Journal Title Here, 2024, pp. 1–8

doi: DOI HERE
Advance Access Publication Date: Day Month Year
Paper

PAPER

Effect of dataset partitioning strategies for evaluating out-of-distribution generalisation for predictive models in biochemistry

Raul Fernandez-Diaz,[1,2,3,*] Hoang Thanh Lam,[1] Vanessa Lopez[1] and Denis C. Shields [2,3,*]

[1]IBM Research, Dublin, IBM Technology Campus Damastown Industrial Park Mulhuddart, D15 HN66, Dublin, Ireland, [2]School of Medicine, University College Dublin, Belfield, D04 C1P1, Dublin, Ireland and [3]Conway Institute of Biomolecular and Biomedical Science, University College Dublin, Belfield, D04 C1P1, Dublin, Ireland

[*]Corresponding authors. raul.fernandezdiaz@ucdconnect.ie denis.shields@ucd.ie

FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

We have developed Hestia, a computational tool that provides a unified framework for introducing similarity correction techniques across different biochemical data types. We propose a new strategy for dividing a dataset into training and evaluation subsets (CCPart) and have compared it against other methods at different thresholds to explore the impact that these choices have on model generalisation evaluation, through the lens of overfitting diagnosis. We have trained molecular language models for protein sequences, DNA sequences, and small molecule string representations (SMILES) on the alternative splitting strategies for training and evaluation subsets. The effect of partitioning strategy and threshold depend both on the specific prediction task and the biochemical data type, for tasks for which homology is important, like enzymatic activity classification, being more sensitive to partitioning strategy than others, like subcellular localization. Overall, the best threshold for small molecules seems to lay between 0.4 and 0.5 in Tanimoto distance, for DNA between 0.4 and 0.5, and for proteins between 0.3 and 0.5, depending on the specific task. Similarity correction algorithms showed significantly better ability to diagnose overfitting in 11 out of 15 datasets with CCPart being more clearly dependent on the threshold than the alternative GraphPart, which showed more instability.

Availability and implementation

The source code is freely available at `https://github.com/IBM/Hestia`. The tool is also made available through a dedicated web-server at `http://peptide.ucd.ie/Hestia`.

Key words: Predictive modelling, Biochemistry, Machine learning, Trustworthy AI

Introduction

Predictive modelling in biochemistry has seen great advances in the last decade, powered by the rapid development of machine learning (ML) algorithms. These new models have achieved remarkable success in complex tasks like protein folding prediction [16] or quantum property prediction [36]. They are data-driven approaches that depend on the data distribution to which they have been fitted [1].

The ideal model would be fitted to a data distribution that is a close approximation of the underlying population [10]. In practice, there is typically a difference in model performance between the population distribution and the training distribution. This reflects that learning algorithms incorporate the biases present in the training data and will struggle to generalise to new data [27]. This problem, commonly referred to as overfitting [44], is exacerbated when models are too complex or data is scarce [10].

Diagnosing overfitting is, hence, crucial for evaluating model generalisation. The gold standard method would be to use one or multiple out-of-distribution evaluation datasets which have been drawn independently from the training distribution, i.e. by being collected by independent teams or stored in different repositories [47, 18]. Often, however, such independent dataset are not available. In such cases, the training distribution is partitioned into a training, a validation, and a hold-out evaluation set. The model is fitted to the training set, the validation set is used to determine the optimal model configuration, and the hold-out evaluation set serves as a simulation of an independent dataset, providing an estimation of future real-world performance [44]. One standard strategy used to create these dataset partitions is random sampling [1].

© The Author 2024. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

The implicit assumption of random partitioning is that all data points are independent and identically distributed, which may be a reasonable assumption for some machine learning tasks, but may not apply in many scientific domains.

Biochemistry is one such domain with two distinct characteristics. First, the population distribution is virtually infinite, whereas training distributions are generally small and significantly biased [23]. A common example is binding affinity databases being biased towards certain protein families and high affinity binders [7]. Secondly, individual data points are not independent from each other, as similar molecules share similar properties [44, 41] (e.g., two protein sequences with a sequence identity above 30% usually share a common ancestor and adopt similar structures [35]). The consequence of these characteristics of biochemical data is that the assumption of independence and identical distribution is not applicable. Therefore, random sampling is not an appropriate method for generating out-of-distribution evaluation sets.

