PROTEOCHEMOMETRIC MODELS USING MULTIPLE SEQUENCE ALIGNMENTS AND A SUBWORD SEGMENTED MASKED LANGUAGE MODEL

A PREPRINT

Héléna A. Gaspar * **Mohamed Ahmed** **Thomas Edlich**

Benedek Fabian **Zsolt Varszegi** **Marwin Segler** **Joshua Meyers** **Marco Fiscato**

BenevolentAI
4-8 Maple St, Bloomsbury
London W1T 5HD

May 17, 2021

ABSTRACT

Proteochemometric (PCM) models of protein-ligand activity combine information from both the ligands and the proteins to which they bind. Several methods inspired by the field of natural language processing (NLP) have been proposed to represent protein sequences. Here, we present PCM benchmark results on three multi-protein datasets: protein kinases, rhodopsin-like GPCRs (ChEMBL binding and functional assays), and cytochrome P450 enzymes. Keeping ligand descriptors fixed, we evaluate our own protein embeddings based on subword-segmented language models trained on mammalian sequences against pre-existing NLP-based descriptors, protein-protein similarity matrices derived from multiple sequence alignments (MSA), dummy protein one-hot encodings, and a combination of NLP-based and MSA-based descriptors. Our results show that performance gains over one-hot encodings are small and combining NLP-based and MSA-based descriptors increases predictive performance consistently across different splitting strategies. This work has been presented at the 3rd RSC-BMCS / RSC-CICAG Artificial Intelligence in Chemistry in September 2020.

***K*eywords** Proteochemometric Modelling · PCM · Language Modelling · Transformer · QSAR

1 Introduction

Ligand descriptors such as Morgan fingerprints, physico-chemical properties and pharmacophoric features have been used extensively for quantitative structure-activity relationship (QSAR) modelling to predict the binding of small molecules to a protein target [1]. Proteochemometric (PCM) models [2, 3] aim to predict the activity of molecules for multiple proteins simultaneously by incorporating both ligand and protein input terms. These multi-protein models can then be used for target prediction to support deconvolution efforts following phenotypic screens, or in hit-finding where there is too little data to build a single target QSAR model, but a useful predictive model can be achieved after incorporating data from related protein targets.

Existing methods to encode the protein include 1D sequence descriptors, 3D protein descriptors, and protein-ligand cross-terms. Protein-ligand interaction fingerprints are an example of 3D protein-ligand cross-terms that require experimental complexes or docked poses as inputs [4]. Protein-ligand interaction hotspots could theoretically be predicted without 3D information using coupling or attention terms learned by a neural network [5].

*Contact email: helena.gaspar@benevolent.ai

3D descriptors require some knowledge of the 3D protein structure and are generally more computationally intensive - they also have much less data available; working with raw sequences allows to build models with millions of proteins.

1D protein sequence descriptors can be processed using traditional amino acid featurizers [6], but are hard to apply to very long amino acid sequences. Protein-protein similarity matrices derived from multiple sequence alignments (MSA) can be an easy way to provide fixed-size features for proteins: each protein can be encoded by its corresponding row in a square similarity matrix. These similarities can be derived from sequence identities, or evolutionary divergence based on an amino acid substitution matrix such as blocks substitution matrix (BLOSUM) [7]. The recent MSA Transformer [8] can provide embeddings from a deep network trained on multiple sequence alignments, based on the Transformer architecture [9].

Recent deep learning techniques for natural language processing (NLP) have permitted to build protein embeddings based uniquely on the protein sequence, such as ProtVec [10] or UniRep [11]. ProtVec, based on word2vec [12], was trained on ∼550K SwissProt sequences, whereas UniRep, based on a multiplicative long short-term memory (LSTM) architecture [13], was trained on 24M UniRef sequences [14].

Training NLP models on large text corpora can be extremely costly, particularly with modern Transformer-derived architectures. This problem is exacerbated in the domain of protein sequences since a single protein may contain up to 35K amino acids. The way the sequence is segmented before being processed by the model becomes of paramount importance for manageable mini-batch learning. N-gram encodings are often used instead of individual amino acids [10]; however, n-grams lack meaning, do not vary in size (e.g. 3-grams), and do not address the sequence length issue.

