lower than the true mean). Most of the performance evaluation metrics suffer from this problem and report underestimated scores (except rank-based metrics such as the Spearman's rank correlation). In order to handle this problem, we calculated an additional version of RMSE scores for family-specific datasets on the large-scale. For this, we modified test prediction results by median correction, so that the median value of predictions becomes equal to the median of the distribution of true values in the training dataset. We then calculated the so called “median corrected RMSE” scores based on the median corrected prediction results.

In order to evaluate the results of PCM-based regression models on the basis of classification, F1-score and MCC scores were calculated. To achieve this, samples were classified as active (1) or inactive (0) at the cut-off value of pKd = 7 (i.e., 100 nM in terms of Kd) for models trained on the medium-scale mDavis kinase benchmark dataset. For large-scale protein family-specific PCM-based models, samples were converted into binary class and multiclass formats to calculate MCC scores. For binary classification, median pChEMBL values of the data points in the training datasets were used as threshold values to separate actives and inactives from each other (i.e., compound-target pairs with bioactivity values higher than the median value of the dataset are accepted as actives, and the ones equal to or lower than the median are accepted as inactives). For the calculation of multi-class MCC scores, samples were placed into six different classes based on their pChEMBL value ranges of <5.0, 5.0 - 5.5, 5.5 - 6.0, 6.0 - 6.5, 6.5 - 7.0, and >=7.0. We also calculated the median corrected version of MCC scores by binarizing test prediction

results as explained above for “median corrected RMSE”, and called this metric “median corrected MCC”. The reason behind using such a variety of performance metrics was to evaluate models from as many different aspects as possible.

The equations of these metrics are given below:

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

$$Recall/Sensitivity = \frac{TP}{FN + TP} \quad (2)$$

$$Precision = \frac{TP}{FP + TP} \quad (3)$$

$$F1 - score = 2\ x\ \frac{precision * recall}{precision + recall} \quad (4)$$

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

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (6)$$

$$Spearman\ rank\ correlation\ (r_s) = 1 - \frac{6\sum_{i=1}^{n} {D_i}^2}{n(n^2-1)} \quad (7)$$

where D_i = R(y_i) - R($\hat{y}_i$); D_i denotes the difference between ranks of true (y_i) and predicted ($\hat{y}_i$) values of samples with size n. TP, TN, FP, and FN are the numbers of true positive, true negative, false positive, and false negative predictions, respectively.

In this study, we used Python (v3) programming language, scikit-learn library [68] for the t-SNE projection and machine learning applications, NetworkX package [48] for splitting protein family-specific datasets, RDKit toolkit [47] for compound featurization and clustering, POSSUM [54] and PROFEAT [65] web tools as well as iFeature stand-alone tool [64] for the protein featurization, and seaborn [71] and matplotlib [72] libraries for the heatmap analysis and data visualization.

3. Results and Discussion

In this section, we evaluate and discuss various protein representation approaches via data-driven DTI prediction on carefully constructed datasets. For this, we first carried out a data exploration analysis (section 3.1), in which we visualized proteins in our datasets using various protein representations and observed protein family-based clusters (3.1.1), and we analyzed characteristics of our datasets under three data splitting strategies: *(i)* random-split, *(ii)* dissimilar-compound-split, and *(iii)* fully-dissimilar-split (3.1.2). Next, we trained DTI prediction models under different settings and measured the performance of different protein representations on small-scale (3.2), medium-scale (3.3), and large-scale (3.4) datasets. At each section, we discussed model performances from various aspects to address shortcomings in bioactivity modelling studies and help for the development of better computational predictive models in the field of drug discovery and repurposing.

3.1. The Exploration of Data Characteristics

In this subsection, we first visualized members of protein family-specific datasets on 2-D via t-SNE projection. Then, we analyzed split-based characteristics of these datasets by obtaining pairwise similarity distributions of proteins and compounds, bioactivity distributions of train-test folds, and t-SNE visualizations of these folds.

3.1.1. t-SNE projection of protein families

For each protein representation, two independent t-SNE projections (belonging to the enzyme and non-enzyme groups) were carried out (Figure 2a and 2b). Projections for 8 representation methods are shown in Figure 2, and the remaining ones are available in ESI Figure S1. As displayed in these t-SNE plots, generally, protein families are clustered for both enzyme and non-enzyme groups, with less apparent clusters in enzymes, probably due to the sharing of enzyme-specific properties between proteins. Also, members of the other-enzymes class are scattered among other clusters as its members do not have distinctive characteristics. Although the majority of protein representations are successful at separating at least some of the families, projections of learned embeddings have clearer clusters in general, which indicates their ability for extracting family-specific features. In terms of the types of conventional descriptors, homology (i.e., k-sep_pssm) and domain profiles (i.e., pfam) are observed to have more distinctive abilities for the classification of protein families, compared to physicochemistry (e.g., apaac, ctdd, ctriad, geary, qso) and sequence composition (i.e., dde). The t-SNE projection of spmap, being a sequence composition-based descriptor set representing protein subsequence clusters, is similar to the projection of random200 descriptor set. This result indicates that 5-residue subsequences of proteins cannot capture family-specific patterns. Highly distinct from other representations, taap has a projection in the form of an S-shaped curve. Feature vectors of proteins with

similar residue content and sequence length are similar for the taap descriptor set, independent from the actual order of amino acids on the sequence, since taap uses the total sum of the amino acid-based property values to represent a protein. Since t-SNE is a locally sensitive algorithm, proteins form a continuous curve similar to time-series data when represented by taap.

3.1.2. Split-based characteristics of protein family-specific datasets

Pairwise similarity distributions

To explore protein and compound diversity in our datasets, we evaluated protein-protein and compound-compound pairwise similarities of the members of the transferase family, in terms of train vs. train, test vs. test, and train vs. test dataset samples comparisons for each split method (i.e., random-split, dissimilar-compound-split, fully-dissimilar-split). For this, we aligned protein sequences using EMBOSS Needle global sequence alignment tool [73] and plotted histograms based on identity values of unique protein pair combinations in the corresponding datasets. We extracted pairwise compound similarities by calculating Tanimoto coefficient between fingerprint representations using the *simsearch* function of the Chemfp python package [74]. Since it was highly infeasible to calculate pairwise similarities for billions of compound pairs, we randomly sampled 10% of all compounds in the dataset and set the minimum similarity detection threshold as 0.1. We only considered unique compound pairs.

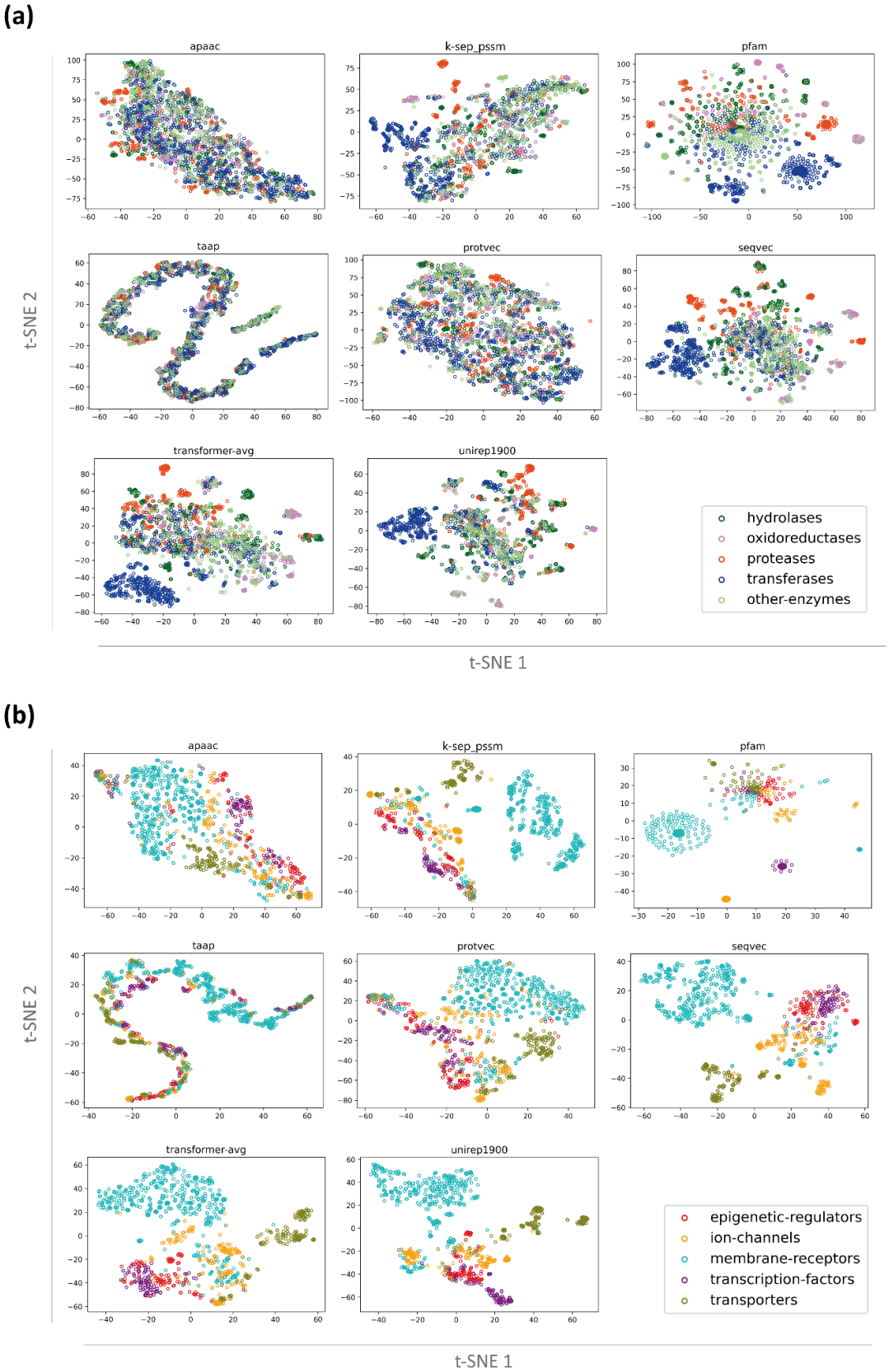

Figure 2. t-SNE based visualization of conventional (apaac, k-sep_pssm, pfam, taap) and learned (protvec, seqvec, transformer-avg, unirep1900) protein representations on **(a)** enzymes including hydrolases, oxidoreductases, proteases, transferases, and other-enzymes, and **(b)** other protein families (non-enzymes) including epigenetic regulators, ion channels, membrane receptors, transcription factors, and transporters, in different colors.

Figure 3 displays similarity distributions of pairs of proteins and compounds involved in the transferases dataset, in which the values may be greater than one since the plot is normalized to equalize the total area to one (i.e., the density plot). Having a similarity value in the range of 0 - 0.5 for the majority of protein and compound pairs in all plots demonstrates the high diversity of samples which is a desirable characteristic for computational bioactivity modelling. As displayed in Figure 3, similarity distributions only slightly change between different split methods, considering train vs. train and test vs. test sample similarities, whereas there are significant differences between the samples of train vs. test, for both compounds and proteins. The absence of similarity values greater than 0.5 for compound train vs. test pairs in the dissimilar-compound-split dataset, and both protein and compound train vs. test pairs in the fully-dissimilar-split dataset validates the similarity-centric characteristics of our datasets. Exceptional pairs of proteins with high similarity values in the fully-dissimilar-split dataset stem from the discrepancies between UniRef50 clusters and our pairwise alignment results, and their number is found to be insignificant (please note that the frequencies are given on logarithmic scale in Figure 3). These results validate the capability of our methodology in terms of producing train-test datasets on which the measured bioactivity prediction model performances have the potential to reflect the real-world performance while discovering novel drug candidates and/or new targets.