In recent years, the practical implications of this problem [20] have led to the development of methods [40, 41, 20] for accounting for this, as well as community guidelines describing proper evaluation workflows [44, 4]. The inherent differences between chemical entities have led to tailored methodologies that share a common objective: to measure the similarity between data points and use it to guide the dataset partitioning process.

Similarity measurement methods for biological sequences (nucleic acids or proteins) often rely on pairwise alignments to obtain a quantitative metric, generally sequence identity, between aligned monomers (nucleotides or amino acids). Historically, these techniques have been developed as strategies for detecting homologous sequences, i.e., sequences related evolutionarily [14, 35]. This concept has greatly informed partitioning strategies within the field which can be divided into two main strategies: homology reduction and homology partitioning. Homology reduction is a two-step method that begins with clustering the sequences based on a predefined similarity threshold using algorithms like greedy incremental clustering (CD-HIT [21] or MMseqs [38]) or greedy linear cover-set clustering (MMseqs [38]) followed by a redundancy reduction step where only the representative member of each cluster is kept. Its main drawback is that it can lead to substantial information loss. Homology partitioning, on the other hand, is a newly proposed method [40] that minimises information loss. It starts with a clustering step followed by an iterative process through which the clusters are merged into partitions and the sequences with most interpartition neighbours are removed progressively until there are no interpartition neighbours.

Similarity metrics for small molecules are much more diverse and have historically been developed for querying existing databases for related compounds [33]. The most accepted methods rely on set comparison metrics like the Tanimoto distance (binary Jacquard similarity) [2] between hash-based substructure fingerprints (e.g., ECFP [33]). Regarding partitioning methods, the theoretical foundation is not as developed as is for biosequences and they may not directly rely on a pairwise similarity metric [41, 5]. The most common approaches are scaffold partition [41] and perimeter split [41, 5]. In scaffold partition, the Bemis-Murcko scaffold [3] is calculated for each molecule and those molecules sharing identical scaffolds are clustered together. In perimeter split, also known as extrapolation-oriented split, the dataset is clustered, typically through k-means clustering on a set of precomputed physico-chemical descriptors and the clusters with maximal distance between their representatives are assigned to different partitions [39, 41].

This study introduces four main contributions:

1. We propose a unified framework that is generalised to any kind of biochemical entity for which a pairwise similarity can be computed for the systematic study of similarity relationships between partitions of any dataset and for the introduction of varied similarity correction techniques. These techniques include: **similarity reduction**, which inherits from the homology reduction concept and entails the clustering of entities according to a predefined similarity threshold followed by redundancy reduction of all cluster members but the representatives; **similarity partitioning**, which inherits from the homology partitioning concept and entails the partitioning of a dataset into subsets such that there are no entities with a similarity above a predefined threshold in different partitions through the removal of a minimal number of entities.
2. We generalise the GraphPart similarity partitioning algorithm to any arbitrary similarity metric and biochemical species.
3. We present a novel algorithm for similarity partitioning leveraging the connected components graph partitioning (CCPart) algorithm and we benchmark its effect against random partitioning and the generalised version of GraphPart in a collection of real-world machine learning datasets both from the perspective of the distribution of similar entities across partitions and from the practical impact on apparent model performance when evaluating predictive methods.
4. We release Hestia, an open-source program that integrates all the aforementioned contributions, as both a command-line application and a Python API (`https://github.com/IBM/Hestia`). Moreover, we have made available a dedicated homonym web-server (`http://peptide.ucd.ie/Hestia`) to facilitate its accessibility.

Hestia suite

Similarity calculation

Hestia implements a diverse array of pairwise similarity functions $f(x, y)$ which need to fulfill three conditions:

1. They are normalised, $f : A \rightarrow [0, 1]$.
2. The similarity between an entity and itself is maximal, $f(x, x) = 1$.
3. They are symmetric, $f(x, y) = f(y, x)$.

Biological sequences. The similarity metric calculated for protein sequences and nucleic acids is the sequence identity in pairwise alignments. Local alignments are calculated using the MMSeqs2 implementation [38] of the Waterman-Smith algorithm [37] with or without prior k-mer prefiltering. Global alignments are calculated using the EMBOSS implementation [32] of the Needleman-Wunch algorithm [25]. More information regarding the empirical differences between local and global alignments can be found in [30, 40]. The denominator used to calculate the identity can be the length of the longest or the shortest sequence, as well as the length of the full alignment. Choice of the most appropriate denominator depends on the dataset [24].