In our NLP models, we use byte-pair-encoding (BPE) [15] to generate a fixed vocabulary of protein tokens. BPE is a subword segmentation algorithm that iteratively generates subwords based on their frequencies in a training corpus. BPE-segmented protein sequences are of a more manageable size and offer a principled way to tokenize proteins based on a corpus that can be chosen to bias the learning process. In our experiments, we use the full set of 3M consensus mammalian sequences available in UniProtKB [16] to generate the subwords and obtain a generic vocabulary for mammalian sequences.

Another issue impacting cost and performance is the choice of a training set for protein sequences and underlying algorithm. We investigate both LSTM and Transformer encoders for our NLP models, and for data efficiency use a representative, non-redundant training set of 1.5M sequences selected from the 3M set; this set is expected to be relevant for human disease whilst allowing us to build models in a matter of days on a single GPU.

2 Methods

2.1 Data and Descriptors

	KINOME	GPCRA	CYP
proteins	321	159	5
ligands	73,642	58,157	16,343
protein-ligand pairs	130,265	88,540	79,245

Table 1: Kinome, GPCRA and CYP datasets

We curated three datasets (cf. Table 1) from ChEMBL (version 26) [17] for protein kinases (KINOME), rhodopsin-like GPCRs (GPCRA), and cytochrome P450 enzymes (CYP).

KINOME and GPCRA contain activity data from heterogeneous sources including all protein-ligand pairs with an assigned value for pChEMBL, which corresponds to activity types with -log10 (molar IC50, XC50, EC50, AC50, Ki, Kd or Potency). On the other hand, CYP only contains functional inhibition AC50s from a single PubChem assay (AID 1851) [18]. Binary activity labels were given to each protein-ligand pair: active if pChEMBL > 6 for KINOME and GPCRA (a threshold deemed relevant for hit id and hit expansion), and pAC50 > 5 for CYP (the recommended threshold according to the assay description). For conflicting duplicates, the most frequent class was selected.

All molecules were standardized using the ChemAxon Standardizer (http://www.chemaxon.com), and proteins were mapped to their UniProt [16] canonical sequences. Pairs with little confidence in the data validity comment or with low ChEMBL confidence score (< 8) were removed. Molecules with empty activity field but labelled as inactive in the activity comment were kept and labelled as inactive.

Class imbalance is an important problem to address for PCM models as each protein has a different label distribution. As the datasets were heavily imbalanced we undersampled the majority class for each individual protein using clustering-based diversity selection to obtain the same number of actives and inactives.

The input for the model is a concatenation of ligand encodings (ECFC4 fingerprints) and protein encodings. In future work, other ligand encodings, e.g. based on graph neural networks or chemical language models could be considered [19, 20]. We used RDKit to generate the fingerprints (http://www.rdkit.org). Seven protein descriptors were investigated (Table 2). SP-Multihead and SP-RNN are our internal models described in the following section. To compute MSA-derived protein-protein similarity matrices, we generated one MSA per investigated protein family, using Clustal Omega [21]: human protein kinases, human rhodopsin-like GPCRs, mammalian cytochrome P450s.

SP-Multihead	our masked language model trained on 1.5M UniProt mammal sequences
SP-RNN	our LSTM next word prediction model trained on 3M UniProt mammal sequences
MSA	MSA-derived pair-wise sequence identity matrix
SP-Multihead+MSA	a concatenation of SP-Multihead encodings and sequence identities
ProtVec	word2vec-based encoding trained on ∼550K Swiss-Prot sequences
UniRep	language model based on multiplicative LSTMs trained on 24M UniRef sequences
OneHot	one-hot encoding of human sequences

Table 2: Benchmarked protein descriptors

2.2 NLP models based on sequence segmentation

We implemented two NLP models for proteins using pytorch v0.4.1 [22]: SP-Multihead and SP-RNN. They both take as input BPE segmentations of the protein sequence and were trained for 20 epochs on 1 GPU, keeping the model with best validation loss. Both models also required 4 CPUs, SP-Multihead 80GB memory and SP-RNN 8GB.