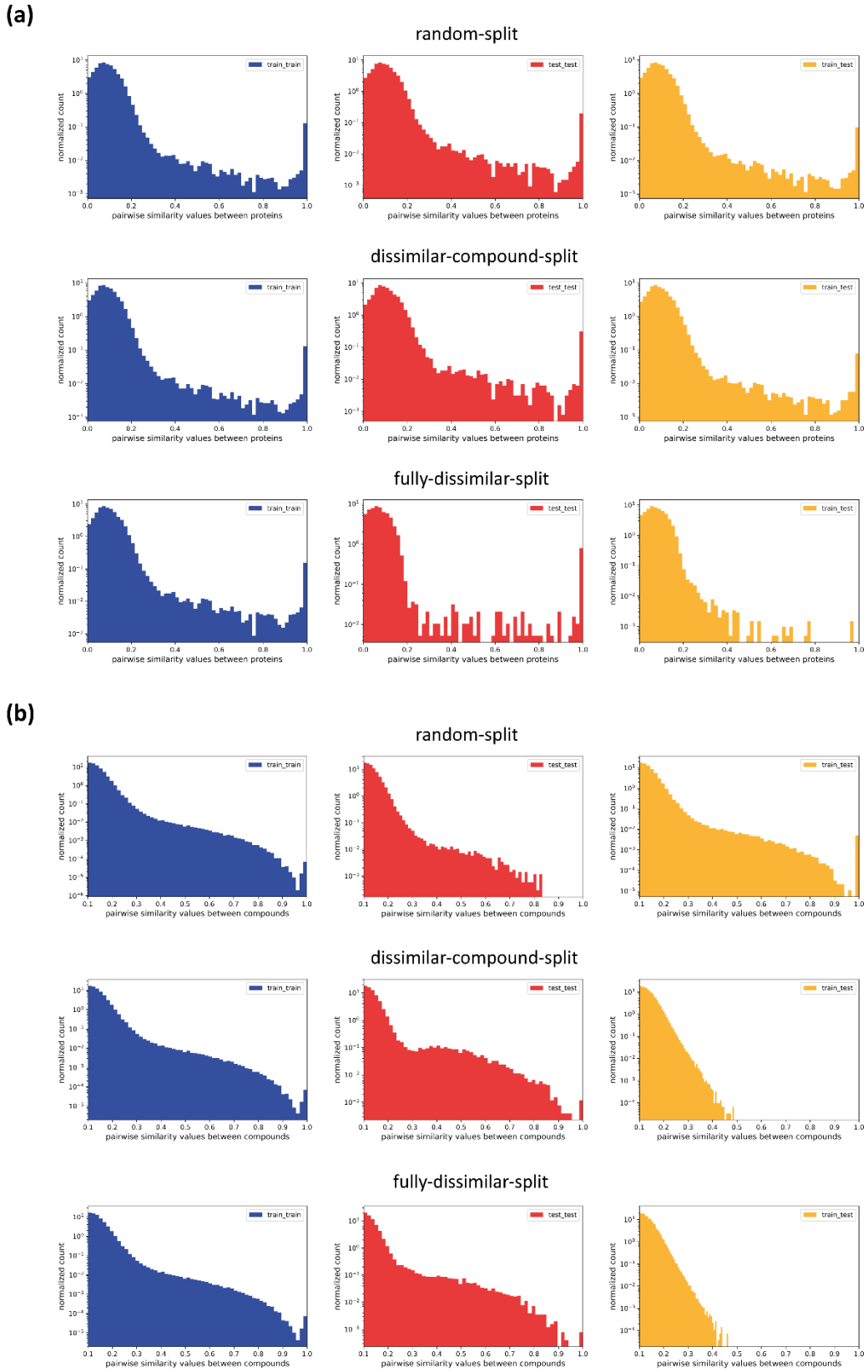

Figure 3. Pairwise similarity distributions of **(a)** proteins and **(b)** compounds on logarithmic scale for train vs. train, test vs. test, and train vs. test samples in random-split, dissimilar-compound-split, and fully-dissimilar-split of transferases dataset.

Bioactivity distributions

We also plotted bioactivity distributions of family-specific datasets based on train-test samples of each split. Figure 4 displays bioactivity distributions of transferases, ion channels, and membrane receptors (plots for the remaining families are available in ESI Figure S2). Median bioactivities in terms of pChEMBL values vary in between 5.7-7.1 for different protein families. When comparing bioactivities of train and test sets of each family, it is observed that distributions have similar shapes, regardless of the dataset split type. In addition, they generally have very similar mean and median values, although the difference is slightly higher in the fully-dissimilar-split datasets of some families. Having bioactivity distributions that are consistent with each other in training and test sets, independent from the splitting strategy, implies good coverage of bioactivity data and supports the suitability of our large-scale datasets for bioactivity modelling. These results also indicate that a stratified-split strategy is not required for our datasets.

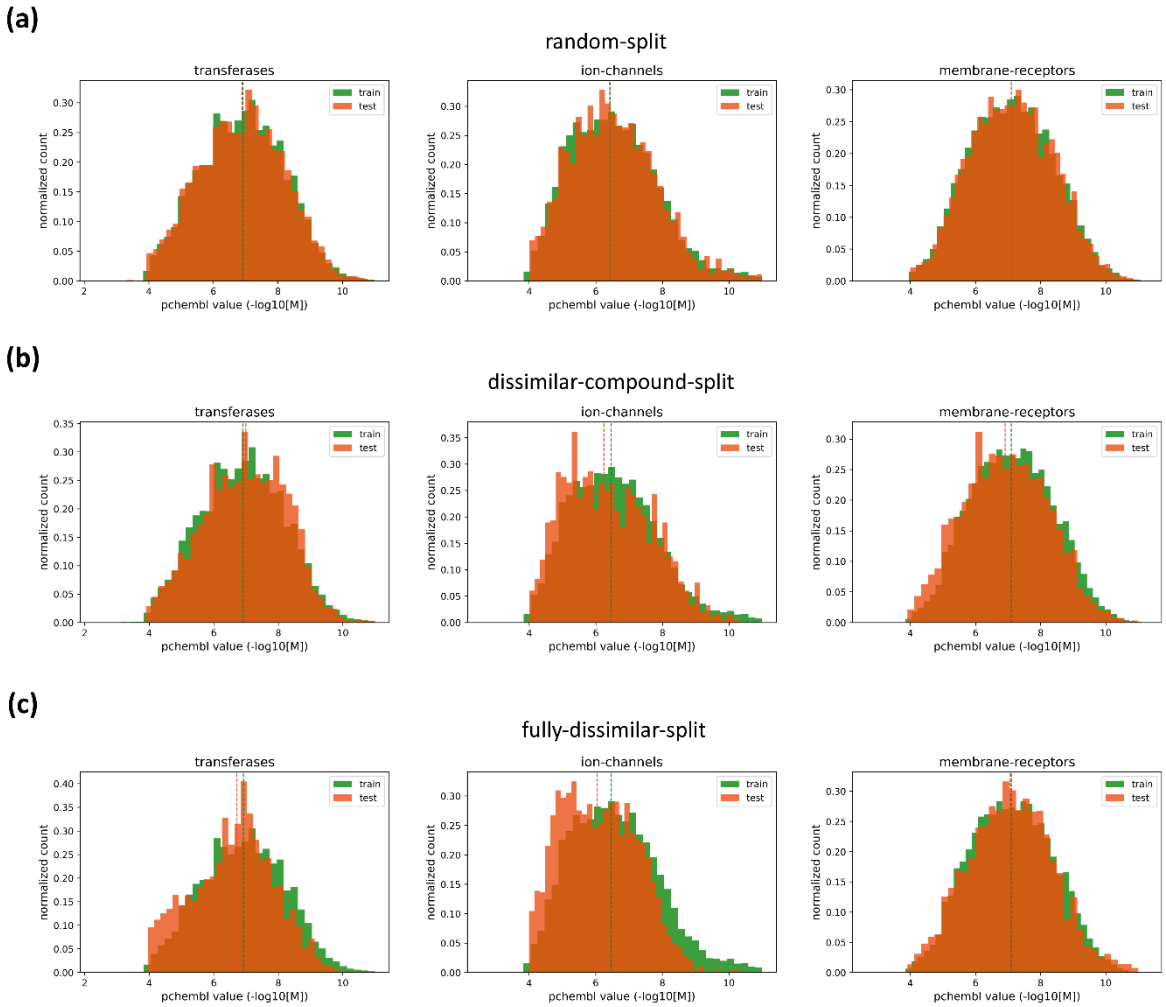

Figure 4. Bioactivity distributions of transferases, ion channels, and membrane receptors based on train (green bars) and test (orange bars) dataset samples of; **(a)** random-split, **(b)** dissimilar-compound-split, **(c)** fully-dissimilar-split datasets, along with their median values.

t-SNE projection of train-test datasets for three splits

In this analysis, we aimed to visualize the distribution of bioactivity data points (i.e., compound-protein pairs) on 2-D via t-SNE projections to observe how train and test fold samples are separated from each other under different split settings. For each protein family-based dataset, 1,500 data points were randomly selected (from both train and test folds), since the number of training samples dominates the number of test samples in the original datasets. Selected bioactivity data points were represented by the concatenation of the respective protein and compound feature vectors, and used as the input of the t-SNE algorithm.

In Figure 5, t-SNE plots of transferases and ion channels are given (as these are widely used protein families in DTI studies) for k-sep_pssm and unirep1900 representations. Panel a, b, and c correspond to random-split, dissimilar-compound-split, and fully-dissimilar-split dataset, respectively. For the random-split dataset, 2-D embeddings of the train and test samples largely overlap, since they share similar proteins and compounds. These overlaps significantly decrease in dissimilar-compound-split dataset and almost disappear in the fully-dissimilar-split dataset, as expected. The t-SNE plots of other protein families and representations portrayed similar results, as well. This analysis can be considered as a visual validation of the implemented splitting strategies, and it provides clues about the difficulty levels of our prediction tasks.