Protein structures. The similarity metric calculated for a pair of protein structures is the probability that they belong to the same SCOPe family [6]. This probability is approximated using the Foldseek alignment algorithm [42] with both sequence and structural interaction representation (3Di+AA) in either global (Foldseek-TM) or local (Foldseek) modes.

For both biological sequences and protein structures, pairwise alignments are not necessarily symmetric so we enforce condition 3 by taking the maximal similarity for each pair of entities: $f^* := max[f(x,y), f(y,x)]$.

Small molecules. Similarity between small molecules is calculated using the Tanimoto similarity [2] between extended connectivity fingerprints (ECFP) [33]. Similarity between Bemis-Murcko scaffolds [3] is defined as a boolean function returning 1 if the scaffold is the same and 0 if it is not. All calculations are performed through the `RDKit` [19] implementation of the algorithms.

Similarity correction algorithms

Basic notation. Let D be an arbitrary dataset comprised of n entities. D can be expressed as a graph $G(N, E)$ where the nodes (N) are the set of all entities in D and the edges (E), the subset of all pairwise similarity measurements between those entities $f(N_i, N_j)$ with values above a threshold λ, thus, $E = \{(n_1, n_2)\ \forall\, n_1, n_2 \in N$ such that $f(n_1, n_2) > \lambda\}$. Let T and V be two partitions of D such that $T \cap V = \emptyset$. Then, E_f, the forbidden edges, can be defined as the subset of all similarities between any two entities in T and V that is above the threshold, $E_f = \{(t, v)\ \forall\, t \in T, v \in V$ such that $f(t, v) > \lambda\}$.

In this paper, similarity correction techniques refer to algorithms targeting the reduction of E_f. We discuss them extensively as outlined below.

Similarity reduction. Similarity reduction aims at reducing redundant entities from D. This is achieved by a two-step process comprising a clustering step and a redundancy reduction step in which the representative entities of each cluster are selected and the rest of cluster members are removed. Hestia relies on custom implementations of greedy incremental clustering and greedy linear cover-set algorithms for the clustering step. These algorithms are commonly used in the context of sequence clustering, by specialised software like CD-HIT (greedy incremental clustering) [12] and MMSeqs (both) [38]. Our custom implementations generalise their utility to any arbitrary data type for which a pairwise distance matrix can be calculated.

Random partitioning. Random partitioning algorithms aim to divide the dataset into subsets through unbiased sampling. The idea is to generate partitions with similar distributions. Strictly speaking, it could only be considered a similarity correction algorithm under the assumption of independent and identically distributed data, e.g., after performing similarity reduction on the dataset. Hestia leverages the corresponding `scikit-learn` implementation [17] of the algorithm.

GraphPart generalisation. Similarity partitioning algorithms aim at dividing the dataset into n subsets or partitions such that $E_f = \emptyset$ between any two partitions. This is achieved through the removal of entities whenever necessary. Hestia relies on a custom implementation of the Graph-Part algorithm [40] that generalises it to any similarity metric and biochemical data type.

The algorithm starts with a clustering step using limited agglomerative clustering with single linkage with the restriction that the clustering stops when either a) a cluster reaches the expected partition size N/n where N is the number of entities in the dataset and n the number of desired partitions or b) there are no more edges above the threshold λ. Then, clusters are iteratively merged into the n partitions so that the generated partitions have a balanced distribution of labels (if the dataset has categorical labels) and similar number of entities. The number of interpartition neighbours (entities with $E > \lambda$) is checked and entities are moved to the partition with which they have the most neighbours. If the number of entities with at least one interpartition neighbour is greater than 0, then the $c \times \log(i/100) + 1$ entities with most interpartition neighbours are removed, where c is the number of interpartition neighbours and i, the current iteration. This iterative process continues until $c = 0$. Our custom implementation leverages `Scipy` [43] and `Networkx` [13].

Connected components partitioning. Herein, we propose and implement connected components partitioning (CCPart) as an algorithm for minimising the similarity between the entities assigned to the training and evaluation subsets.

The algorithm first identifies the set of all unconnected subgraphs of G, $\{U_1, U_2, \ldots, U_k\}$. These unconnected subgraphs, by definition, will not have any intercluster neighbours, otherwise they would not be unconnected. We have observed that generally this strategy leads to one cluster being populated with most entities and a variable number of much smaller clusters (10 - 10^3). The smaller a cluster is, the more unique it is with respect to the dataset distribution. Based on this assumption, the algorithm builds the evaluation set by assigning clusters in ascending order of number of members, i.e., smaller clusters first. As above, there is also the optional additional objective of maximising evaluation label balance. This sampling strategy biases the evaluation subset towards the regions of the dataset distribution that are the most unconnected and thus the most dissimilar to other members of the dataset.