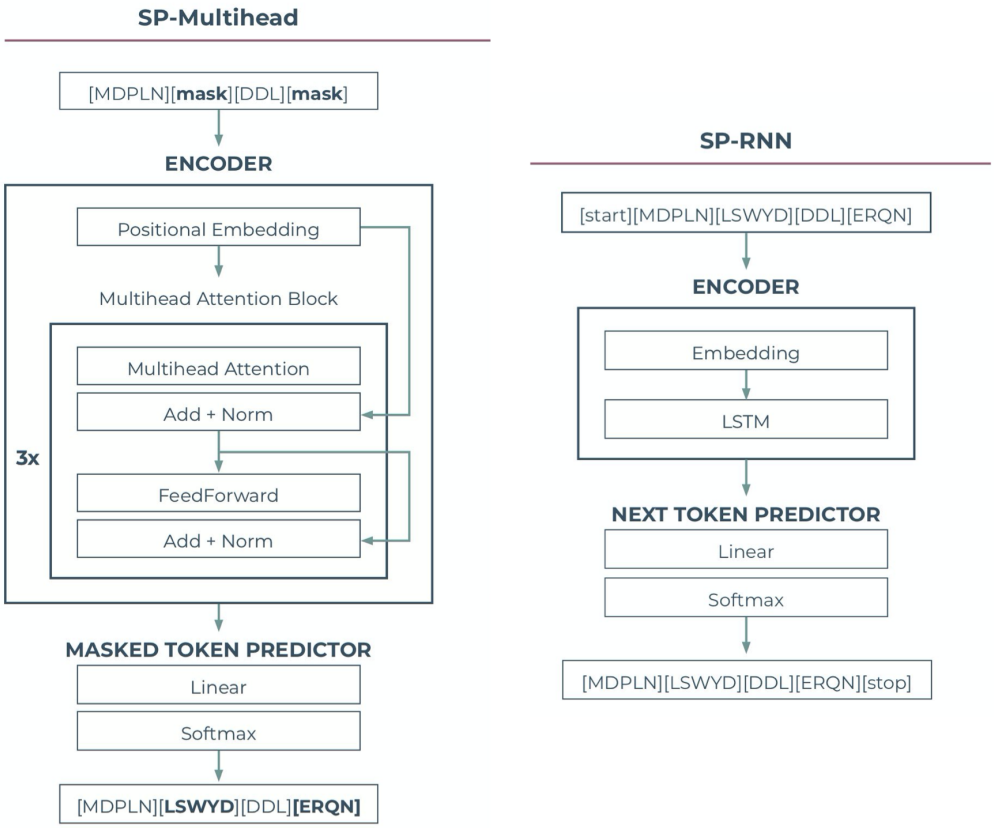

Figure 1: SP-Multihead: masked language model implementing a transformer-like encoder whereas. SP-RNN: LSTM-based next word predictor.

The BPE vocabulary was trained using SentencePiece [23] v0.1.9 on the complete corpus of 3M mammalian sequences in UniProtKB [16], incorporating both SwissProt and TrEMBL sets. SwissPort is a set of manually annotated and reviewed sequences, whereas TrEMBL have not yet been manually annotated.

A subset of representative sequences with less than 30% sequence identity was constructed using mmseq2 [24], divided into train (1.5M) and valid sets (15K), and used to train the two networks.

SP-Multihead is a masked language model based on the Transformer Encoder [9]. We used BERT [25] settings for the masked language model task: 15% of the tokens are used for prediction, of which 80% are masked, 10% are assigned random tokens and 10% stay the same. A fixed positional encoding was defined using sine and cosine functions of different frequencies as described in the Transformers paper [9]. The length of the positional encoding was set to 600 during training, as we found that most proteins sequences did not exceed 600 BPE tokens. At inference time, the positional encoding length can be adjusted for longer protein sequences. Other hyperparameters included embedding size = 512, batch size = 64, learning rate = 0.001.