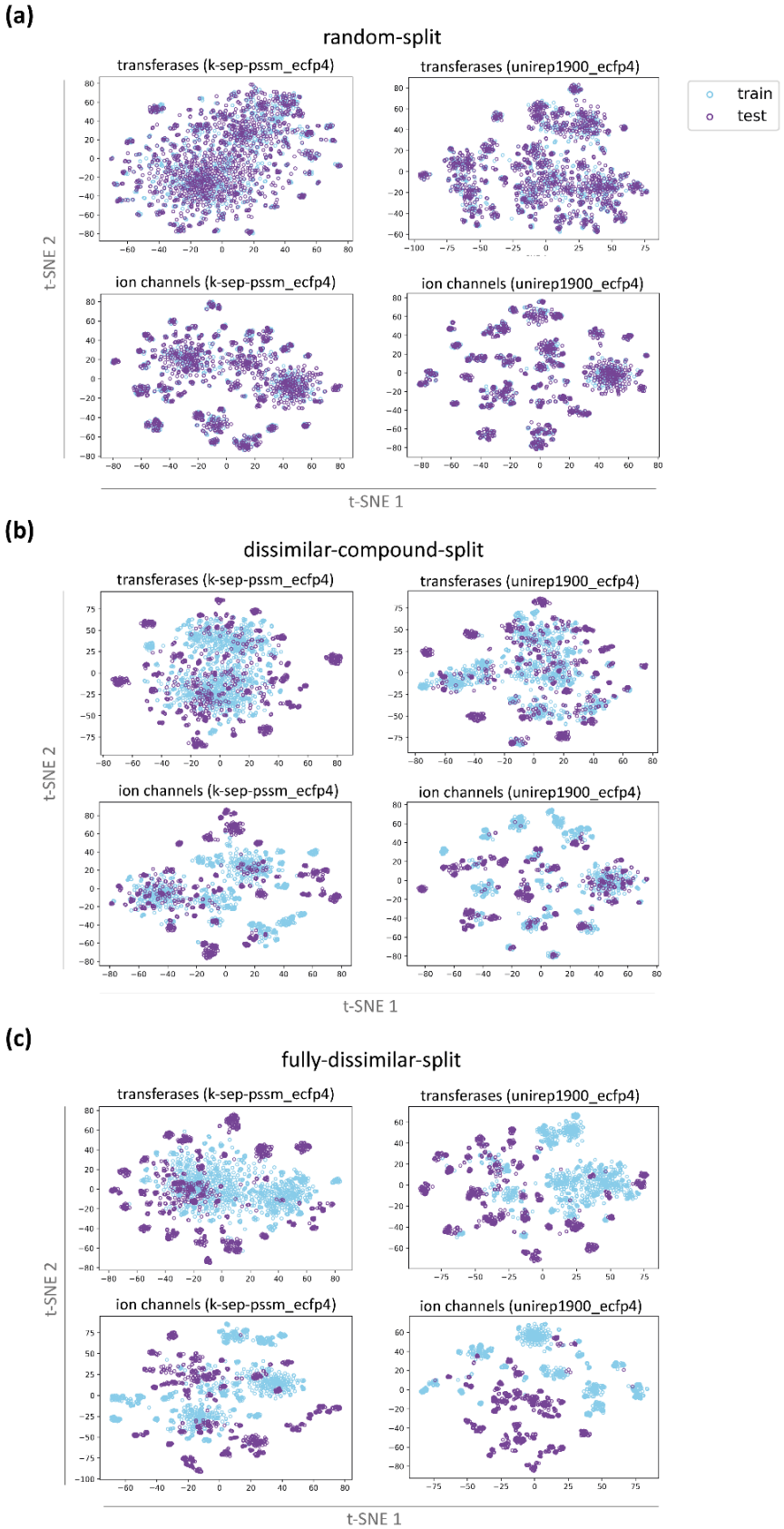

Figure 5. t-SNE visualization of train-test samples (i.e., bioactivity data points / compound-protein pairs) of transferase and ion channel families for k-sep_pssm and unirep1900 representations on; **(a)** the random-split, **(b)** dissimilar-compound-split, and **(c)** fully-dissimilar-split datasets.

3.2. Small-Scale Analysis (Target Feature-based Modelling)

There are numerous conventional descriptor sets for proteins in the literature, most of which can easily be utilized for DTI prediction. Evaluating all descriptor sets on our large-scale datasets would not be feasible considering the computational costs; as a result, we decided to carry out a small-scale pre-analysis to select the descriptor sets that are successful in DTI prediction, and use these selected descriptors in large-scale analysis later. As a second aim, it was required to determine the supervised learning algorithm to be used for DTI prediction in this study, and due to, again, the computational complexity related issues, we decided to make a performance comparison (between SVM and RF) on these small-scale datasets.

In this analysis, we assessed the success of SVM- and RF-based DTI prediction models of 42 conventional protein descriptor sets together with 2 baseline models (fed by the random200 descriptor set). The models are trained and tested on 9 independent compound-centric datasets (i.e., Curcumin, Tamoxifen, Quercetin, Genistein, Econazole, Levoketoconazole, Amiodarone, Miconazole, and Clotrimazole) via nested cross-validation using the target feature-based modelling approach (please see Methods 2.3.1). In this approach, the system only employs protein features as input, so it eliminates the effect of compound representations on the model prediction performance, which is expected to present a suitable setting for the comparison of protein representations. The task of each model is the binary classification of input proteins, as active or inactive, against the corresponding compound cluster.

Figure 6 displays mean F1-scores and MCC values of 9 datasets for each representation model, in which orange and blue colors correspond to SVM and RF models, respectively (all results including accuracy, precision, recall, F1-score, and MCC metrics are given in ESI Table S5). The ranking of protein descriptor sets on the horizontal axis was done according to decreasing RF model scores. Figure 6 clearly displays that RF models outperform SVM models with some exceptions including pfam model in terms of MCC score. When model performances are compared in terms of protein representations, pssm-based descriptors perform better than other descriptors in general. These results indicate that evolutionary relationships of proteins carry important knowledge regarding bioactivity mechanisms. Some sequence composition-based descriptors such as dde, tpc, and spmap, or physicochemistry-based descriptors such as apaac and paac also performed well. Moreover, obtaining scores that are higher than the baselines, even for the models with lowest performances, implies that protein representations carry signals/patterns relevant to bioactivity modelling, which are successfully utilized during the learning process. However, these results cannot be generalized as they cover only a small portion of the compound space; thus, it is important to observe how they will be affected with the change in data scale.

At the end of this analysis, we decided to continue with RF as the utilized learning algorithm throughout the study. Also, we selected 10 descriptor sets with both high and low performances, and distinct properties regarding the protein features they incorporated (i.e., apaac, ctdd, ctriad, dde, geary, k-sep_pssm, pfam, qso, spmap and taap) and used them in the following analyses.

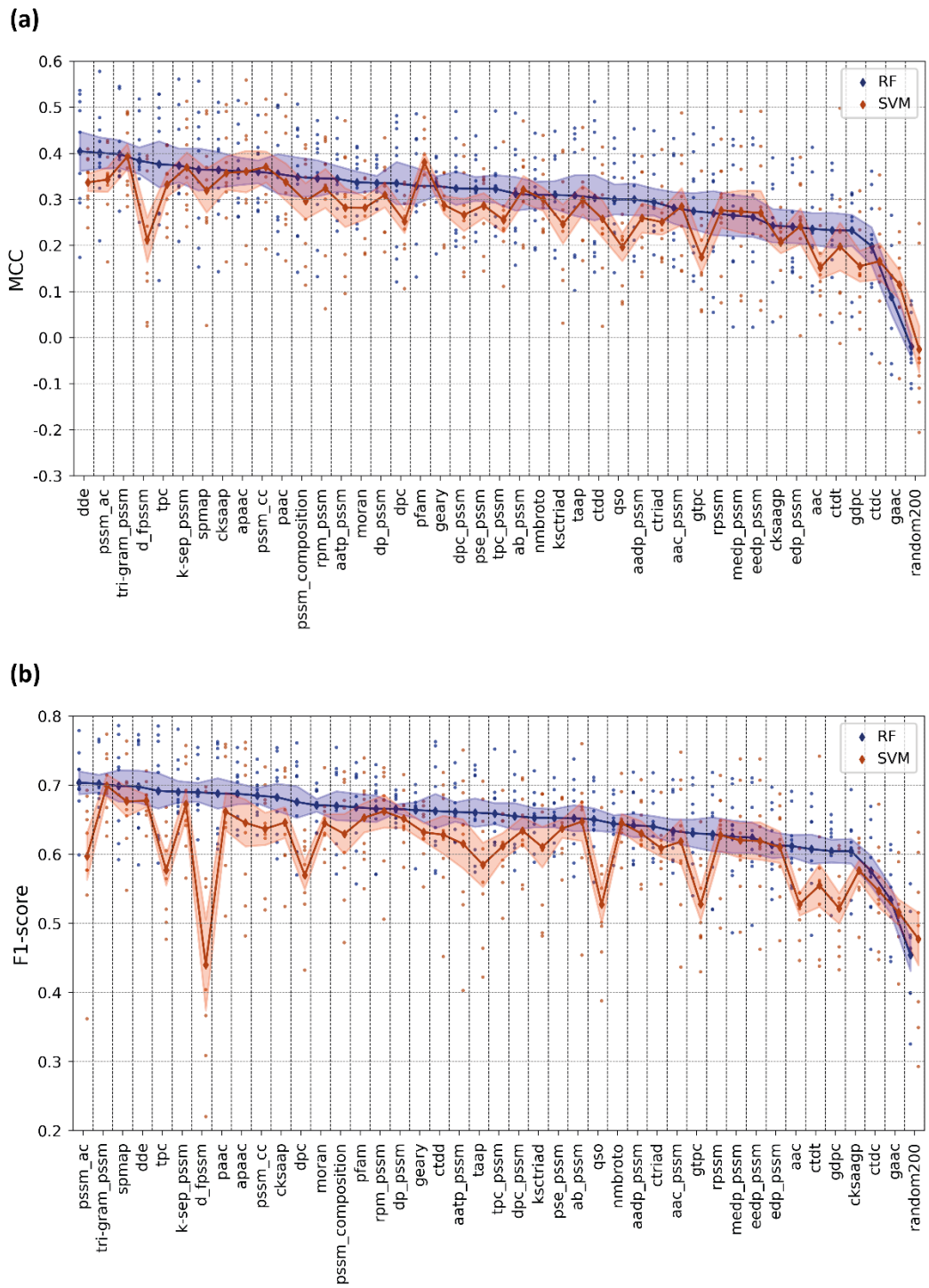

Figure 6. Mean **(a)** MCC and **(b)** F1-score test results of RF- and SVM-based DTI prediction models constructed via target feature-based modelling approach.

3.3. Medium-Scale Analysis (PCM Modelling)

PCM modelling approach can handle high numbers of training instances, belonging to different compounds and proteins, within a single predictive model, in contrast to ligand- and target feature-based modelling which requires the generation of separate models for each protein and compound (or compound cluster), respectively. This brings the advantage of learning from larger datasets, which is a critical requirement in machine learning, in general. Another advantage of PCM modeling is the joint utilization of compound and protein features to better model their interaction-related properties, without the requirement of 3-D structural information, unlike target-based structure modelling approaches. Here, we aimed to evaluate protein representations in terms of PCM modeling, over the problem of regression-based DTI prediction.

For this, we constructed PCM models for 10 conventional protein descriptor sets (selected based on the results of the previous analysis) and 6 learned protein embeddings using RF regression algorithm on the mDavis kinase dataset, which is a well-utilized benchmark dataset for machine learning-based DTI prediction. We also built 2 baseline models using randomly generated protein representations, within the same setting. In this benchmark study, the reason behind using a classical machine learning algorithm (i.e., RF) rather than more complex deep learning-based architectures is that: (i) RF is an algorithm that has been used in this field for a while and shown to work well, and (ii) these complex architectures have already been used in the training stage of these learned representations; thus, the use of another complex architecture in the supervised DTI prediction task could have prevent the observation of the ability of learned representations in extracting ligand interaction-related properties of proteins, and also, hinder the evaluation of model-driven (i.e., conventional descriptors) and data-driven (i.e., learned representations) approaches on a common ground.