Analysis of partition independence

Dependence between the training and evaluation partitions is defined as the proportion of elements in the training partition for which there exists at least one similar entity in the evaluation set. Two entities is considered to be similar if the distance between them is above a threshold, λ.

Materials and methods

Data collection

Five **protein datasets** were collected from protein representation learning benchmark datasets [31, 46, 8], for the prediction of: stability, enzyme class (according to enzyme commission classification), solubility, optimal temperature for catalysis, and subcellular localization.

Five **nucleic acids (DNA) datasets** were collected from DNA representation learning benchmark datasets from [48]: core promoter regions, larger promoter regions, epigenetic markers on H3, epigenetic markers on H4, and mouse transcription factor-binding regions.

Five **small molecule datasets** were collected from the Therapeutics Data Commons collection [15]: cytochrome P450 2C19 inhibition, cell-effective permeability in Caco-2 colon epithelial cancer cell line, P-glycoprotein inhibition, mutagenicity (as defined by the Ames test), and lipophilicity.

Similarity calculations

For protein and nucleic acids datasets, the similarity metrics used were sequence identity in local sequence alignment (Smith-Waterman [37]) with prior k-mer prefiltering [38]. The denominator for calculating the sequence identity was the shortest sequence [24]. For small molecules, the similarity metrics used were Tanimoto similarity [2] between ECFP-4 fingerprints with 1,024 bits [33].

Dataset partitions

For every dataset 3 different partition strategies were used to generate the training and evaluation sets: i) random partitioning, ii) similarity partitioning with our custom implementation of Graph-Part (see 2.2), and iii) CCPart. Overfitting is defined as a quotient between the model performance against the validation (in-distribution) and against the evaluation set (potentially out-of-distribution).

The threshold used for the similarity correction algorithms was 0.3 for protein sequences, and 0.4 for nucleic acids [35, 44]. Following [20], the threshold for Tanimoto similarity was set to 0.4.

Machine learning modelling

Effect of the partitioning strategy is examined by the use case of fine-tuning molecular language models (MLMs) to the training partitions using a single linear layer as either classification or regression head. The MLMs used were ESM-2 8M [22] for protein datasets, multi-species Nucleotide Transformer v-2 250M [9] for DNA, and MolFormer-XL [34] for small molecules. Fine-tuning is performed for all model weights, with Adam optimizer and a learning rate of 10^{-4} for a maximum of 50 epochs with early-stopping with a patience of 5 epochs and tolerance of 10^{-3} on the validation set using the appropriate dependencies of `PyTorch` [29] and `Transformers` [45]. To account for the stochasticity inherent to the optimisation algorithms, every fine-tuning experiment is repeated three times.

Model performance is measured through correlation metrics for small molecules and nucleic acid datatsets: Matthew's correlation coefficient for classification tasks and Spearman's ρ coefficient for regression tasks. For protein datasets, where tasks include multilabel classification, weighted F1 was used instead.

Results and discussion

DNA

The effect of similarity correction techniques are represented in for the DNA datasets in Figure 1 (A-E). The results show that 1) GraphPart is roughly equivalent to random partitioning and 2) CCPart has a significant impact for three of the five datasets at the lowest threshold (0.4). For the remaining two datasets, the functionality of promoter (D) and transcription-factor-binding (E) sequences is determined by small motifs [28, 26] that are not captured by the alignment analysis.

Proteins

The results for the protein datasets are represented in Figure 1 (F-J) and show that CCPart is significantly better than both random partitioning and GraphPart at low thresholds (0.3 - 0 .4) for three out of five datasets, i.e., (F) enzyme class, (G) stability, and (H) subcellular localization). The dependence on the threshold is tied to the nature of the biological task, with enzyme class being the dataset most affected. This was to be expected as homology is important for predicting enzymatic activity whereas subcellular localisation does not depend so heavily on extended homology but on smaller subsequences [11] that are not well captured by the alignment algorithms.

Solubility (I) and optimal temperature (J), similarly, are governed by intermolecular or intramolecular interactions, respectively, which are typically dependent on a limited number of residues and mutations would also not be not well captured by alignment methods.

