A NEW FRAMEWORK FOR EVALUATING MODEL OUT-OF-DISTRIBUTION FOR THE BIOCHEMICAL DOMAIN *

Raúl Fernández-Díaz
IBM Research
UCD School of Medicine
UCD Conway Institute of Biomolecular and Biomedical Research
SFI Center for Research Training Genomics Data Science
Dublin, Ireland
raul.fernandezdiaz@ucdconnect.ie

Thanh Lam Hoang and Vanessa Lopez
IBM Research
Dublin, Ireland
{t.l.hoang, vanlopez}@ie.ibm.com

Denis C. Shields
UCD School of Medicine
UCD Conway Institute of Biomolecular and Biomedical Research
denis.shields@ucd.ie

ABSTRACT

Quantifying model generalization to out-of-distribution data has been a longstanding challenge in machine learning. Addressing this issue is crucial for leveraging machine learning in scientific discovery, where models must generalize to new molecules or materials. Current methods typically split data into train and test sets using various criteria — temporal, sequence identity, scaffold, or random cross-validation — before evaluating model performance. However, with so many splitting criteria available, existing approaches offer limited guidance on selecting the most appropriate one, and they do not provide mechanisms for incorporating prior knowledge about the target deployment distribution(s).

To tackle this problem, we have developed a novel metric, AU-GOOD, which quantifies expected model performance under conditions of increasing dissimilarity between train and test sets, while also accounting for prior knowledge about the target deployment distribution(s), when available. This metric is broadly applicable to biochemical entities, including proteins, small molecules, nucleic acids, or cells; as long as a relevant similarity function is defined for them. Recognizing the wide range of similarity functions used in biochemistry, we propose criteria to guide the selection of the most appropriate metric for partitioning. We also introduce a new partitioning algorithm that generates more challenging test sets, and we propose statistical methods for comparing models based on AU-GOOD.

Finally, we demonstrate the insights that can be gained from this framework by applying it to two different use cases: developing predictors for pharmaceutical properties of small molecules, and using protein language models as embeddings to build biophysical property predictors.

***K*eywords** First keyword · Second keyword · More

1 Introduction

The last decade has been characterised by the impact that the introduction of machine learning models has had in the acceleration of scientific discovery. These models are frequently used to predict the properties of entities (drug candidates, materials, cells, etc.) that are inherently different from those present in their training distribution. This deployment scenario, known as out-of-distribution (OOD), is particularly frequent within the biochemical domain

* *Citation*: **Authors. Title. Pages.... DOI:000000/11111.**

which encompasses both biological and chemical modelling. Proper OOD evaluation of models is necessary, within this domain specifically, due to the enormous economic and societal impact that wrong predictions might have, for example, when a drug candidate that was predicted as non-toxic [1] goes through to the latter phases of the drug development pipeline, including pre-clinical or clinical trials. Further tasks where OOD evaluation is critical are: modelling of the interaction between a known molecular target and new compounds [2], the prediction of the structure of molecular targets mediating disease [3], or cell type annotation [4]. Overall, robust OOD evaluation is necessary for the development of trustworthy predictive models, driving advancements in biochemistry.

Prior literature, in the biochemistry domain specifically, has already highlighted the importance of evaluating model generalisation to OOD data, though there is no prior work attempting to build a framework for both measuring the generalisation capabilities of any given model and to provide a statistical method to compare the performance of different models. Instead, prior works have focused on developing algorithms i) for measuring pairwise similarity values between biochemical entities [5, 6, 7], ii) for splitting a dataset into independent train-test subsets [8, 9, 10], and iii) for combining different partitioning algorithms to simulate the composition of a single target deployment distribution [11]. Due to the absence of a comprehensive framework for evaluating model generalisation to out-of-distribution (OOD) data, validating claims about machine learning generalisation remains challenging. Recent findings from a drug discovery competition indicate that out of 2,000 submitted solutions, none successfully predicted the binding affinity of unseen drugs to known proteins, despite training on the largest labeled affinity dataset, which contains over 95 million affinity pairs[2].

In response to this gap, we first present a framework to study and quantify model generalisation to OOD data for biochemistry. Unlike existing algorithms [8, 11, 10], we propose a novel dataset partitioning method that is broadly applicable across various biochemistry contexts, including proteins and small molecules. Our approach is agnostic to the underlying data types; instead, it relies on defined similarity functions between any two data instances. When there are various similarity functions of interest, we present a set of criteria to identify the best similarity function for defining out-of-distribution generalisation with minimal reliance on domain knowledge. Additionally, we propose a statistical metric to compare the generalisation capabilities of different models.

2 Mathematical framework and computational methodology

This section provides formal definition of out-of-distribution generalisation in biochemistry, the main methodological contributions of this study. It is divided into four subsections.

1. To facilitate the evaluation of out-of-distribution generalisation, we begin by defining the model's empirical risk based on the partitioning strategy used to create the train and test subsets. A further simplification focuses on the scenario where the partitioning strategy depends on the similarity between the train and test splits.
2. We use this mathematical formulation to derive a new generalisation metric, which corresponds to the expected empirical risk of a model trained on a given dataset with varying train/test similarity.
3. We propose an algorithm for dataset partitioning that biases the testing subset towards out-of-distribution points within the dataset without removing any data points.
4. We propose a statistical test to compare the generalisation capabilities of two different models.