Model performance results based on RMSE, Spearman rank correlation, MCC and F1-score (all computed on the hold-out test set of the mDavis dataset) are given in Figure 7 (also available in ESI Table S6). It indicates that the rankings of models are mostly consistent among both classification and regression metrics with slight differences, excluding pfam. As a domain profile-based descriptor set, pfam is the best performing model for F1-score (0.538) and has a moderately high MCC score (0.41), which also had satisfactory MCC and F1-score results in the previous target feature-based modelling; however, it is also one of the worst performers in terms of RMSE (0.854) and Spearman (0.497) scores. It can be inferred from these results that domain profiles of proteins might not contain sufficient information to make precise bioactivity value predictions, but it can be useful if the aim is to classify protein-compound pairs as active or inactive (i.e., binary prediction). The results also indicate that the seqvec model displays the best performance for almost all metrics (RMSE: 0.794, Spearman: 0.571, MCC: 0.445, F1-score: 0.53). Apart from seqvec, other learned embeddings have higher performance scores as well compared to

conventional descriptors in general with mean performances of 0.53 for learned representations and 0.51 for conventional descriptors in terms of Spearman rank correlation. Learned embeddings do not utilize any molecular or biological knowledge during their unsupervised protein feature vector construction, but still, they are capable of representing proteins to yield high performance DTI prediction. Well performing descriptors in the small-scale analysis, k-sep_pssm (homology) and apaac (physicochemistry) have competitive performance results (Spearman: 0.545 and 0.532, respectively) against learned embeddings. On the other hand, dde (Spearman: 0.508) and spmap (Spearman: 0.491) could not yield their high ranks on the medium-scale dataset (i.e., ranks of 1 and 8 on the small-scale, and 9 and 16 on the medium-scale). It is possible to state that while homology and physicochemistry are descriptor types that gained from increased dataset size (i.e., for apaac and k-sep-bigrams, small-scale analysis mean MCCs are 0.361 and 0.374, respectively, whereas their medium-scale analysis mean MCCs are 0.418 and 0.434), sequence composition could not obtain improved performances when trained on larger datasets.

Also, there is an overall increase in MCC scores of conventional descriptors (excluding dde and spmap) when we compare the results on small- and medium-scale datasets. In addition to the contribution of the increased sample size, this situation can be associated with the involvement of compound properties (along with protein features) in these PCM-based models, which probably led to a better learning over the joint protein-compound interaction space, and yielded higher performance results. On the other hand, PCM models had lower F1 scores than the target feature-based models. In order to calculate MCC and F1-scores for PCM models, we converted real-valued predictions into binary format at the cut-off value pChEMBL = 7, which is also used in other studies as a bioactivity threshold for kinase inhibitors [75]. However, only 27% of the test samples became active at this threshold, causing a class imbalance in the mDavis kinase dataset. Although F1-score is one of the most widely used metrics for classification tasks, it is not as resistant to the class imbalance problem as the MCC metric, which makes MCC a more appropriate option in this scenario. Similar to the target feature-based modelling results, baseline models displayed the lowest performances here.

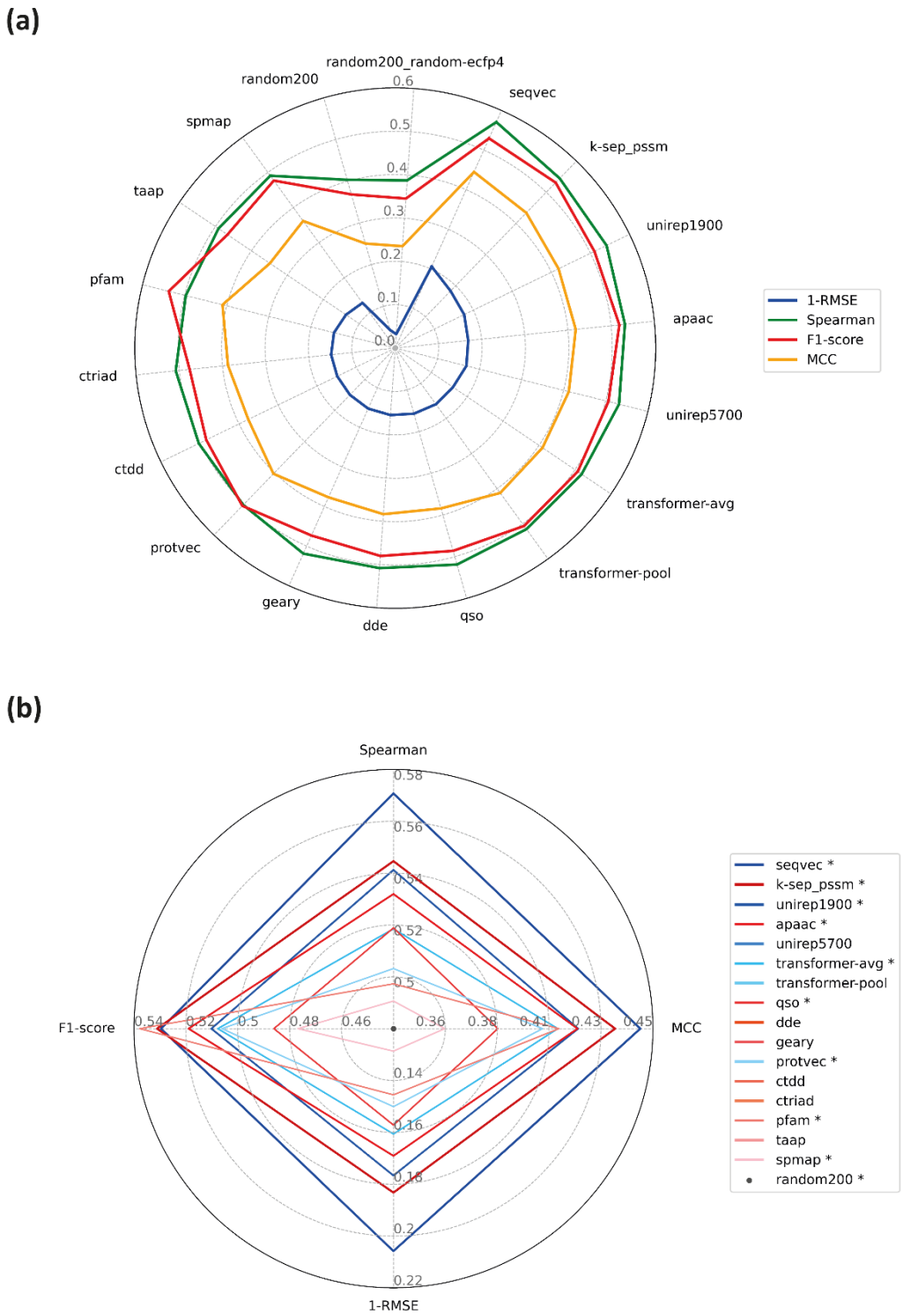

Figure 7. Test performance results of medium-scale PCM models (on the mDavis dataset) based on RMSE (the scores are reported as 1-RMSE, so higher values represent better performance), Spearman' s rank correlation, MCC and F1-score; **(a)** each color corresponds to an evaluation metric, and **(b)** scores are displayed only for the selected representative models (marked with asterisk in the legend). The ranking in the legend is based on the models' performance from best to worst according to their RMSE scores. Shades of red and blue represent conventional descriptors and learned representations, respectively.

3.4. Large-Scale Analysis (PCM Modelling)

The goal behind this analysis is evaluating protein representations over a highly realistic scenario, especially in terms of discovering new drugs and/or targets, using our carefully split large-scale train-test datasets, and to compare their overall performance in machine learning-based DTI prediction. We also aimed to display how model performances can change dramatically when the same samples are distributed to train and test sets differently, to point out the importance of train-test dataset preparation.

For the large-scale benchmark analysis of protein representations, we constructed protein family-specific bioactivity datasets including enzyme (i.e., transferases, hydrolases, oxidoreductases, proteases, and other enzymes) and non-enzyme groups (i.e., membrane receptors, ion channels, transporters, transcription factors, and epigenetic regulators). For each family, three versions of train-test splits with different difficulty levels were constructed by considering pairwise similarities of proteins and/or compounds (please see Methods 2.1.3 for details). PCM models were trained independently on each of these splits with the same protein representations used in the previous (medium-scale) analysis, and thus 600 DTI prediction models were built, trained, and tested in total (i.e., 3 dataset splits * 10 protein families * 20 representations, composed of 10 conventional descriptor sets, 6 learned embeddings and 4 baseline models).

We evaluated model performances from several perspectives using multiple scoring metrics. Median corrected RMSE and Spearman correlation scores are displayed in Figure 8, in which the light colored (transparent) circles correspond to individual model performances on each protein family, and the dark colored diamonds represent mean scores averaged over all families. The models are ranked according to the results of the fully-dissimilar-split dataset for both metrics. In Figure 9, model performances are evaluated with three different forms of the MCC metric. The models are ranked according to mean values of median corrected MCC scores for fully-dissimilar-split and dissimilar-compound-split datasets, and multiclass MCC scores for random-split dataset. Other performance tables are available in ESI Table S7.

Metric-based comparison

The intra-family rankings of models are generally consistent with each other among five different metrics (Table S7). However, there are some discrepancies between the scores depending on the split set. Considering regression metrics, some of the models trained/tested on the fully-dissimilar-split and dissimilar-compound-split datasets show high performance in terms of RMSE (i.e., low RMSE values), whereas at the same time, low Spearman correlations, which indicates inconsistency. In challenging scenarios (e.g., on the fully-dissimilar-split and dissimilar-compound-split datasets), continuous value-based prediction of bioactivities (via regression) is

unstable and unreliable due to the difficulty of the task. Thus, it would be a better choice to evaluate model performances on Spearman correlation scores on these challenging datasets, while RMSE could be a better option to differentiate models trained and tested on easy cases (i.e., the random-split dataset). In classification-based assessment, the binary class MCC metric is not as restrictive as the regression or multiclass evaluation metrics on challenging datasets since it is less sensitive to deviations in prediction values. However, it may suffer from the shifted mean problem in originally regression-based PCM models (see Methods 2.5). Obtaining MCC values close to 0 (Figure 9) despite moderate Spearman correlation scores (Figure 8) on challenging datasets is a sign of systematic shift on model prediction outputs, which we handled by conducting median correction on the real-valued prediction results, as explained in Methods 2.5. In Figure 9, it can be observed that median correction provided a significant increase in binary class MCC scores of fully-dissimilar-split and dissimilar-compound-split datasets. Also, median corrected MCC scores are highly consistent with the Spearman correlation scores (see Table S7). Considering the multiclass MCC metric, prediction scores are around zero for most of the models on challenging split sets. Since this metric is based on multiple thresholds, it is more restrictive than the binary classification metrics. However, this seems to be an advantage for evaluating models on the random-split set. As seen in Figure 9a, fluctuations among prediction scores of different models on the random-split set are greater considering multiclass MCC scores compared to the binary class MCC scores. Furthermore, the ranking is highly consistent with the results of the medium-scale experiments in which the top performers were learned representations, and k-sep_pssm and apaac descriptor sets. Thus, it can be inferred that this metric discerns models better than binary class MCC in random-split sets, and it tackles the over optimization problem in this splitting strategy, which especially occurs on the large-scale datasets.