Small molecules

Finally, Figure 1 (K-O) shows the results for the small molecule datasets: 1) CCPart offers greater overfitting diagnosis than random partitioning at low thresholds (0.4 - 0.5), and 2) no *a priori* decision can be made about which algorithm will generate the most challenging overfitting-diagnosing dataset, however, 3) CCPart offers a clearer dependence on the threshold than GraphPart suggesting that the algorithm behaves more predictably.

Conclusion

Hestia is a computational tool that embodies a new unified framework to study the similarity between partitions of biochemical datasets and implement correction techniques to generate better generalisation-evaluating hold-out subsets.

The results for DNA, proteins, and small molecules show that there is no one-size-fits-all approach that can guarantee proper evaluation of generalisation. Furthermore, the choice of similarity partitioning algorithm seems to be task dependent, though CCPart demonstrated a clearer dependence on threshold suggesting it is a more robust algorithm overall for creating overfitting-evaluating datasets.

Regarding the dependence of the algorithms in the biological task, this could be mitigated by the choice of similarity metrics that capture the underlying biology of the predictive task.

In any case, the results are an important demonstration of the importance of considering both the biological implications of the task under study and the need for more comprehensive techniques for evaluating model generalisation to unseen data.

Competing interests

No competing interest is declared.

References

1. Solveig Badillo, Balazs Banfai, Fabian Birzele, Iakov I Davydov, Lucy Hutchinson, Tony Kam-Thong, Juliane Siebourg-Polster, Bernhard Steiert, and Jitao David Zhang. An introduction to machine learning. *Clinical pharmacology & therapeutics*, 107(4):871–885, 2020.
2. Dávid Bajusz, Anita Rácz, and Károly Héberger. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of cheminformatics*, 7(1):1–13, 2015.
3. Guy W Bemis and Mark A Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.
4. Andreas Bender, Nadine Schneider, Marwin Segler, W Patrick Walters, Ola Engkvist, and Tiago Rodrigues.