The SP-RNN network is a next token predictor based on an LSTM encoder with similar hyperparameters and embedding size = 512. It was used to provide a baseline with a simpler architecture.

2.3 Modelling and validation

PCM models are paired input models. In this paper, we use a concatenation of ligand and sequence features (Figure 2).

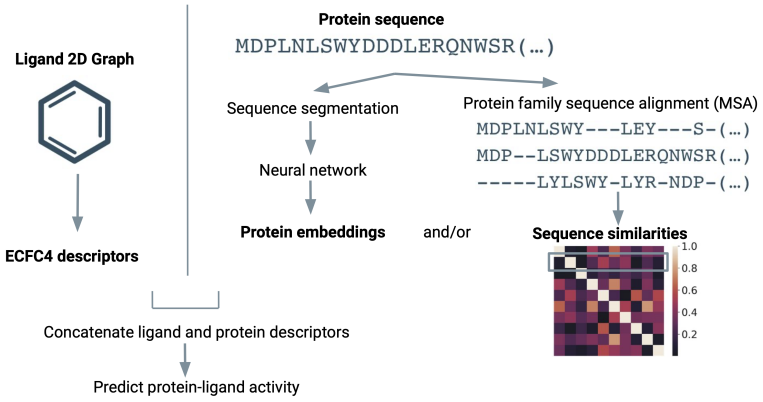

Figure 2: The input of PCM models is a concatenation of ligand and sequence features.

To figure out how protein descriptors perform in different experimental settings, depending on the amount of information available from the protein or ligand side, different validation strategies must be applied. According to Park and Marcotte [26], models using paired data should be validated with three different settings where one, both, or neither pair members can be found in the training set. However, in practical applications we may not necessarily be interested in all three tasks at the same time (cf. Discussion). This validation scheme can be further extended for PCM by generating clustered splits instead of splits based on individual proteins or ligands. Considering the limited quantity of binding data available, we decided to keep only four different experiments with increasing difficulty levels:

- Random splits
- Leave ligands out: ligands in the training set are absent from the test set
- Leave clusters of ligands out, with clusters defined k-means clustering [27] on ErG descriptors [28]
- Leave proteins out: proteins in the test set are absent from the training set

ErG descriptors were generated using RDKit (http://www.rdkit.org), and k-means clusters using scikit-learn [29]. We exclude the experiment where clusters of proteins are held out as this removed too much information from the training set, the experimental protein-ligand matrix being very sparse.

For each splitting strategy, five folds were constructed for cross-validating Random Forests models, using scikit-learn [29]. Nine parameter settings were evaluated for each descriptor with different combinations of maximum depth ($[25, 50, 75]$) and number of estimators ($[25, 50, 75]$).

3 Results

The performance achieved for each protein descriptor and splitting strategy is reported in Figures 3,4 as measured by the Matthews Correlation Coefficient (MCC), and a global ranking across datasets is reported in Table 3.