2.1 Empirical risk in terms of the dataset partitioning strategy

This subsection demonstrates that the empirical risk can be expressed in terms of the partitioning operation used to divide the data into training and testing subsets.

Let us consider a model f_θ defined by a set of parameters θ and trained on a data distribution $P(x, y)$ where each datum is defined by a set of features $x \in \mathcal{X}$ and a set of labels $y \in \mathcal{Y}$. The model will attempt to approximate an unknown function g that provides a mapping from the feature space to the label space $f_\theta \approx g\colon \mathcal{X} \to \mathcal{Y}$. The quality of the approximation can be described in terms of a loss function $\mathcal{L}$ that measures the error in each individual mapping. In the end, the parameters of the model will be optimised to minimise the expectation of the loss across the whole population which is also known as the population risk:

$$\mathcal{R}(f_\theta) \coloneqq \mathbb{E}_{(x,y)\sim P(x,y)}[\mathcal{L}(f_\theta(x), y)] \tag{1}$$

[2]https://leashbio.substack.com/p/belka-results-suggest-computers-can

In practice, we do not have access to the data distribution, only to a subset, $\mathcal{D} \sim P(x, y)$. Without access to the data distribution, the population risk has to be approximated by the empirical risk. The empirical risk of a model given a dataset $\mathcal{D}$ is calculated by partitioning $\mathcal{D}$ into two mutually exclusive subsets $\mathcal{T}, \mathcal{E} \in \mathcal{D}$ such that $\mathcal{T} \cap \mathcal{E} = \emptyset$, where $\mathcal{T}$ is the train subset and $\mathcal{E}$ is the test subset defined by the partitioning operation denoted as $\Phi\colon \mathcal{D} \to \mathcal{T}, \mathcal{E}$. The model trained on the train subset is denoted as $f_{\theta,\mathcal{T}}(x)$ Therefore, the empirical risk can be estimated in terms of the partitioning strategy:

$$\hat{\mathcal{R}}_{\Phi(\mathcal{D})}(f_\theta) \coloneqq \frac{1}{n} \sum_{i \in \mathcal{E}} \mathcal{L}(f_{\theta,\mathcal{T}}(x_i), y_i) \tag{2}$$

where n is the number of elements in $\mathcal{E}$.

2.2 Definition of a metric estimating model generalisation to a target distribution

This subsection builds upon the previous definition of empirical risk in terms of the partitioning operator and derives a metric for model generalisation to any target data distribution based on the similarity between the elements of the training and target distributions.

Let us consider a hypothetical function that measures the similarity between the features of any two elements in the population, $s\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Let us also consider a partitioning operator that enforces a maximal value of similarity λ between the elements in training and testing. Such an operator would necessarily be a function of both the similarity function s and the threshold similarity λ, $\Phi_{\lambda|s}(D)$. For conciseness, this expression will be simplified to λ_s when Φ is clear from the context. Then, the empirical risk would also depend on the similarity function and the similarity threshold:

$$\hat{R}_{\lambda_s}(f_\theta) \coloneqq \frac{1}{n} \sum_{i \in \mathcal{E}} \mathcal{L}(f_{\theta,\mathcal{T}}(x_i), y_i) \tag{3}$$

Model generalisation to a target deployment distribution $\mathcal{E}^*$ that has been drawn from a population distribution $\mathcal{E}^* \sim P^*(x, y)$ different from $P(x, y)$ could then be described as the expectation of the empirical risk across the distribution of the similarity λ_s, denoted as $P(\lambda_s|P^*)$ induced from the unknown target distribution $P^*(x, y)$, within the bounds of a minimal similarity λ_0 and a maximal similarity λ_n:

$$\mathcal{G}_\Phi(f_\theta|P^*) \coloneqq \mathbb{E}_{[\lambda_s|P^*]} \hat{R}_{\lambda_s}(f_\theta) = \int_{\lambda_0}^{\lambda_n} \hat{R}_{\lambda_s} P(\lambda_s|P^*) d\lambda_s \tag{4}$$

where $P(\lambda_s|P^*)$ is the probability density distribution of the similarity between $\mathcal{D}$ and $\mathcal{E}^*$, i.e., the expected distribution of similarities between $\mathcal{D}$ and $\mathcal{E}^*$: $P(\lambda_s|P^*) = P(\max\{s(x_i, x_j)\ \forall\ x_i, x_j \in \mathcal{D}, \mathcal{E}^*\} = \lambda_s)$. This integral can be approximated numerically by:

$$\mathcal{G}_\Phi(f_\theta|P^*) \approx \sum_{i=\lambda_0}^{\lambda_n} \hat{R}_{\lambda_s^i} P(\lambda_s^i|P^*) \Delta\lambda_s \tag{5}$$

We refer to the curve describing model performance as a function of the maximum similarity between training and testing partitions as the Generalisation to Out-Of-Distribution curve, or GOOD curve. This curve offers a clear overview of the expected performance of the models at different levels of similarity. On the other hand, we refer to the metric defined in Eq. 5 as the Area under the GOOD curve, abbreviated to AU-GOOD. By definition, the value of AU-GOOD will depend on the target distribution being considered.