Representation-based comparison

Performance results in Figure 8 and 9 clearly display that the representation capability of different protein descriptor sets depend on the protein family and the difficulty level of the split set used for training and testing. However, there is no significant difference between the performances of different protein descriptor sets for a particular dataset split, with a few exceptions. Considering family-based performance averages, pfam is one of the best representations on the fully-dissimilar-split and dissimilar-compound-split datasets, while it is the lowest performer on the random-split dataset (Figure 8 and 9). Contrary to pfam, k-sep_pssm is one of the best performers on the random-split and dissimilar-compound-split datasets but the worst one on the fully-dissimilar-split dataset (Figure 8 and 9), though the performance values on the random-split dataset are very close to each other. As a homology-based descriptor set, k-sep_pssm is expected to capture hidden similarities between evolutionarily related sequences, especially by taking advantage of the presence of highly similar proteins between the train and test

splits. On the other hand, the utilization of protein domain profiles seems to make pfam more suitable for acquiring bioactivity related information from evolutionarily distant sequences. Interestingly, taap, which is a simple descriptor set, is involved in the top-performing PCM models for all dataset splits. However, taap was one of the lowest performers in the small- (among the selected 10 out of the total 42 conventional descriptors) and medium-scale experiments. Its simplicity is observed to become an advantage with the increase in bioactivity dataset size and complexity. Apart from these, physicochemistry-based descriptors including apaac (in all split sets), ctriad (in fully-dissimilar-split dataset) and qso (in fully-dissimilar-split and dissimilar-compound-split datasets), and learned representations perform well in the large-scale experiments. In particular, the top performance results of unirep5700 and transformer-avg on the fully-dissimilar-split dataset demonstrate the potential of protein representation learning methods in the data-driven DTI prediction.

We also made family-specific evaluations here to understand whether different protein representations have the same effect across families. In Figure 10, we plotted the performance of the models of protease and the ion channel families, in terms of a conventional descriptor vs. learned representation comparison, using the Spearman and median corrected MCC scores on three dataset splits. For a fair comparison, we selected four well-performing conventional descriptors instead of including all of them since we have only four different types of learned representations. For this, we involved apaac, k-sep_pssm, pfam, and taap as conventional descriptors and protvec, seqvec, transformer-avg, and unirep5700 as learned representations. As displayed in Figure 10, learned representations outperform conventional descriptors in the challenging splits of proteases, considering both metrics. However, the results are the opposite for the ion channel family, in which conventional descriptors perform better than the learned representations. On the random-split dataset, there is no observable difference between conventional descriptor sets and the learned representations, probably due to the dataset split characteristics which resulted in extremely easy cases for the models. When taking all these results into account, we can clearly state that the representation capabilities of featurization approaches considerably vary among different protein families and splitting strategies, even though some common inferences can be made. We believe that, while choosing featurization approaches in DTI prediction, protein family-specific performances should be taken into account, rather than considering the overall (i.e., average) performances.

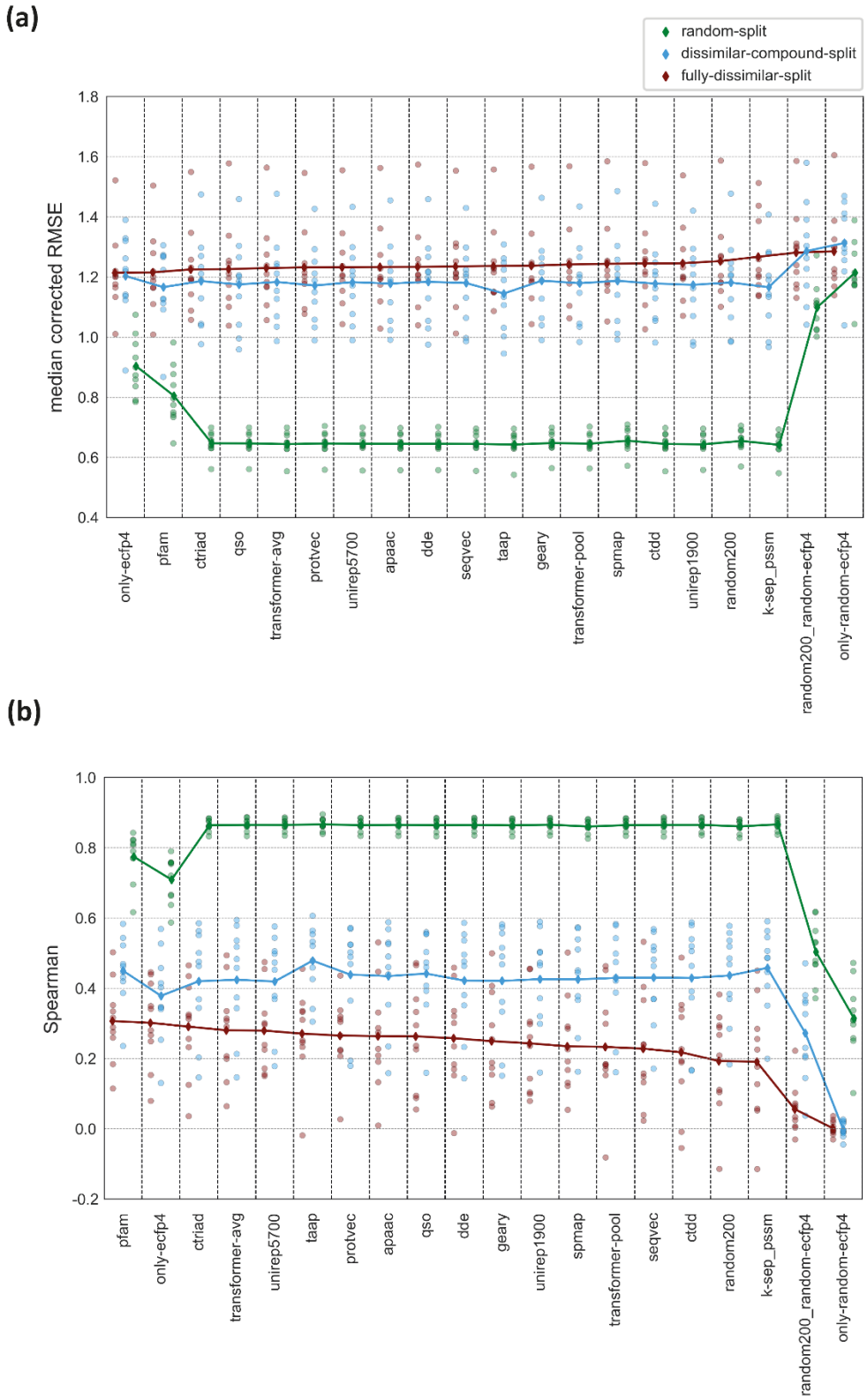

Figure 8. Regression-based test performance results of protein family-specific PCM models (each using a different representation type as input feature vectors) for random-split, dissimilar-compound-split, and fully-dissimilar-split datasets based on **(a)** median corrected RMSE, and **(b)** Spearman correlation scores. The models are ranked according to decreasing performance on the fully-dissimilar-split dataset.

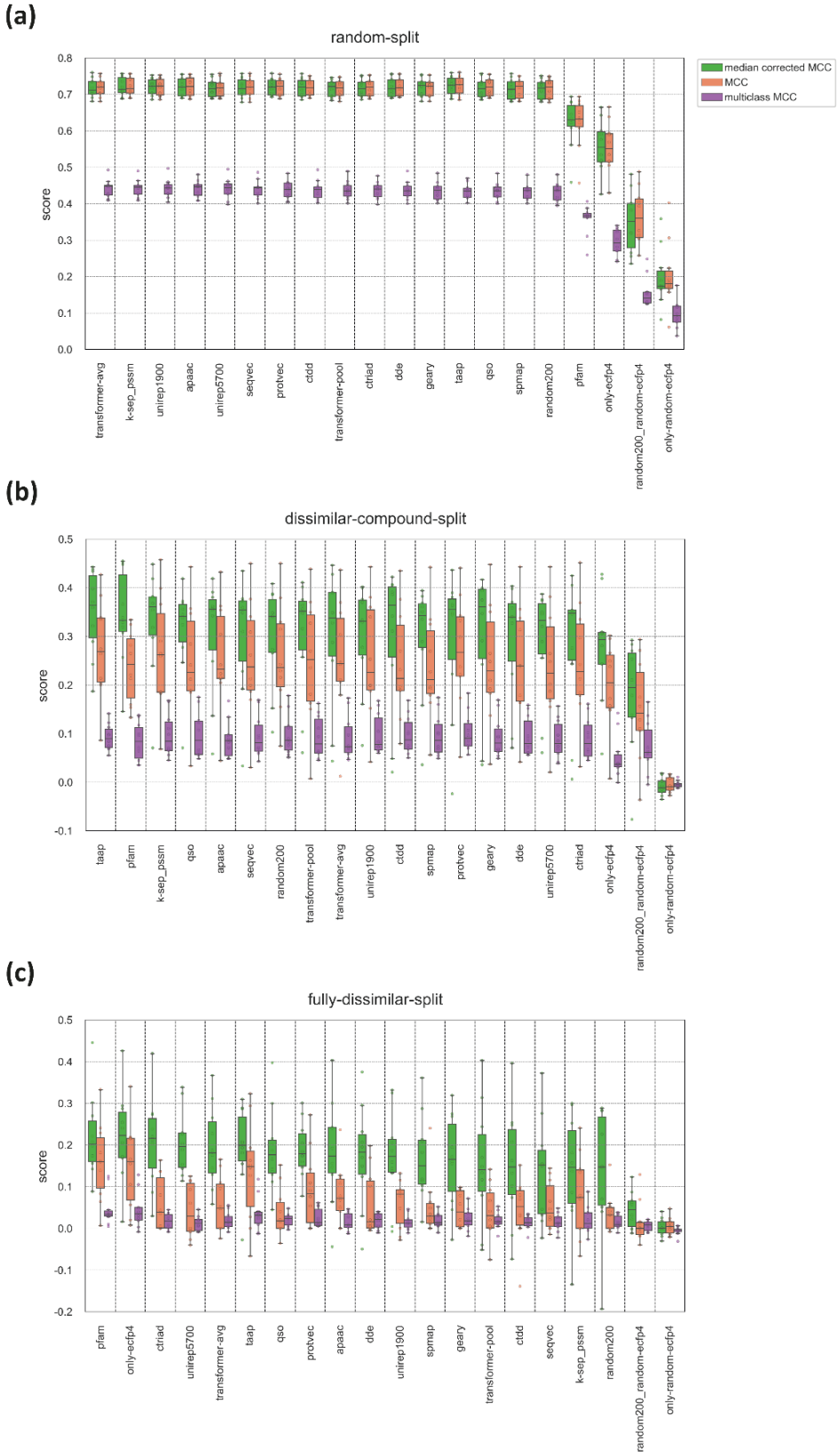

Figure 9. Classification-based test performance results of protein family-specific PCM models (each using a different representation type as input feature vectors) in terms of MCC scores for **(a)** random-split, **(b)** dissimilar-compound-split, and **(c)** fully-dissimilar-split datasets. The models are ranked according to decreasing performance on the fully-dissimilar-split dataset.