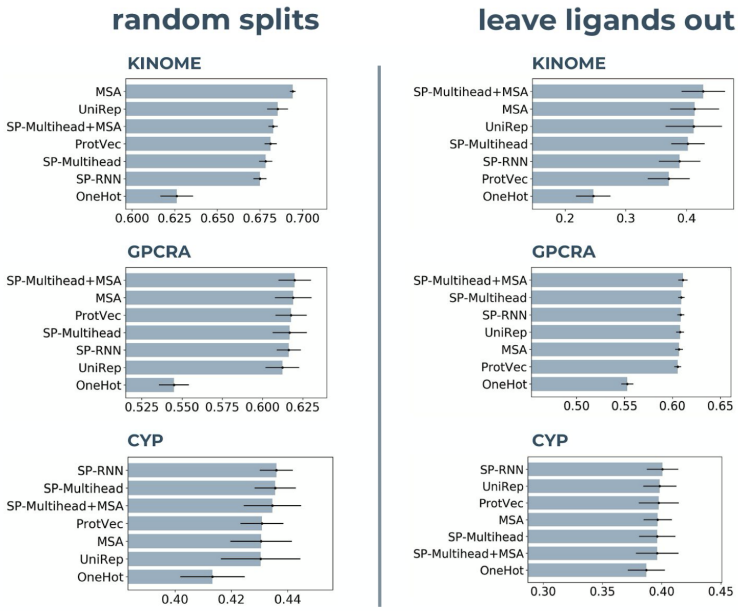

Figure 3: Protein descriptor performance: averaged Matthews Correlation Coefficient (MCC) for best hyperparameter combination in 5-fold cross-validation, with random splits or leave ligands out splitting strategies. Errors bar indicate $\pm$ one standard deviation from the mean across all folds. The x-axes were individually scaled.

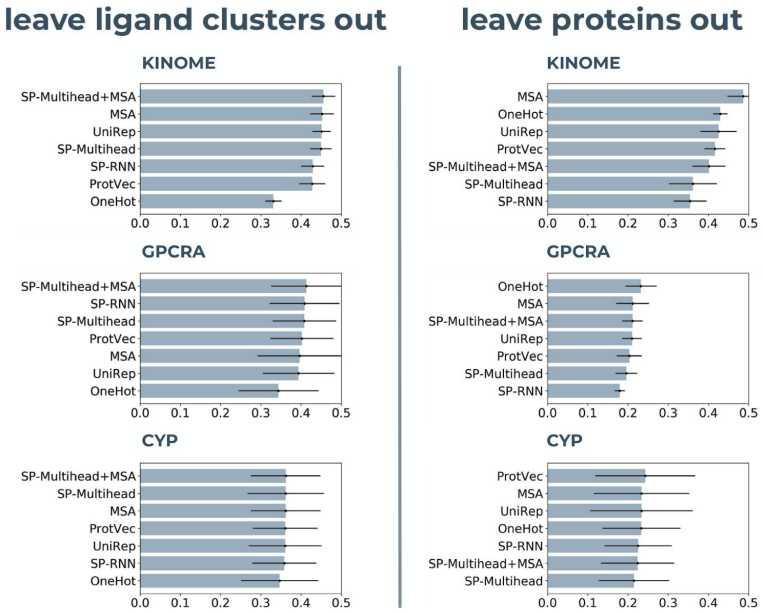

Figure 4: Protein descriptor performance: averaged Matthews Correlation Coefficient (MCC) for best hyperparameter combination in 5-fold cross-validation, with leave ligand clusters out or leave proteins out splitting strategies. The errors bar indicate $\pm$ one standard deviation from the mean across all folds. The x-axes were individually scaled.

The difference between worst and best descriptor combinations for PCM models is small (between 0.1-0.2 MCC units) and axes in Figures 3,4 are scaled to highlight differences. The one-hot baseline is usually the worst in terms of performance, and a combination of SP-Multihead and MSA the best. However, this is not the case for the "leave proteins out" task, where MSA identities and one-hot encodings perform better than other descriptors, and UniRep performs best amongst NLP descriptors.

random splits

SP-Multihead+MSA	2.3
MSA	2.7
ProtVec	3.7
SP-Multihead	3.7
SP-RNN	4
UniRep	4.7
OneHot	7

leave ligands out

SP-Multihead+MSA	2.7
SP-RNN	3
UniRep	3
SP-Multihead	3.7
MSA	3.7
ProtVec	5
OneHot	7

leave clusters out

SP-Multihead+MSA	1
SP-Multihead	3
MSA	3.3
SP-RNN	4.3
UniRep	4.7
ProtVec	4.7
OneHot	7

leave proteins out

MSA	1.7
OneHot	2.3
UniRep	3.3
ProtVec	3.3
SP-Multihead+MSA	4.7
SP-RNN	6.3
SP-Multihead	6.3

Table 3: Protein descriptor rank averaged across three datasets (Kinome, GPCRA, CYP) for each splitting strategy.