Let us consider a simple simulation, which will serve to illustrate the relationship between GOOD curve, target distribution(s) and AU-GOOD. Let us consider two models, "Model A" and "Model B", with GOOD curves as shown in Figure 1.A. It is clear from the Figure, that Model A performs better when exposed to entities more similar to its training data, indicating a degree of overfitting. In contrast, Model B exhibits less variance with respect to similarity, suggesting better generalisation.

Let us now consider two different target deployment distributions, "Target 1" and "Target 2". The histograms reflecting $P(\lambda_s|P^*)$ are shown in Figures 1.B-C. Target 1 (Figure 1.B) comprises entities with low similarity to the training data,

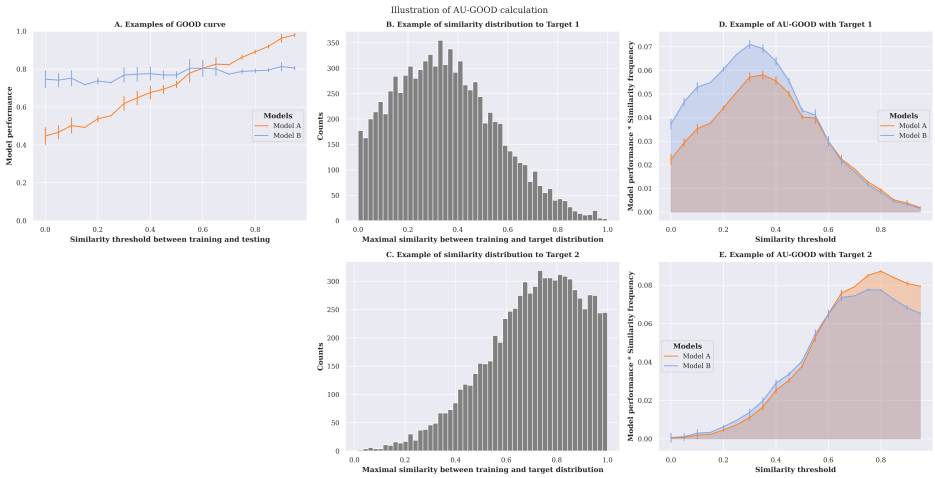

Figure 1: Illustration of the calculation of the AU-GOOD metric. A-C correspond to synthetic data.

while Target 2 (Figure 1.C) represents the opposite case, where most entities are similar to those the model has been trained on.

Finally, the geometrical representation of the AU-GOOD values for both models evaluated against each target distribution are presented in Figures 1.D-E. These visualizations clearly indicate that Model B is the optimal choice for Target 1, as it minimizes the risk[3] within the application domain defined by Target 1. In contrast, Model A is the superior option for Target 2.

This simple example also highlights one of the main strengths of using Eq. 5, which is that it allows to obtain estimators of model generalisation for an arbitrary number of target deployment distributions without the need for additional experiments. This is a significant advantage when compared with previous methods for estimating model performance to out-of-distribution datasets, such as the method proposed by [11] which modifies the partitioning operator to generate testing subsets that better approximate the composition of the target deployment distribution. This requires repeating training/evaluation cycles for each new target deployment distribution the model has to be evaluated against. Instead, Eq 5 only requires modifying the $P(\lambda_s^i|P^*)$ term, avoiding the repetition of the training/testing cycle.

2.3 Dataset partitioning to simulate out-of-distribution data given a similarity value

This subsection formally defines the concept of similarity function and introduces the new disimilarity-based partitioning algorithm we propose.

Consider a function $s : \mathcal{X} \times \mathcal{X} \rightarrow [0, 1]$ that measures the similarity between two feature vectors $x_1, x_2 \in X$. In this study, we consider any s function such that it fulfills two conditions:

1. The similarity between an entity and itself is maximal, $s(x_1, x_1) = 1$.
2. It is symmetric, $s(x_1, x_2) = s(x_2, x_1)$.

This function can be used to calculate all pairwise similarities between any two entities in the dataset, generating a similarity matrix, $S_{i,j} = s(x_i, x_j) \; \forall \; i, j \in \mathcal{X}$. We can then define $G_{\lambda_s}(\mathcal{X}, S^{\lambda_s})$ as a graph with $\mathcal{X}$ nodes and adjacency matrix, $S_{i,j}^{\lambda_s} = 1$ if $S_{i,j} \geq \lambda_s$ or 0 otherwise. Let us also consider the sampling strategy $\Phi_{\lambda_s} : G_{\lambda_s} \rightarrow \mathcal{T}, \mathcal{E}$ that generates the training/testing splits. We consider $\mathcal{E}$ an OOD test set with respect to $\mathcal{T}$ and a given λ_s threshold, if $s(x_i, x_j) < \lambda_s \; \forall \; x_i, x_j \in \mathcal{T}, \mathcal{E}$.