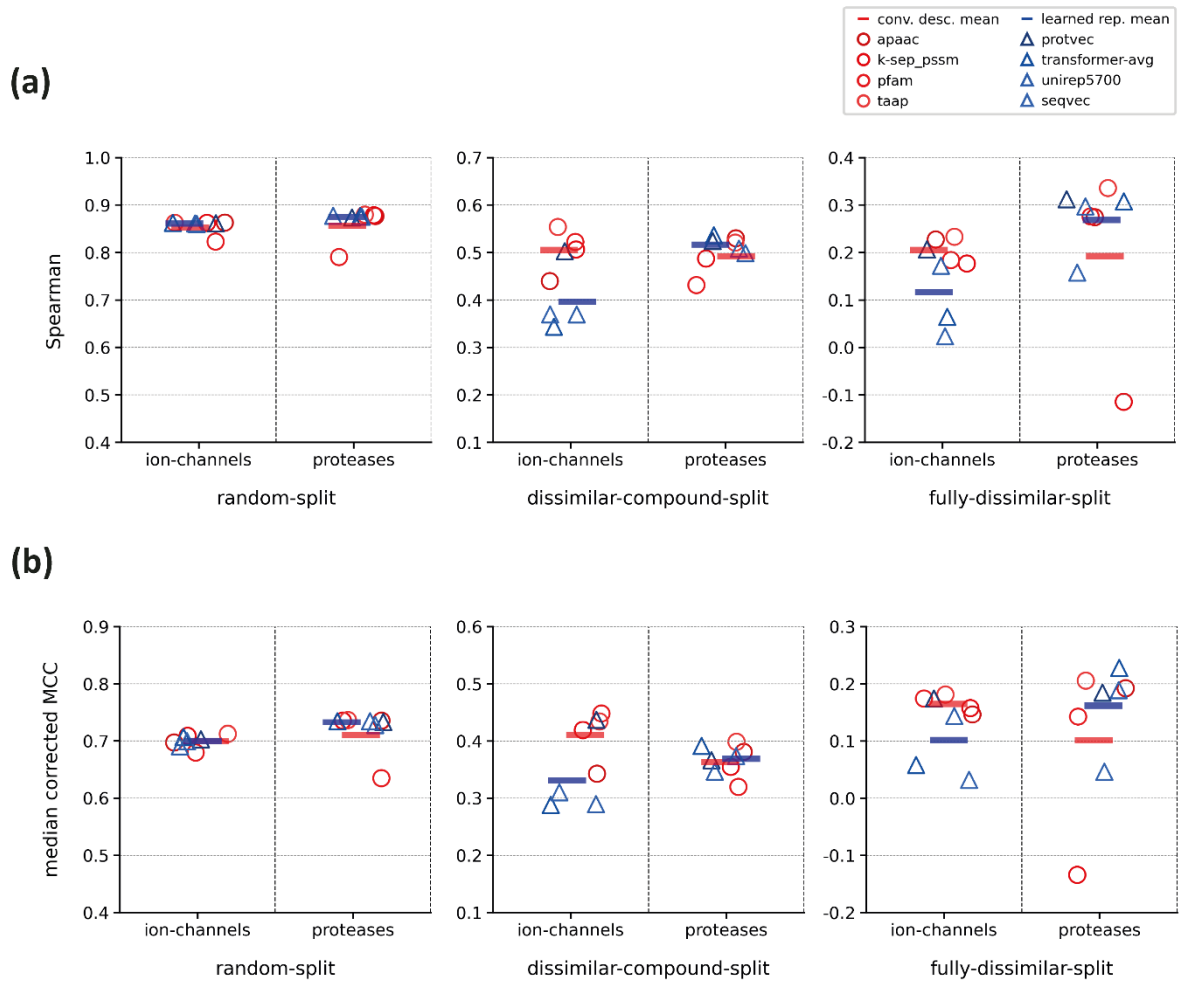

Figure 10. Comparison of well performing conventional descriptors and learned representations for three different splits of ion-channel and protease families in terms of; **(a)** Spearman correlation, and **(b)** median corrected MCC scores.

Dataset Split-based comparison

To compare models across three dataset splits, we plotted performance scores by pooling 200 models of each split (including baseline models) with grouping by neither families nor representations. The results are displayed in Figure 11 via violin plots. This figure shows that there is a significant decrease in overall performances with the increasing difficulty levels of splits, which is not a surprising outcome. Nevertheless, it highlights the importance of splitting train/test datasets in performance evaluation, to prevent the reporting of over-optimistic results and yield fair assessment of model successes. Figure 11 also displays that the differences between model performance scores in the fully-dissimilar-split and dissimilar-compound-split datasets are distributed more evenly over the whole range of scores, compared to the random-split dataset in which, most of the models produced very similar scores creating a dense region on the plot. This observation indicates the random-split dataset has less power in terms of distinguishing different descriptor sets/representations from each other.

In the fully-dissimilar split, neither similar proteins nor similar compounds are shared between train and test sets. As a result, this dataset is suitable to evaluate the performance of DTI prediction models in the cases of predicting novel ligands to understudied targets (or completely new candidate targets). Whereas in the dissimilar-compound split, similar proteins are presented in between train and test sets. Nevertheless, it is useful for discovering completely novel compounds against well-studied target proteins, or proteins for which structurally similar well-studied targets exist in the training set.

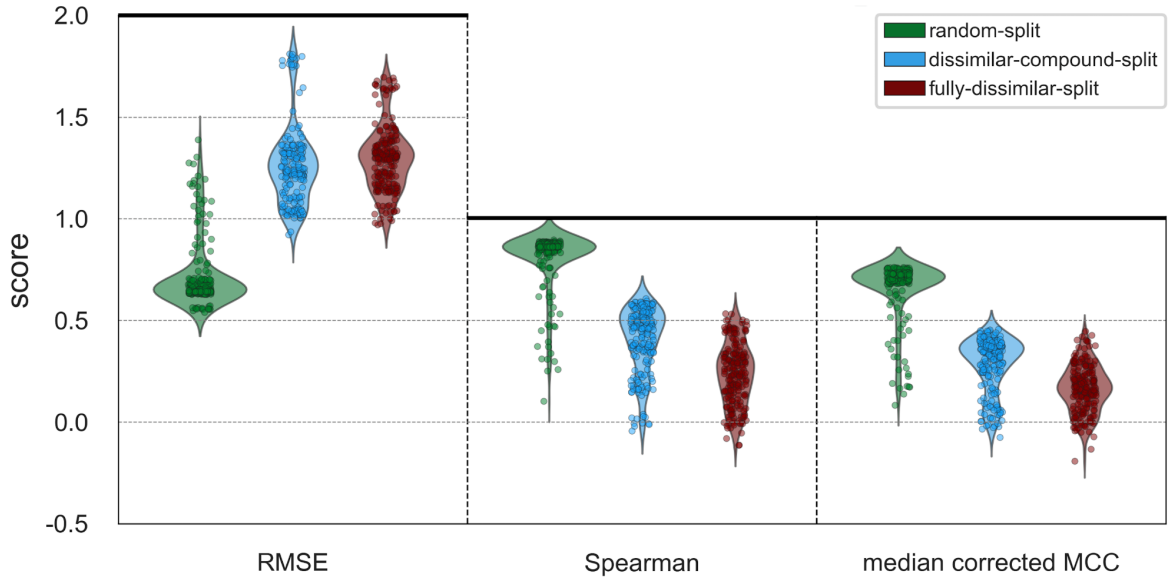

Figure 11. Split-based test performance scores of family-specific PCM models on RMSE, Spearman, and median corrected MCC metrics.

Comparison with baseline models

Table 2 involves family-based average Spearman scores of the best performing models and baseline models for each dataset split. The models based on randomly generated protein and/or compound representations have low performance scores on the fully-dissimilar-split dataset, which is mainly due to the absence of identical proteins and compounds (or ones with high similarity) in between train and test samples. On the other hand, the average Spearman correlation score of even the best performing model in this split is around 0.3, which is quite close to the model that uses only compound representations (i.e., only-ecfp4 model). It is important to note that the only-ecfp4 model does not utilize protein vectors. As a result, the model learns activities over only the compound vectors without any information regarding which protein this compound interacts with. This indicates that this low amount of success is coming from learning irrelevant features from compound vectors. The fully-dissimilar-split dataset is reflecting the performance in terms of discovering novel drug candidate compounds against understudied target proteins that have little

or no bioactivity data for training. Thus, these results reveal the need for novel and improved featurization techniques to construct robust DTI prediction models that can be utilized in the pharmaceutical industry, especially considering these difficult cases.

Due to the inclusion of similar (or identical) proteins, model performances are higher on the dissimilar-compound-split dataset compared to the fully-dissimilar-split dataset. Also, the models based on completely randomly generated vectors (on both the compound and protein sides) have lower performances, expectedly. On both of the challenging datasets, the best model is well differentiated from the baseline models. Although the overall mean difference between the best model and random200 model is considerably low on the dissimilar-compound-split, the differences are distinct when making protein family-specific comparisons rather than taking averages of all families (e.g., for ion channels; average Spearman score of the top performing models including k-sep_pssm, pfam, taap, and protvec is 0.52, and Spearman score of random200 model is 0.37). On the dissimilar-compound-split, random200 model outperformed the only-ecfp4 model by learning the relationship between the bioactivity data points of the same proteins (which are shared between training and test). As experimental bioactivity measurements are mainly obtained from target-based assays, the number of bioactivity data points per protein is considerably high, compared to the number of bioactivity data points per compound. Also, in many assays, different derivatives of the same compound are tested, which result in similar bioactivity values. Due to this bias in experimental assays, memorization over protein identity yields falsely successful results, as reflected in the performance of the random200 model on the dissimilar-compound-split (average Spearman score = 0.436).

On the random-split dataset, the best model displays a high success rate (Spearman score: 0.868). However, high performance scores of the baseline models, including those based on randomly generated representations (e.g., only-ECFP4 and random200), clearly indicates the overoptimistic evaluation and emphasizes the importance of train-test data splitting, once again. These results also demonstrate the importance of baseline model-based investigation in DTI prediction, for a fair and realistic performance evaluation.

Table 2. Protein family-based average Spearman scores of the best models and baseline models in each dataset split.

Name of the descriptor set/representation (with explanation)	**Fully-dissimilar-split**	**Dissimilar-compound-split**	**Random-split**
the best performing protein representation (compound side: ECFP4)	0.363	0.518	0.868
random200 (protein side: random continuous vectors, compound side: ECFP4)	0.193	0.436	0.861
only-ecfp4 (no protein vector at all, compound side: ECFP4)	0.302	0.379	0.709
random200-random-ecfp4 (protein side: random continuous vectors, compound side: random binary vectors)	0.056	0.272	0.504
only-random-ecfp4 (no protein vector at all, compound side: random binary vectors)	0.002	-0.002	0.315

Exploration of the prediction similarities between family-specific PCM models

We conducted a heatmap analysis based on the pairwise similarities between the protein family-specific PCM model predictions via calculating intersections in between, based on a categorization into six different classes (i.e., pChEMBL-based bioactivity predictions in the bins of <5, 5.0 to 5.5, 5.5 to 6.0, 6.0 to 6.5, 6.5 to 7.0, and 7.0>=). To clarify differences between representations, color scales were arranged so that the darkest color corresponds to the maximum value, and the lightest color was set to 85%, 65%, and 20% similarity for the random-split, dissimilar-compound-split, and fully-dissimilar-split datasets, respectively.