4 Discussion

4.1 PCM model usage

In practical applications, different splitting strategies and descriptors should be used depending on the use case and data available.

If the goal is to obtain predictions for ligands for which some data is already available (e.g. known activities for similar ligands in the training set), or to build a broadly applicable selectivity model, random splits or leave ligands out cross-validation can be used to identify suitable hyperparameters, and a combination of NLP-based and MSA-based descriptors may be useful as protein descriptors. On the other hand, if the only data available against a specific protein of interest is from dissimilar ligands, the "leave ligand clusters out" splitting strategy may be more relevant for hyperparameter tuning, depending on the sparsity of the experimental protein-ligand activity matrix. The "leave proteins out" strategy mimics the situation where there are no known data against a target of interest and in this setting, the NLP-based descriptors benchmarked in this paper do not seem more useful than MSA-based descriptors or simple one-hot encodings.

Our results only show global performances but not per-protein performance; performances will largely fluctuate depending on the initial protein package, and using applicability domain methods or at least per-protein performance estimations is of paramount importance in real-life applications for drug discovery projects. In Annex 2, we show how PCM compares to single-protein QSAR models for 149 protein kinases.

4.2 Protein descriptors

Our internal NLP-based descriptors have similar performances to UniRep or ProtVec for PCM, and we do not see a large difference between SP-Multihead and SP-RNN, i.e. Transformer or LSTM encoders. It should be noted that although ProtVec, SP-Multihead and SP-RNN are trained with fewer sequences than UniRep, a similar performance is attained, suggesting that representation learning with additional sequence data may not be necessary although there may be a balance between amount of data and noise from irrelevant sequences to be further investigated. There is a scarcity of literature evidence on the impact of using information from other organisms in PCM models - we chose to focus on mammalian sequences to build our NLP models, which are evolutionary closer to humans and expected to provide

translatable information. On the other hand, our three datasets used for PCM modelling only included human data, and it would be potentially interesting to try building a model with other related organisms to enrich the dataset.

When no ligand binding a protein of interest is included in the training set ("leave proteins out" task), our results show that the NLP-based descriptors benchmarked in this paper do not help predictions. In this specific setting, inclusion of binding site information or 3D descriptors of the protein might provide complementary information, so that the model might learn to derive translatable ligand features from the protein pocket information.

5 Conclusion

We studied the ability of PCM models to predict protein-ligand activity using three datasets of protein kinases, rhodopsin-like GPCRs, and cytochrome P450 enzymes. A number of splitting tasks were designed to simulate the available data landscape in different prospective settings. We show that a combination of SP-Multihead descriptors, based on a transformer encoder, and MSA-derived sequence identities achieve the best predictive performance across our datasets for three tasks: random splits, leaving ligands out, and leaving clusters of ligands out. We see different results for the hardest task "leave proteins out" where one-hot encodings or MSA-derived sequence identities achieve the best performance and NLP-based methods do not provide much gain. We also present a data-efficient language model for proteins trained using mammalian sequences and a pretrained sentence-piece tokenizer, for specific applications to drug discovery. In future work, we are interested in investigating binding site descriptors, combining MSA/NLP-based descriptors with 3D descriptors of the ligands and the protein, and exploring different architectures to take into account specific protein-ligand interaction hotspots.

6 Appendix

6.1 Length of BPE-segmented human sequences

The lenghs of protein sequences in UniProtKB segmented using the BPE algorithm are shown in Figure S1. Most sequences have less tan 600 tokens - we set 600 as the maximum length for our NLP models based on BPE-segmented sequences.