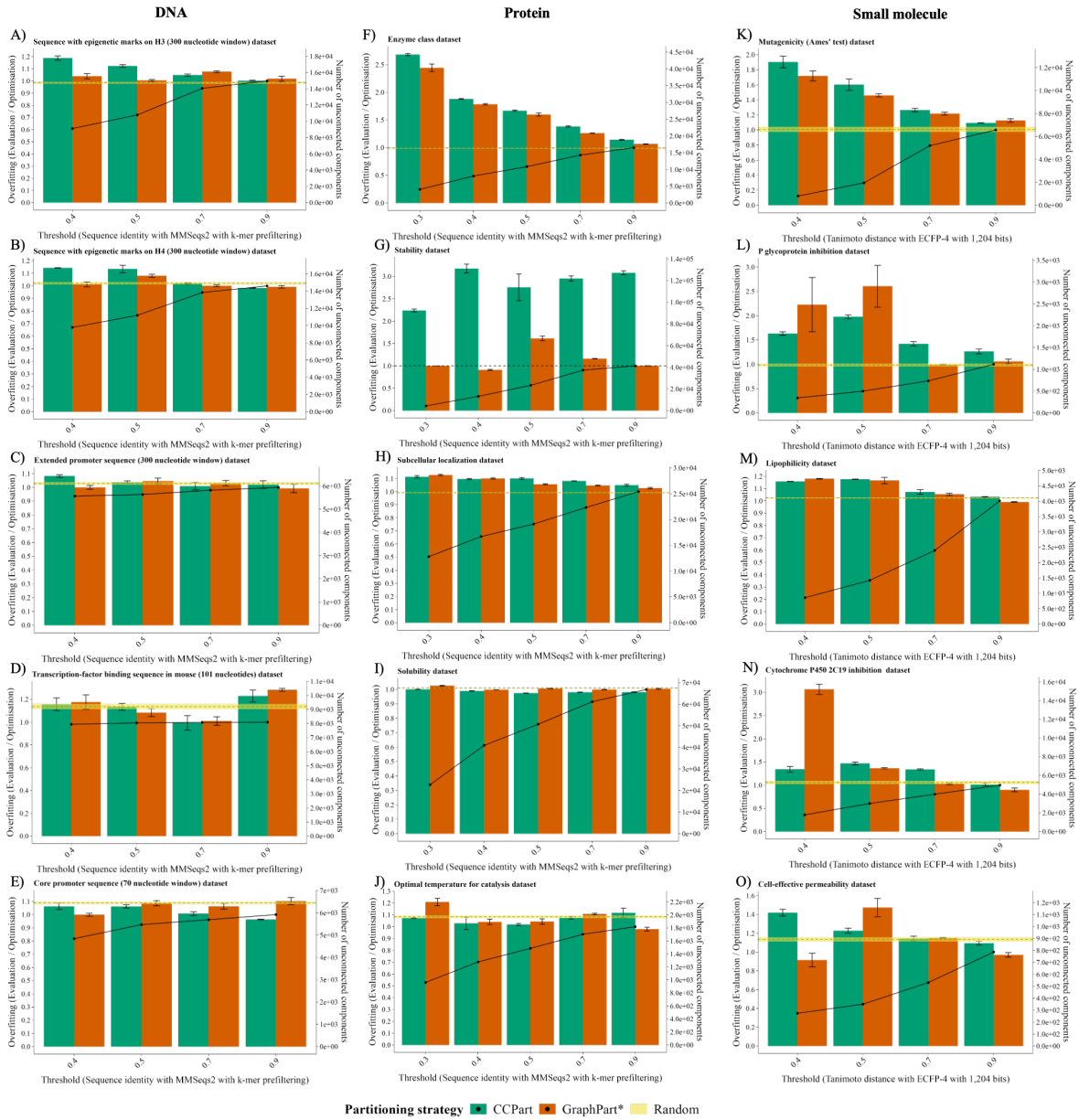

Fig. 1. Comparison of different algorithms at increasing similarity thresholds. Yellow line represents the performance of random partitioning and acts as a baseline, its width represent the error of the mean across four replicates. The error bars represent the standard error of the mean of three replicates.

Evaluation guidelines for machine learning tools in the chemical sciences. *Nature Reviews Chemistry*, 6(6):428–442, 2022.

5. Jackson W Burns, Kevin A Spiekermann, Himaghna Bhattacharjee, Dionisios G Vlachos, and William H Green. Machine learning validation via rational dataset sampling with astartes. *Journal of Open Source Software*, 8(91):5996, 2023.
6. John-Marc Chandonia, Naomi K Fox, and Steven E Brenner. Scope: classification of large macromolecular structures in the structural classification of proteins—extended database. *Nucleic acids research*, 47(D1):D475–D481, 2019.
7. Ayan Chatterjee, Robin Walters, Zohair Shafi, Omair Shafi Ahmed, Michael Sebek, Deisy Gysi, Rose Yu, Tina Eliassi-Rad, Albert-László Barabási, and Giulia Menichetti. Ai-bind: Improving binding predictions for novel protein targets and ligands. *arXiv preprint arXiv:2112.13168*, 2021.
8. Bo Chen, Xingyi Cheng, Yangli-ao Geng, Shen Li, Xin Zeng, Boyan Wang, Jing Gong, Chiming Liu, Aohan Zeng, Yuxiao Dong, et al. xtrimopglm: unified 100b-scale pre-trained transformer for deciphering the language of protein. *Biorxiv*, pages 2023–07, 2023.
9. Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P

de Almeida, Hassan Sirelkhatim, et al. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *bioRxiv*, pages 2023–01, 2023.