In Figure 12, heatmaps of transferases and ion channels are given for random-split, dissimilar-compound-split, and fully-dissimilar-split datasets (heatmaps for the remaining families are available in ESI Figure S3). The consensus between models decreases with increasing difficulty levels. To be more precise, there is an overlap of over 80% for most of the models in the random-split dataset while this value drops to 30-60% in the fully-dissimilar-split dataset. Considering both families, models that utilize pfam and taap descriptor sets are quite differentiated from the others on the random-split and dissimilar-compound-split datasets. Although clusters vary across different splits and protein families, generally the learned embeddings and physicochemistry-based conventional descriptors are clustered among themselves (i.e., considering the fully-dissimilar-split dataset of transferases; the average prediction similarity between the models that utilize learned representations - transformer-avg, transformer-pool, seqvec, unirep1900, unirep5700- is 60.8%, and among the models that use physicochemistry-based conventional descriptor sets - qso, apaac, geary, ctriad- is 68.2%, whereas the average prediction similarity

between the physicochemistry-based conventional vs. learned representations - considering the same models- is 46.5%), which is also parallel to the t-SNE projection results provided above (Figure 2). Considering the type of utilized information, learned representations are similar to each other in the sense that they employ a data-driven approach and mainly exploit sequence similarities. On the other hand, physicochemistry-based descriptors employ a model-driven approach and aggregate pre-calculated amino acid-based features to obtain protein representations. This difference is also reflected in their prediction similarities. Spmap and random200 representations are also often clustered together and have similar t-SNE projections, as well.

This analysis can be used to obtain rational combinations of different featurization approaches, which may increase the chance of capturing the relevant properties of proteins from different perspectives, with the aim of improving the overall model performances.

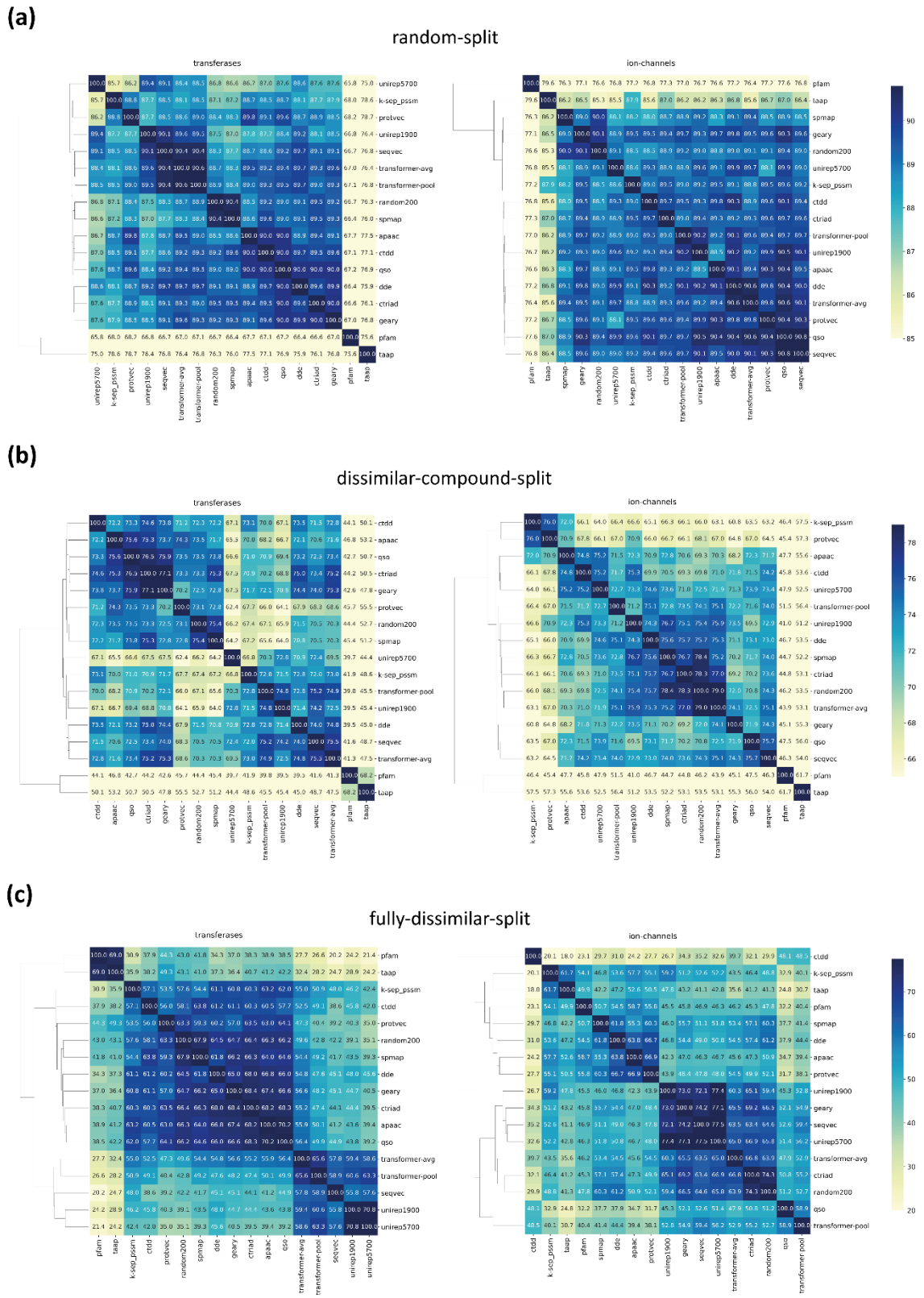

Figure 12. Clustered heatmaps of different protein representation approaches for transferase and ion channel families on; **(a)** the random-split, **(b)** the dissimilar-compound-split, and **(c)** the fully-dissimilar-split datasets.

4. Conclusion

In this study, we performed a rigorous benchmark analysis to investigate the representation capability of various protein featurization techniques on DTI prediction modelling. For a multi-perspective evaluation, we built target feature-based and PCM-based models, and trained/tested them on carefully constructed datasets with varying sizes and difficulty levels. Below, we summarized the major contributions of our study to the literature:

(i) We proposed challenging benchmark datasets with high coverage on both compound and protein spaces that can be used as reliable, reference/gold-standard datasets for DTI modelling tasks. These datasets are protein family-specific, and each has three versions in terms of train/test folds/splits for different prediction task scenarios (i.e., random split for predicting known inhibitors for known targets, dissimilar-compound split for predicting novel inhibitors for known targets, and fully-dissimilar split for predicting new inhibitors for new targets). Thus, they yield fair evaluation of models at multiple difficulty levels and facilitate the elimination of models with over-optimistic performance results.

(ii) We employed a network-based strategy for splitting train/test folds of these protein family-specific datasets, by considering both protein-protein and compound-compound pairwise similarities, which is proposed for the first time to our knowledge. This strategy ensures that train and test folds are totally dissimilar from each other with a minimum loss of data points. One of the current limitations in drug development is the problems related to discovering novel molecules that are structurally different from existing ones. The network-based splitting strategy we applied here forces prediction models to face this limitation by supplying more realistic, hard-to-predict test samples. Hence, it can aid researchers in designing more powerful and robust DTI prediction models that have a real translational value.

(iii) Protein representation learning methods have a wide range of applications with promising results in different sub-fields of protein science, despite being a relatively new approach. However, the studies regarding their usage in DTI prediction modelling are limited, and there is no comprehensive benchmark study to evaluate their performance against well-known and widely used representation approaches. Due to this reason, we extended the scope of our study by involving state-of-the-art protein representation learning methods and discussed their potential in DTI prediction.

One of the critical observations of this study is the dramatic change in performance scores when the samples are distributed to train and test sets differently, (i.e., scores on datasets with challenging splits are significantly lower compared to the results on random-split based datasets), which highlights the importance of splitting train/test datasets to conduct realistic evaluations for drug and/or target discovery. This study also emphasizes the importance of exploratory analysis of datasets and the usage of

multiple scoring metrics as well as the inclusion of baseline models for a proper discussion of model successes.

Regarding the performance-based comparison of different protein featurization approaches, it is not possible to put forward an outstanding representation method, as their success largely depends on the dataset and the applied splitting strategy. Although both conventional descriptors and learned embeddings have their own strengths and weaknesses depending on the case, competitive results of learned embeddings display their potential wide-spread utilization in drug discovery and development in the near future. On the other hand, considerably low performance results on challenging dataset splits (e.g., fully-dissimilar) in the overall evaluation reveal the requirement of further improved protein featurization techniques to capture hidden and complex features shared between highly distant homologs.

As future work, we plan to develop a new computational DTI prediction approach that utilizes numerous types of biological/biomedical entities on top of compounds and target proteins. Drug discovery and development is composed of a series of complex problems, as there are multiple factors affecting the success of a drug candidate. This is mainly due to the extremely dynamic and complicated structure of biological systems. As a result, it is not possible to computationally handle drug discovery solely by simple virtual screening. Considering this fact, taking a systems-based approach with the integration and utilization of direct and indirect relationships in molecular and cellular processes including protein-protein interactions, drug/compound-target protein interactions, and signaling/metabolic pathways, together with high level concepts such as protein-disease relationships, drug-disease indications, pathway-disease modulations, and phenotypic implications could increase the success rate in drug discovery. Thus, we aim to construct a new type of systems-level DTI representation and subsequent prediction, using CROssBAR [76] which is an open-source system that integrates large-scale biological/biomedical data and represents them in the form of heterogeneous and computable knowledge graphs. The newly proposed DTI prediction system will utilize graph representation learning algorithms to process these biomedical knowledge graphs, and will be trained, validated/optimized, and tested on our realistic and challenging datasets, and compared to other prominent models in the literature.

We hope that the results of this study, together with the data-driven approaches proposed, and datasets prepared and shared, will aid the ongoing work in computational drug discovery and repurposing, in the context of representing biomolecules for high performance drug-target interaction prediction.

References

1. A. S. Rifaioglu, H. Atas, M. J. Martin, R. Cetin-Atalay, V. Atalay and T. Doğan, *Brief. Bioinform.*, 2019, **20**, 1878–1912.
2. A. S. Rifaioglu, E. Nalbat, V. Atalay, M. J. Martin, R. Cetin-Atalay and T. Doğan, *Chem. Sci.*, 2020, **11**, 2531–2557.
3. A. Lavecchia and C. Di Giovanni, *Curr. Med. Chem.*, 2013, **20**, 2839–60.
4. I. Cortés-Ciriano, Q. U. Ain, V. Subramanian, E. B. Lenselink, O. Méndez-Lucio, A. P. IJzerman, G. Wohlfahrt, P. Prusis, T. E. Malliavin, G. J. P. van Westen and A. Bender, *Medchemcomm*, 2015, **6**, 24–50.
5. Y. Tabei, E. Pauwels, V. Stoven, K. Takemoto and Y. Yamanishi, *Bioinformatics*, 2012, **28**, 487–494.
6. T. Qiu, J. Qiu, J. Feng, D. Wu, Y. Yang, K. Tang, Z. Cao and R. Zhu, *Brief. Bioinform.*, 2017, **18**, 125–136.
7. A. Cereto-Massagué, M. J. Ojeda, C. Valls, M. Mulero, S. Garcia-Vallvé and G. Pujadas, *Methods*, 2015, **71**, 58–63.
8. I. Muegge and P. Mukherjee, *Expert Opin. Drug Discov.*, 2016, **11**, 137–148.
9. R. Sawada, M. Kotera and Y. Yamanishi, *Mol. Inform.*, 2014, **33**, 719–731.
10. K.-C. Chou, *Bioinformatics*, 2005, **21**, 10–19.
11. S. A. Ong, H. H. Lin, Y. Z. Chen, Z. R. Li and Z. Cao, *BMC Bioinformatics*, 2007, **8**, 300.
12. G. J. P. van Westen, R. F. Swier, I. Cortes-Ciriano, J. K. Wegner, J. P. Overington, A. P. I. Jzerman, H. W. T. Van Vlijmen and A. Bender, *J. Cheminform.*, 2013, **5**, 41.
13. M. Sun, X. Wang, C. Zou, Z. He, W. Liu and H. Li, *BMC Bioinformatics*, 2016, **17**, 231.
14. D. Wu, Q. Huang, Y. Zhang, Q. Zhang, Q. Liu, J. Gao, Z. Cao and R. Zhu, *BMC Bioinformatics*, 2012, **13**, 212.
15. V. Saravanan and N. Gautham, *Omi. A J. Integr. Biol.*, 2015, **19**, 648–658.
16. L. Perlman, A. Gottlieb, N. Atias, E. Ruppin and R. Sharan, *J. Comput. Biol.*, 2011, **18**, 133–145.
17. T. Doğan, E. A. Güzelcan, M. Baumann, A. Koyas, H. Atas, I. R. Baxendale, M. Martin and R. Cetin-Atalay, *PLOS Comput. Biol.*, 2021, **17**, e1009171.
18. Y. Yamanishi, E. Pauwels, H. Saigo and V. Stoven, *J. Chem. Inf. Model.*, 2011, **51**, 1183–1194.
19. T. Doğan, *PeerJ*, 2018, **6**, e5298.
20. T. Doğan, A. Macdougall, R. Saidi, D. Poggioli, A. Bateman, C. O'Donovan and M. J. Martin, *Bioinformatics*, 2016, **32**, 2264.
21. H. Saini, G. Raicar, S. Lal, A. Dehzangi, S. Imoto and A. Sharma, *J. Softw.*, 2016, **11**, 756–767.
22. S. Unsal, H. Atas, M. Albayrak, K. Turhan, A. C. Acar and T. Doğan, *Nat. Mach. Intell.*, 2022, .
23. E. Asgari and M. R. K. Mofrad, *PLoS One*, 2015, **10**, 141287.
24. E. C. Alley, G. Khimulya, S. Biswas, M. AlQuraishi and G. M. Church, *Nat. Methods*, 2019, **16**, 1315–1322.
25. M. Heinzinger, A. Elnaggar, Y. Wang, C. Dallago, D. Nechaev, F. Matthes and B. Rost,