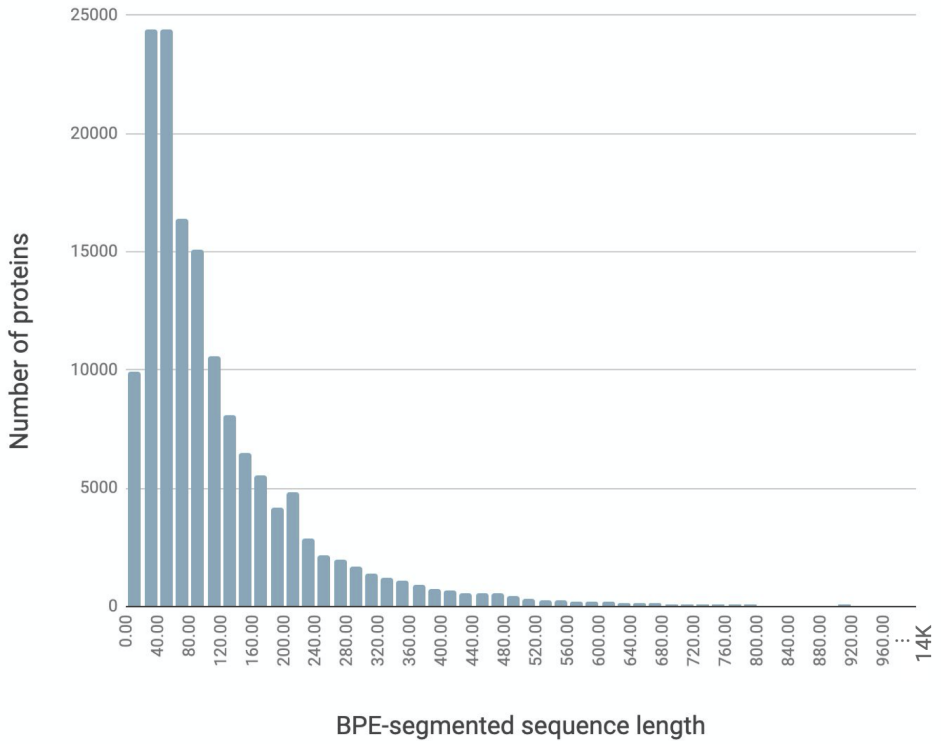

Figure S1: Length of BPE-segmented sequences of human proteins in UniprotKB

6.2 PCM versus single-target QSAR for 149 kinases

A PCM model was built using a subset of 149 protein kinases with at least 100 known ligands (> 50 actives); 80% of the data for each kinase was randomly included in the training set, 20% held out to build the external test set. In parallel, 149 single-target QSAR models were individually optimized. Both single-target and PCM models were built using Random Forests and optimized with 5-fold random cross-validation, and protein descriptors for the PCM model were arbitrarily set as MSA-derived protein-protein similarities. The PCM and single-target models with the best performing hyperparameters (greatest MCC) were then used to predict each external test set per protein. Our results show that protein-ligand binding is better predicted with PCM for 74% of tested targets (110/149), although the difference is generally small.

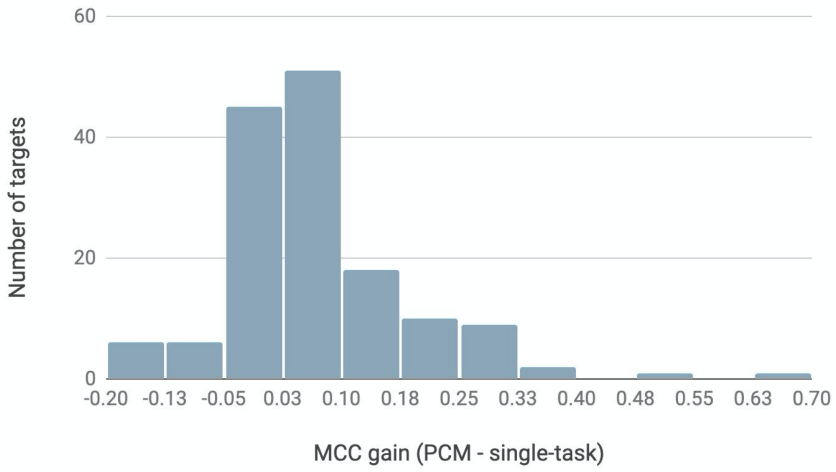

Figure S2: Difference in Matthews Correlation Coefficient between PCM and single target QSAR model for 149 protein kinases, on a 20% holdout

References

[1] Alexander Tropsha. Best practices for QSAR model development, validation, and exploitation. *Molecular Informatics*, 29(6-7):476–488, July 2010.