We propose the CCPart (**C**onnected **C**omponents **Part**itioning) algorithm to implement Φ_{λ_s}. The algorithm first constructs a graph G_{λ_s}, where nodes are entities and edges connect nodes with similarity greater than λ_s. It then identifies all unconnected subgraphs of G_{λ_s}, $\{U_1, U_2, \ldots, U_k\}$. Smaller subgraphs are considered more unique, and the evaluation set is built by prioritizing smaller subgraphs. Additionally, stratified sampling can be applied to maintain

[3]Common model performance metrics such as accuracy, recall, precision, Matthew's, Pearson's, or Spearman's correlation coefficients are inversely proportional to risk.

label balance. CCPart thus focuses the evaluation on the most dissimilar regions of the dataset distribution. Figure 2 illustrates the process (see Algorithm S1).

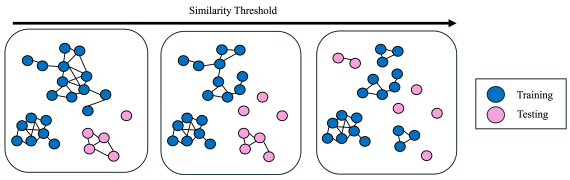

Figure 2: Schema showing how CCPart builds the training and testing subsets based on the topology of the similarity graph at different similarity thresholds.

CCPart differs from other algorithms like GraphPart [8] in that 1) it does not remove any data points, preventing information loss and 2) similarly to perimeter split [12] or maximum dissimilarity split [11], it biases the testing dataset towards the most unique (i.e., the most OOD) points in the dataset. However, unlike the perimeter or dissimilarity split algorithms, it enforces a similarity threshold between the entities in training and testing. Using CCPart as the sampling strategy will lead to a more pessimistic estimation of the expected model performance, whereas using more unbiased sampling strategies like GraphPart [8] or the mixture of splits proposed by [11] will lead to a more precise estimation of the expected model OOD performance. Our framework is agnostic to the specific partitioning strategy used and choice between these methods will depend on the specific use case, however, we consider that the more challenging estimation might help improve trust in models in real-world scenarios by generating more sober expectations of model reliability in OOD situations.

2.4 Statistical test for comparing AU-GOOD values

Generally, comparison between the statistical significance of the difference in performance between two given models is made by comparing the confidence intervals (or a proxy value, like the standard deviation) of the average performances across several runs or cross-validation splits. This standard practice, implicitly assumes that the distribution of the performance values across the different runs or cross-validation splits is normally distributed or tends to normality given enough samples.

However, the different performance values at different thresholds used to compute the AU-GOOD metric cannot be assumed to be normally distributed as the values are dependent on the similarity threshold. Therefore, the analysis of the statistical significance of the difference between individual AU-GOOD values[4] requires a different kind of statistical test. We propose to use the non-parametric equivalent to the one-tailed T-test, the one-tailed paired Wilcoxon ranked-sum test [13]. Given two paired distributions A and B, the null hypothesis (H_0) of this test is that the distribution of $A_i - B_i$ is symmetric around $\mu = 0$; the alternative hypothesis (H_1) for the one-tailed test is that the distribution of $A_i - B_i$ is symmetric around $\mu > 0$.

If we want to compare the performance of n models, $n \times n$ pairwise tests can be performed, generating a matrix that collects the probability of the null hypotheses. The standard p-value threshold conventionally used for considering that the null hypothesis does not explain the observations is 0.05 according to general consensus. In cases where we are performing multiple tests ($n > 5$), the Bonferroni correction for multiple testing [14] can be applied so that the significance threshold depends on the number of models: $0.05/n$.

2.5 HESTIA: Computational embodiment

We have released the code open source [5], including: 1) wrappers for most common similarity functions between a) small molecules, b) biosequences including protein and nucleic acids, c) protein structures, and d) pre-trained representation learning model embeddings; 2) implementation of different partitionings algorithms such as: a) GraphPart and b) CCPart; and 3) calculation of AU-GOOD values. More details in Appendix - A.

[4]Please note that average AU-GOOD values across several runs can still be compared with the normal confidence intervals as they are expected to be normally distributed (with a big enough sample size).

[5]Anonymized Repository: `https://github.com/IBM/Hestia-OOD`

3 Experiments

We have chosen two different real-world use cases to demonstrate the insights that can be drawn from using our framework for the comparison of ML models within the biochemical domain. The first one is the selection of predictive models for pharmaceutical properties given a target deployment distribution and the second is the selection of what protein language model embedding to use for different target deployment distributions (see Appendix - 4).

3.1 Use case 1: Prediction of pharmaceutical properties of small molecules

ADMET (Absorption, Distribution, Metabolism, Excretion, and Toxicity) properties encompass different biophysical attributes of small molecules that may affect both their pharmacokinetic and pharmacodynamic behaviour within the organism. Computational predictions of these properties are used routinely in the pharmaceutical industry to inform the prioritization of candidate molecules for advancing to the next phases into the drug discovery pipeline from lead optimization to pre-clinical and clinical trials [15]. The estimation of which model will generalise better to which target deployment distribution, will allow researchers to decide on what model to use for each particular project. For the purposes of this study, we have selected the Drug Repurposing Hub [16] as the library of choice as it represents a realistic scenario with known approved drugs.