BMC Bioinformatics, 2019, **20**, 723.

26. C. Mirabello and B. Wallner, *PLoS One*, 2019, **14**, e0220182.
27. R. Rao, N. Bhattacharya, N. Thomas, Y. Duan, X. Chen, J. Canny, P. Abbeel and Y. S. Song, in *33rd Conference on Neural Information Processing Systems*, 2019.
28. P. T. Kim, R. Winter and D. A. Clevert, *Int. J. Mol. Sci.*, 2021, **22**, 12882.
29. H. Öztürk, E. Ozkirimli and A. Özgür, *arXiv*, 2019, 1902.04166.
30. A. S. Rifaioglu, R. Cetin Atalay, D. Cansen Kahraman, T. Doğan, M. Martin and V. Atalay, *Bioinformatics*, 2021, **37**, 693–704.
31. A. Dutta, T. Dubey, K. K. Singh and A. Anand, *Comput. Biol. Chem.*, 2018, **74**, 434–441.
32. R. You, X. Huang and S. Zhu, *Methods*, 2018, **145**, 82–90.
33. N. Strodthoff, P. Wagner, M. Wenzel and W. Samek, *Bioinformatics*, 2020, **36**, 2401.
34. Q. U. Ain, O. Méndez-Lucio, I. C. Ciriano, T. Malliavin, G. J. P. van Westen and A. Bender, *Integr. Biol.*, 2014, **6**, 1023–1033.
35. G. J. van Westen, R. F. Swier, I. Cortes-Ciriano, J. K. Wegner, J. P. Overington, A. P. Ijzerman, H. W. van Vlijmen and A. Bender, *J. Cheminform.*, 2013, **5**, 42.
36. Y. Xu, D. Verma, R. P. Sheridan, A. Liaw, J. Ma, N. M. Marshall, J. McIntosh, E. C. Sherer, V. Svetnik and J. M. Johnston, *J. Chem. Inf. Model.*, 2020, **60**, 2773–2790.
37. E. B. Lenselink, N. Ten Dijke, B. Bongers, G. Papadatos, H. W. T. Van Vlijmen, W. Kowalczyk, A. P. Ijzerman and G. J. P. Van Westen, *J. Cheminform.*, 2017, **9**, 45.
38. S. Liang and H. Yu, *Bioinformatics*, 2020, **36**, 4490–4497.
39. A. Mayr, G. Klambauer, T. Unterthiner, M. Steijaert, J. K. Wegner, H. Ceulemans, D. A. Clevert and S. Hochreiter, *Chem. Sci.*, 2018, **9**, 5441–5451.
40. Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing and V. Pande, *Chem. Sci.*, 2018, **9**, 513–530.
41. Q. Ye, C. Y. Hsieh, Z. Yang, Y. Kang, J. Chen, D. Cao, S. He and T. Hou, *Nat. Commun. 2021*, 2021, **12**, 1–12.
42. A. Gaulton, A. Hersey, M. Nowotka, A. P. Bento, J. Chambers, D. Mendez, P. Mutowo, F. Atkinson, L. J. Bellis, E. Cibrián-Uhalte, M. Davies, N. Dedman, A. Karlsson, M. P. Magariños, J. P. Overington, G. Papadatos, I. Smit and A. R. Leach, *Nucleic Acids Res.*, 2017, **45**, D945–D954.
43. S. Jasial, Y. Hu, M. Vogt and J. Bajorath, *F1000Research*, 2016, **5**, 591.
44. M. I. Davis, J. P. Hunt, S. Herrgard, P. Ciceri, L. M. Wodicka, G. Pallares, M. Hocker, D. K. Treiber and P. P. Zarrinkar, *Nat. Biotechnol.*, 2011, **29**, 1046–1051.
45. H. Öztürk, A. Özgür and E. Ozkirimli, *Bioinformatics*, 2018, **34**, i821–i829.
46. B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey and C. H. Wu, *Bioinformatics*, 2015, **31**, 926–932.
47. G. Landrum, RDKit: Open-Source Cheminformatics Software, http://www.rdkit.org/.
48. A. Hagberg, P. Swart and D. S Chult, *Exploring Network Structure, Dynamics, and Function using NetworkX*, United States, 2008.
49. V. D. Blondel, J. L. Guillaume, R. Lambiotte and E. Lefebvre, *J. Stat. Mech. Theory Exp.*, 2008, **2008**, P10008.
50. I. Dubchak, I. Muchnik, C. Mayor, I. Dralyuk and S.-H. Kim, *Proteins Struct. Funct. Genet.*, 1999, **35**, 401–407.

51. J. Shen, J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, Y. Li and H. Jiang, *Proc. Natl. Acad. Sci. U. S. A.*, 2007, **104**, 4337–41.
52. R. C. Geary, *Inc. Stat.*, 1954, **5**, 115–146.
53. Z. R. Li, H. H. Lin, L. Y. Han, L. Jiang, X. Chen and Y. Z. Chen, *Nucleic Acids Res.*, 2006, **34**, W32–W37.
54. J. Wang, B. Yang, J. Revote, A. Leier, T. T. Marquez-Lago, G. Webb, J. Song, K.-C. Chou and T. Lithgow, *Bioinformatics*, 2017, **33**, 2756–2758.
55. S. El-Gebali, J. Mistry, A. Bateman, S. R. Eddy, A. Luciani, S. C. Potter, M. Qureshi, L. J. Richardson, G. A. Salazar, A. Smart, E. L. L. Sonnhammer, L. Hirsh, L. Paladin, D. Piovesan, S. C. E. Tosatto and R. D. Finn, *Nucleic Acids Res.*, 2019, **47**, D427–D432.
56. H. Liu, J. Sun, J. Guan, J. Zheng and S. Zhou, *Bioinformatics*, 2015, **31**, i221–i229.
57. G. Schneider and P. Wrede, *Biophys. J.*, 1994, **66**, 335–344.
58. K.-C. Chou, *Biochem. Biophys. Res. Commun.*, 2000, **278**, 477–483.
59. O. S. Sarac, O. Gürsoy-Yüzügüllü, R. Cetin-Atalay and V. Atalay, *Comput. Biol. Chem.*, 2008, **32**, 122–30.
60. A. S. Rifaioglu, T. Doğan, Ö. S. Saraç, T. Ersahin, R. Saidi, M. V. Atalay, M. J. Martin and R. Cetin-Atalay, *Proteins Struct. Funct. Bioinforma.*, 2018, **86**, 135–151.
61. A. Dalkiran, A. S. Rifaioglu, M. J. Martin, R. Cetin-Atalay, V. Atalay and T. Doğan, *BMC Bioinformatics*, 2018, **19**, 1–13.
62. S. Kawashima, P. Pokarowski, M. Pokarowska, A. Kolinski, T. Katayama and M. Kanehisa, *Nucleic Acids Res.*, 2008, **36**, D202–D205.
63. M. M. Gromiha and M. Suwa, *Biochim. Biophys. Acta - Proteins Proteomics*, 2006, **1764**, 1493–1497.
64. Z. Chen, P. Zhao, F. Li, A. Leier, T. T. Marquez-Lago, Y. Wang, G. I. Webb, A. I. Smith, R. J. Daly, K.-C. Chou and J. Song, *Bioinformatics*, 2018, **34**, 2499–2502.
65. P. Zhang, L. Tao, X. Zeng, C. Qin, S. Y. Chen, F. Zhu, S. Y. Yang, Z. R. Li, W. P. Chen and Y. Z. Chen, *J. Mol. Biol.*, 2017, **429**, 416–425.
66. A. Vaswani, G. Brain, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, in *31st Conference on Neural Information Processing Systems*, 2017.
67. D. Rogers and M. Hahn, *J. Chem. Inf. Model.*, 2010, **50**, 742–754.
68. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, M. Brucher and É. Duchesnay, *J. Mach. Learn. Res.*, 2011, **12**, 2825–2830.
69. L. Van Der Maaten and G. Hinton, *J. Mach. Learn. Res.*, 2008, **9**, 2579–2605.
70. D. Chicco and G. Jurman, *BMC Genomics*, 2020, **21**, 1–13.
71. M. Waskom, *J. Open Source Softw.*, 2021, **6**, 3021.
72. J. D. Hunter, *Comput. Sci. Eng.*, 2007, **9**, 90–95.
73. P. Rice, I. Longden and A. Bleasby, *Trends Genet.*, 2000, **16**, 276–277.
74. A. Dalke, *J. Cheminformatics 2019 111*, 2019, **11**, 1–21.
75. A. Cichońska, B. Ravikumar, R. J. Allaway, F. Wan, S. Park, O. Isayev, S. Li, M. Mason, A. Lamb, Z. Tanoli, M. Jeon, S. Kim, M. Popova, S. Capuzzi, J. Zeng, K. Dang, G. Koytiger, J. Kang, C. I. Wells, T. M. Willson, T. I. Oprea, A. Schlessinger, D. H. Drewry, G. Stolovitzky, K. Wennerberg, J. Guinney and T. Aittokallio, *Nat. Commun. 2021 121*,

2021, **12**, 1–18.

76. T. Doğan, H. Atas, V. Joshi, A. Atakan, A. S. Rifaioglu, E. Nalbat, A. Nightingale, R. Saidi, V. Volynkin, H. Zellner, R. Cetin-Atalay, M. Martin and V. Atalay, *Nucleic Acids Res.*, 2021, **49**, e96.