[2] Isidro Cortés-Ciriano, Qurrat Ul Ain, Vigneshwari Subramanian, Eelke B. Lenselink, Oscar Méndez-Lucio, Adriaan P. IJzerman, Gerd Wohlfahrt, Peteris Prusis, Thérèse E. Malliavin, Gerard J. P. van Westen, and Andreas Bender. Polypharmacology modelling using proteochemometrics (PCM): recent methodological developments, applications to target families, and future prospects. *MedChemComm*, 6(1):24–50, 2015.

[3] Tianyi Qiu, Jingxuan Qiu, Jun Feng, Dingfeng Wu, Yiyan Yang, Kailin Tang, Zhiwei Cao, and Ruixin Zhu. The recent progress in proteochemometric modelling: focusing on target descriptors, cross-term descriptors and application scope. *Briefings in Bioinformatics*, 18(1):125–136, February 2016.

[4] C. Da and D. Kireev. Structural protein–ligand interaction fingerprints (SPLIF) for structure-based virtual screening: Method and benchmark study. *Journal of Chemical Information and Modeling*, 54(9):2555–2561, August 2014.

[5] Masashi Tsubaki, Kentaro Tomii, and Jun Sese. Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics*, 35(2):309–318, July 2018.

[6] Gerard JP van Westen, Remco F Swier, Isidro Cortes-Ciriano, Jörg K Wegner, John P Overington, Adriaan P IJzerman, Herman WT van Vlijmen, and Andreas Bender. Benchmarking of protein descriptor sets in proteochemometric modeling (part 2): modeling performance of 13 amino acid descriptor sets. *Journal of Cheminformatics*, 5(1), September 2013.

[7] S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, November 1992.

[8] Roshan Rao, Jason Liu, Robert Verkuil, Joshua Meier, John F. Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. MSA transformer. *bioRxiv*, February 2021.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[10] Ehsaneddin Asgari and Mohammad R. K. Mofrad. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLOS ONE*, 10(11):e0141287, November 2015.

[11] Ethan C. Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M. Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods*, 16(12):1315–1322, October 2019.

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[14] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, and C. H. Wu and. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, November 2014.

[15] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics.

[16] The UniProt Consortium. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research*, 47(D1):D506–D515, November 2018.

[17] CHEMBL database release 26. Technical report, March 2020.

[18] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E Bolton. PubChem in 2021: new data content and improved web interfaces. *Nucleic Acids Research*, 49(D1):D1388–D1395, November 2020.

[19] Benedek Fabian, Thomas Edlich, Héléna Gaspar, Marwin Segler, Joshua Meyers, Marco Fiscato, and Mohamed Ahmed. Molecular representation learning with language models and domain-relevant auxiliary tasks. *arXiv preprint arXiv:2011.13230*, 2020.

[20] Marwin HS Segler, Thierry Kogej, Christian Tyrchan, and Mark P Waller. Generating focussed molecule libraries for drug discovery with recurrent neural networks. *arXiv preprint arXiv:1701.01329*, 2017.

[21] Fabian Sievers and Desmond G. Higgins. Clustal omega for making accurate alignments of many protein sequences. *Protein Science*, 27(1):135–145, October 2017.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[23] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[24] Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, October 2017.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[26] Yungki Park and Edward M Marcotte. Flaws in evaluation schemes for pair-input computational predictions. *Nature Methods*, 9(12):1134–1136, December 2012.

[27] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[28] Nikolaus Stiefl, Ian A. Watson, Knut Baumann, and Andrea Zaliani. ErG: 2d pharmacophore descriptions for scaffold hopping. *Journal of Chemical Information and Modeling*, 46(1):208–220, January 2006.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.