We examine our framework with seven datasets containing different ADMET properties from the Therapeutics Data Commons collection [17]: Ames' mutagenicity [18], cell effective permeability in Caco-2 cell line [19], drug-induced liver injury (DILI) [20], acute toxicity LD50 [21], drug half-life [22], parallel artificial membrane permeability assay (PAMPA), and skin reaction [23].

Each experiment described uses as featurization method Extended-Connectivity FingerPrints (ECFP) with radius 2 and 2,048 number of bits, as they are a well-known baseline for all the datasets under consideration [17]. For each experiment, Bayesian hyperparameter-optimisation is conducted with Optuna [24] (see Table - S1). Each experiment consists of 5 independent runs with different seeds.

3.1.1 What is the best similarity function?

Experimental setup

The first step for using the Hestia framework is to determine what is the optimal similarity function to use. We evaluated three types of fingerprints ECFP with radius 2 and 2,048 bits, MACCS, and chiral Min-hashed atom pair (MAPc) with radius 2 and 2,048 bits. We considered 4 binary set similarity functions for the binary fingerprints (ECFP and MACCS): Tanimoto [25], Dice [26], Sokal [27], and Rogot-Goldberg [28]. We also considered cosine similarity as an alternative geometry-based similarity function. For MAPc, which is not a binary fingerprint, we only considered the Jaccard similarity.

Criteria for selecting similarity function

We observed that different similarity functions led to GOOD curves with notably different shapes (see Appendix - C). The two properties of the curves that experienced the biggest variance across similarity functions were 1) their dynamic range, i.e., the difference between the minimum and maximum similarity thresholds that could be used to generate viable partitions with test sets with at least 18.5% of the total data[6]; and 2) their slope, i.e., the relationship between similarity threshold and model performance changes. Additionally, the functions with the lowest dynamic range tended to have the greatest variance in their slopes and resulted in visually incoherent GOOD curves. This is easily explained by the calculation of the GOOD curve being more sensitive to noise, when there are not enough thresholds sampled.

We propose the following selection criteria: 1) Choose the similarity function with the largest dynamic range. If multiple functions are close (within 0.1–0.2); 2) select the one with the highest slope, provided its variance is comparable to others; otherwise, choose the next best. The first criterion aims to produce GOOD curves that span a greater similarity range to minimise the impact of noise. The second focuses on improving discrimination between low and high similarity thresholds for better evaluation of model generalization.

Effect of similarity on the GOOD curve dynamic range

We, then, analysed the effect of the similarity function on the dynamic range of the GOOD curves. Figure 3.A summarises the results across the 6 datasets. It is clear that the MAPc fingerprint with Jaccard similarity presents the most comprehensive dynamic range across all datasets, which is consistent with prior literature describing this similarity function as the one with most resolution for chemical structure recovery from chemical databases [7]. On the other

[6]We keep the conventional 20% value and allowed for a 1.5% margin of error.

hand, the narrowest dynamic ranges corresponded to the MACCS fingerprints, which was to be expected due to the small size of the fingerprints (166 as opposed to ECFP and MAPc with 2,048) leading to worse resolution. Finally, ECFP fingerprints show stable performance consistent with their widespread use for chemical structure recovery [6]. Finally, the effect of the specific similarity index is noticeable, with Sokal outperforming the more common Tanimoto.

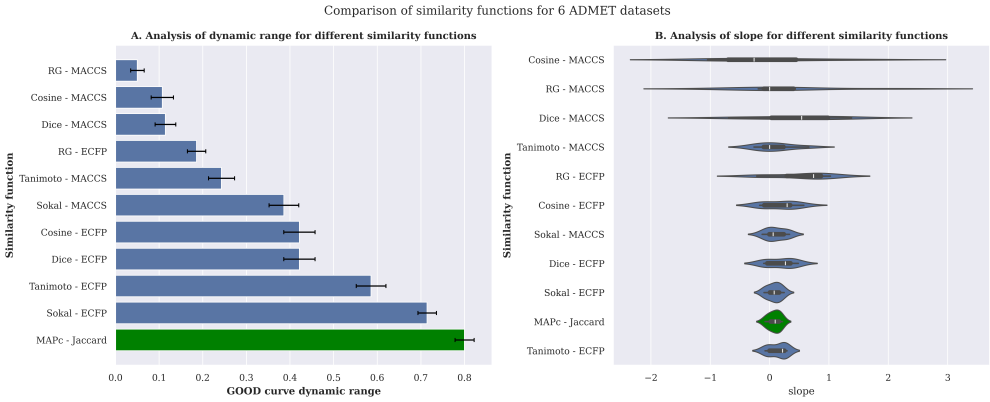

Figure 3: A: Analysis of dynamic range for different similarity functions. Error bars correspond to the standard error of the mean across all 6 datasets. B: Analysis of slope for different similarity functions. Violin plot show the dispersion in the GOOD curve slope across all 6 datasets and 5 runs per dataset (total of 30 experiments).