10. Yehuda Dar, Vidya Muthukumar, and Richard G Baraniuk. A farewell to the bias-variance tradeoff? an overview of the theory of overparameterized machine learning. *arXiv preprint arXiv:2109.02355*, 2021.
11. Pierre Dönnes and Annette Höglund. Predicting protein subcellular localization: past, present, and future. *Genomics, proteomics & bioinformatics*, 2(4):209–215, 2004.
12. Limin Fu, Beifang Niu, Zhengwei Zhu, Sitao Wu, and Weizhong Li. Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 28(23):3150–3152, 2012.
13. Aric Hagberg and Drew Conway. Networkx: Network analysis with python. *URL: https://networkx. github. io*, 2020.
14. Desmond G Higgins and Paul M Sharp. Clustal: a package for performing multiple sequence alignment on a microcomputer. *Gene*, 73(1):237–244, 1988.
15. Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *arXiv preprint arXiv:2102.09548*, 2021.
16. John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Kathryn Tunyasuvunakool, Olaf Ronneberger, Russ Bates, Augustin Žídek, Alex Bridgland, et al. Alphafold 2. *Fourteenth Critical Assessment of Techniques for Protein Structure Prediction; DeepMind: London, UK*, 2020.
17. Oliver Kramer and Oliver Kramer. Scikit-learn. *Machine learning for evolution strategies*, pages 45–53, 2016.
18. Andriy Kryshtafovych, Torsten Schwede, Maya Topf, Krzysztof Fidelis, and John Moult. Critical assessment of methods of protein structure prediction (casp)—round xv. *Proteins: Structure, Function, and Bioinformatics*, 2023.
19. Greg Landrum. Rdkit documentation. *Release*, 1(1-79):4, 2013.
20. Jie Li, Xingyi Guan, Oufan Zhang, Kunyang Sun, Yingze Wang, Dorian Bagni, and Teresa Head-Gordon. Leak proof pdbbind: A reorganized dataset of protein-ligand complexes for more generalizable binding affinity prediction. *ArXiv*, 2023.
21. Weizhong Li and Adam Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, 2006.
22. Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.
23. Yang Liu and Hisashi Kashima. Chemical property prediction under experimental biases. *Scientific Reports*, 12(1):8206, 2022.
24. Alex CW May. Percent sequence identity: the need to be explicit. *Structure*, 12(5):737–738, 2004.
25. Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.
26. Mhaned Oubounyt, Zakaria Louadi, Hilal Tayara, and Kil To Chong. Deepromoter: robust promoter predictor using deep learning. *Frontiers in genetics*, 10:286, 2019.
27. Chunjong Park, Anas Awadalla, Tadayoshi Kohno, and Shwetak Patel. Reliable and trustworthy machine learning for health using dataset shift detection. *Advances in Neural Information Processing Systems*, 34:3043–3056, 2021.
28. Sungjoon Park, Yookyung Koh, Hwisang Jeon, Hyunjae Kim, Yoonsun Yeo, and Jaewoo Kang. Enhancing the interpretability of transcription factor binding site prediction using attention mechanism. *Scientific reports*, 10(1):13413, 2020.
29. 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 *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
30. Valery O Polyanovsky, Mikhail A Roytberg, and Vladimir G Tumanyan. Comparative analysis of the quality of a global algorithm and a local algorithm for alignment of two sequences. *Algorithms for molecular biology*, 6(1):1–12, 2011.
31. Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019.
32. Peter Rice, Ian Longden, and Alan Bleasby. Emboss: the european molecular biology open software suite. *Trends in genetics*, 16(6):276–277, 2000.
33. David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.
34. Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4(12):1256–1264, 2022.
35. Burkhard Rost. Twilight zone of protein sequence alignments. *Protein engineering*, 12(2):85–94, 1999.
36. Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24), 2018.
37. Temple F Smith, Michael S Waterman, et al. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.
38. Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.
39. Csaba Szántai-Kis, István Kövesdi, György Kéri, and László Örfi. Validation subset selections for extrapolation oriented qspar models. *Molecular diversity*, 7:37–43, 2003.
40. Felix Teufel, Magnus Hallder Gislason, José Juan Almagro Armenteros, Alexander Rosenberg Johansen, Ole Winther, and Henrik Nielsen. Graphpart: Homology partitioning for biological sequence analysis. *bioRxiv*, pages 2023–04, 2023.

41. Prudencio Tossou, Cas Wognum, Michael Craig, Hadrien Mary, and Emmanuel Noutahi. Real-world molecular out-of-distribution: Specification and investigation. 2023.
42. Michel van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Foldseek: fast and accurate protein structure search. *Biorxiv*, pages 2022–02, 2022.
43. Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.
44. Ian Walsh, Dmytro Fishman, Dario Garcia-Gasulla, Tiina Titma, Gianluca Pollastri, Jennifer Harrow, Fotis E Psomopoulos, and Silvio CE Tosatto. Dome: recommendations for supervised machine learning validation in biology. *Nature methods*, 18(10):1122–1127, 2021.
45. Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.
46. Zuobai Zhang, Minghao Xu, Vijil Chenthamarakshan, Aurélie Lozano, Payel Das, and Jian Tang. Enhancing protein language models with structure-based encoder and pre-training. *arXiv preprint arXiv:2303.06275*, 2023.
47. Naihui Zhou, Yuxiang Jiang, Timothy R Bergquist, Alexandra J Lee, Balint Z Kacsoh, Alex W Crocker, Kimberley A Lewis, George Georghiou, Huy N Nguyen, Md Nafiz Hamid, et al. The cafa challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome biology*, 20(1):1–23, 2019.
48. Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. Dnabert-2: Efficient foundation model and benchmark for multi-species genome. *arXiv preprint arXiv:2306.15006*, 2023.