Effect of similarity function on the slope of the GOOD curve

We then considered the effect of the similarity function on the slope of the GOOD curves (see Figure 3.B). The main differences occur with the fingerprints with the lowest dynamic range showing increased variance. On the other hand, the similarity functions with larger dynamic ranges tend to converge to the same slope values and variances. These observations inform the prioritization of dynamic range over slope in our proposed selection criteria.

Visualisation of the partitioning algorithm

We also demonstrate that the CCPart algorithm is able to generate OOD-challenging partitions at every threshold by visualizing the overlap between the training and testing partitions (generated with MAPc and Jaccard similarity) using uniform manifold approximation and projection (UMAP) to project the ECFP fingerprints of each dataset into two dimensions [29]. Figure 4 shows how the testing partition of the Cell-effective permeability dataset gets clustered together at low thresholds and gets more evenly distributed over the training data distribution as the threshold increases. It is particularly interesting how the partitions at threshold 0.9 are still less evenly distributed than those obtained with random partitioning. Figure S3 contains similar representations for the rest of the datasets considered in the study.

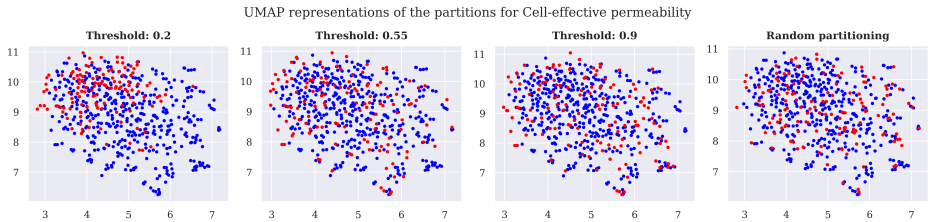

Figure 4: UMAP representation of the chemical space covered by the training (blue) and testing (red) partitions of the Cell-effective permeability dataset at different similarity thresholds.

3.1.2 What models generalise better to the Drug Repurposing Hub?

Experimental setup

We evaluated the performance of four traditional ML algorithms to build models for the ADMET datasets: K-nearest neighbours (KNN), support vector machines (SVM), random forest (RF), and light gradient boosting (LightGBM) and evaluated their AU-GOOD values using as target deployment distribution the Drug Reporpusing Hub [16].

Demonstration of an in-depth Hestia analysis

To demonstrate the depth of the analysis that can be performed with our framework, let us consider one of the datasets in detail. Figure S4 contains equivalent analyses for the rest of the datasets. Figure 5.A shows the GOOD curves for all models considered. The behaviour of KNN is particularly interesting as it tends to perform worse at lower similarity thresholds and better at higher similarity. This observation is not surprising in itself (KNN explictly depends on the distance to neighbouring data points within the data), but it exemplifies the type of situation our framework is built to address.

Figure 5.B shows the distribution of maximal similarities between training and target deployment distribution. In this particular case, it seems that the distribution is skewed towards higher similarities. Figure 5.C shows the distribution of AU-GOOD values of the Matthew's correlation coefficient calculated for each model against the target deployment distribution. It shows that RF has the best average AU-GOOD value, even though it shows greater variance across runs. Interestingly, KNN is the second best model, despite it performing worse at the lower similarity thresholds. This is because the maximal similarity distribution is skewed towards higher similarities and, therefore, the performance at those thresholds has a greater contribution towards the final AU-GOOD score.

Figure 5.D shows the p-values of comparing the 4 models against each other with the Wilcoxon signed-rank test. In this case, the alternative hypothesis is that Model A (y-axis) has better AU-GOOD values than Model B (x-axis). This clearly shows that RF is significantly better ($p < 0.001$) than the other three models, and that KNN is significantly better than SVM ($p < 0.001$), but not than LightGBM ($0.05 < p < 0.0125$).[7]

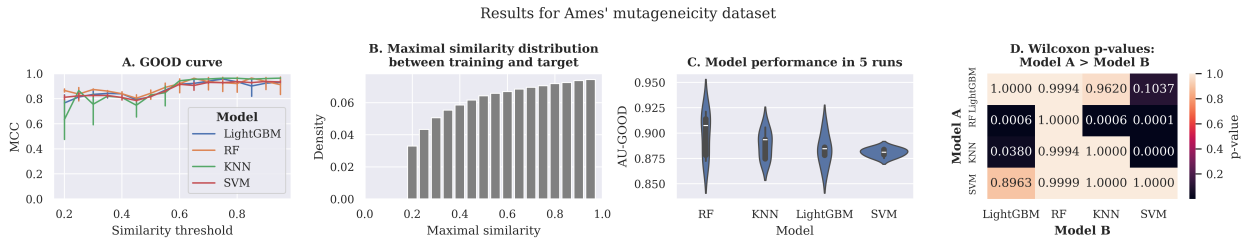

Figure 5: HESTIA analysis for Ames' mutageneicity dataset. A. GOOD curve with MAPc (radius 2 and 2,048 bits) and jaccard index; B. Maximal similarity between entities in Ames' mutageneicity dataset and target distribution (Drug Repurposing Hub); AU-GOOD values across 5 different runs; Wilcoxon test p-values with alternative hypothesis Model A > Model B.

Demonstration of a summary of the Hestia analysis

The results for the rest of the datasets are summarised in Table 1. The significant rank reflects the Wilcoxon signed-rank test results. For each model, it is calculated as the difference between the total number of models and the number of models to which it is significantly superior. For example, the significant rank for RF in Figure 5 would be 4 (number of models) - 3 (number of models RF to which is significantly superior) = 1. KNN and SVM would be $4 - 1 = 3$ and LightGBM would be $4 - 0 = 4$.

Table 1: Comparison between the different downstream models for each dataset. AU-GOOD refers to average AU-GOOD value across 5 runs, the error corresponds to the standard error of the mean. The value the AU-GOOD is calculated for is Matthew's correlation coefficient (MCC) for Ames, DILI, and PAMPA. Spearman's ρ is used for Caco2, Half life, and LD50. S. Rank: Significant rank.

	KNN		**SVM**		**RF**		**LightGBM**	
Dataset	AU-GOOD	S. Rank	AU-GOOD	S. Rank	AU-GOOD	S. Rank	AU-GOOD	S. Rank
Ames	0.884 ± 0.006	3	0.879 ± 0.002	4	0.90 ± 0.01	1	0.881 ± 0.007	4
DILI	0.9632 ± 0.0001	3	0.973 ± 0.003	1	0.960 ± 0.003	3	0.78 ± 0.05	4
PAMPA	0.349 ± 0.004	4	0.81 ± 0.01	3	0.830 ± 0.003	3	0.878 ± 0.003	1
Caco2	0.883 ± 0.002	4	0.941 ± 0.001	1	0.9403 ± 0.0003	2	0.904 ± 0.002	3
Half life	0.8861 ± 0.0001	1	0.842 ± 0.001	2	0.704 ± 0.006	3	0.238 ± 0.006	4
LD50	0.8673 ± 0.0003	2	0.839 ± 0.002	4	0.893 ± 0.002	1	0.753 ± 0.04	4
Average	0.81 ± 0.04	2.8	0.88 ± 0.01	2.5	0.87 ± 0.02	**2.2**	0.66 ± 0.05	3.3

4 Use case 2: Evaluation of Protein Language Models

Protein Language Models (PLMs) are pre-trained representation learning models that rely on the transformer architecture and trained with a masked-language modelling objective to predict the conditional probability that a given residue r will

[7] If we apply the Bonferroni correction, $n = 4$ models, therefore the significant p-value is $p = 0.05/4 = 0.0125$.

occupy position i within a sequence S, given the rest of the sequence, $p(r_i|S-i)$. Recent works like [30] have pointed out that their ability for representation transfer (using the embeddings from the original pre-trained model without further fine-tuning), performs notably worse than in the fine-tuning setting. Properly evaluating which PLMs might perform better as out-of-the-box embedding models is critical for guiding future research to more scalable solutions.

Here, we explore which PLM generates an embedding space that is better suited for generalising to different target deployment distributions. We considered a biophysical property prediction task, Optimal temperature for catalysis [31], from [32]. To this end, we downloaded all protein sequences for three model organisms in the SwissProt database [33]: human (*Homo sapiens*, Baker's yeast (*Saccharomyces cerevisiae*) strain YJM789, and bacteria (*Escherichia coli*) strain CFT073 / ATCC 700928 / UPEC. We included a fourth organism that is less well studied, to provide an extreme case: the human hepatitis B virus (HPV-B) *Orthohepadnavirus hepatitis B virus*.

4.1 Experimental setup

We considered the following models: ESM2-8M [34], ESM2-150M [34], ESM2-650M [34], ProtBERT [35], Prot-T5-XL [35], Prost-T5 [35], Ankh-base [36], and Ankh-large [36]. We extracted the embeddings for all sequences in our downstream datasets and for each sequence s we computed its embeddings using average pooling $s_e = \frac{1}{n}\sum_{i=0}^{n} r_i$ where r_i is the residue at position i and n the total number of residues on the sequence s [37]. We used SVM as the downstream model to reduce the amount of noise in our analyses due to model selection.

4.2 What is the best similarity function?

The task we are considering concerns protein sequence property prediction. We decided to compare two computationally efficient similarity functions MMSeqs2 and MMSeqs2 with prior k-mer prefiltering [5]. The results clearly indicate that MMSeqs2 with prefilter is the best option in all respects i) dynamic range, (Figure S5.A), ii) average slope, and iii) slope variance (Figure S5.B).