Performance of the models

Figure 1 shows the absolute performance of the models against the corresponding evaluation sets.

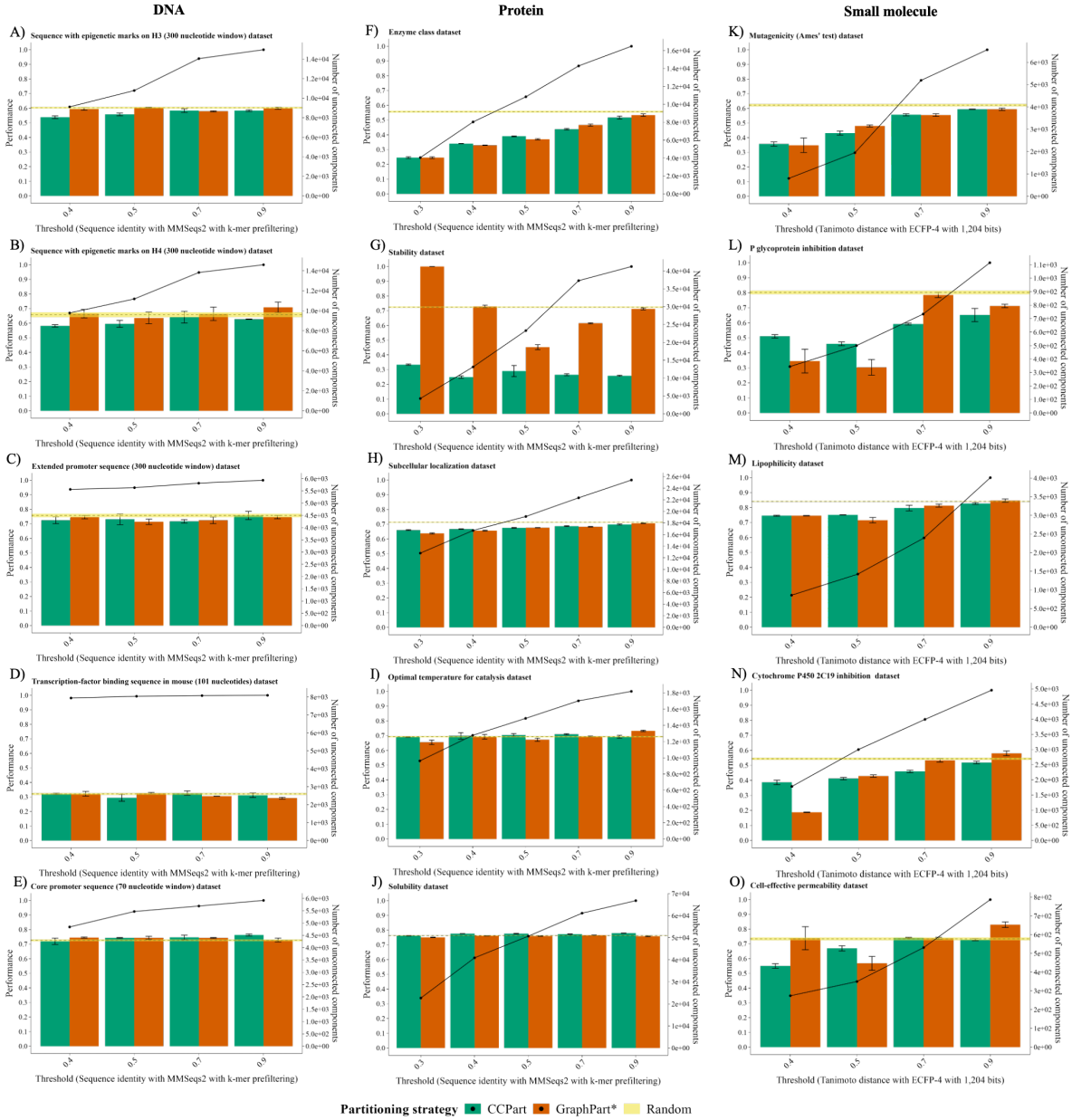

Fig. 1. Absolute performance of the models. Performance corresponds to Matthew's correlation coefficient for classification tasks for DNA and small molecules, Pearson's ρ coefficient for regression tasks for DNA, protein, and small molecule tasks, and weighted F1 for protein classification tasks. Error bars correspond to the standard error of the mean of three replicates.