4.3 What PLM generates the most useful embeddings?

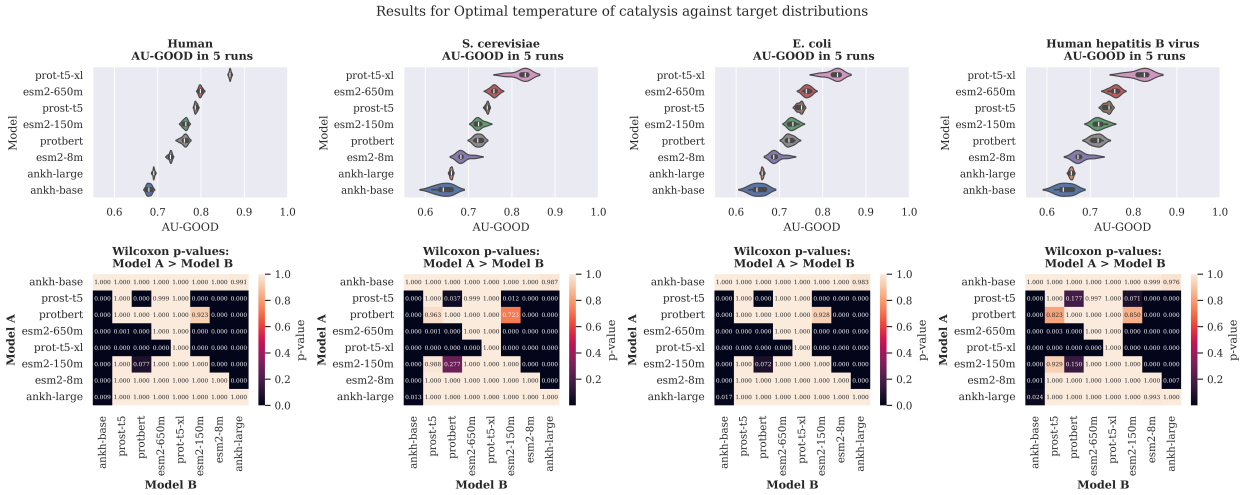

Figure 6: Hestia analyses for Optimal Temperature of Catalysis, comparing different PLM representations. AU-GOOD values correspond to Spearman's ρ. The variance corresponds to 5 different runs.

Figure 6 shows the analyses of the different PLM embeddings (More details in Figure S6). Clearly, the bigger models (Prot-T5-XL, ESM2-650M, and Prost-T5) tend to generalise better than their smaller counterparts. This is particularly clear with the ESM2 models which are ordered in ascending performance according to their size ESM2-8M, ESM2-150M, and ESM2-650M. The worst performing models are the Ankh models, which is surprising as the original paper [36] reported better performance in several tasks than ESM2-650M and Prot-T5-XL. This result is not conclusive as we are only evaluating a single task, and a more extensive evaluation of PLM as embedding models is out of the scope of this study, which focuses on the introduction of the AU-GOOD metric. However, it is an interesting finding that opens the question as to whether scaling down model parameter size might be detrimental for the ability of the model to generalise to new data.

It is also worth noting that, although the ranking of the models is being the same regardless of the target distribution, the significance of the differences between different representation methods varies; and in some cases, like human or the hepatitis virus, the use of a much smaller model like ESM2-150M instead of Prost-T5 could be statistically justified, leading to a more optimal allocation of resources.

5 Conclusion

The AU-GOOD is a new metric for estimating the expected model performance against new target deployment distribution(s) that are out-of-distribution of the training data of a given machine learning model. This metric is applicable to any biochemical entities for which a relevant similarity function can be defined.

The calculation of this metric requires partitioning the training data into training/testing subsets that are increasingly dissimilar. We propose CCPart a new partitioning algorithm to generate challenging splits without the need for removing any data points. We also propose a set of criteria for selecting the most adequate similarity functions for a given dataset. Finally, we discuss how to obtain statistical support for comparing different AU-GOOD values.

We demonstrate the use of this framework for two different use cases: the development of models for predicting properties of small molecules, and the selection of a protein language model to generate embeddings upon which to build biophysical property predictors.

Acknowledgments

RFD was supported by Science Foundation Ireland through the SFI Centre for Research Training in Genomics Data Science under Grant number 18/CRT/6214.

References

[1] David Z Huang, J Christian Baber, and Sogole Sami Bahmanyar. The challenges of generalizability in artificial intelligence for adme/tox endpoint and activity prediction. *Expert opinion on drug discovery*, 16(9):1045–1056, 2021.

[2] Gabriele Corso, Arthur Deng, Benjamin Fry, Nicholas Polizzi, Regina Barzilay, and Tommi Jaakkola. Deep confident steps to new pockets: Strategies for docking generalization. *ArXiv*, 2024.

[3] Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. Colabfold: making protein folding accessible to all. *Nature methods*, 19(6):679–682, 2022.

[4] Felix Fischer, David S Fischer, Roman Mukhin, Andrey Isaev, Evan Biederstedt, Alexandra-Chloé Villani, and Fabian J Theis. sctab: Scaling cross-tissue single-cell annotation models. *Nature Communications*, 15(1):6611, 2024.

[5] 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.

[6] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

[7] Markus Orsi and Jean-Louis Reymond. One chiral fingerprint to find them all. *Journal of cheminformatics*, 16(1):53, 2024.

[8] Felix Teufel, Magnús Halldór Gíslason, José Juan Almagro Armenteros, Alexander Rosenberg Johansen, Ole Winther, and Henrik Nielsen. Graphpart: homology partitioning for biological sequence analysis. *NAR genomics and bioinformatics*, 5(4):lqad088, 2023.

[9] Roman Joeres, David B Blumenthal, and Olga V Kalinina. Datasail: Data splitting against information leakage. *bioRxiv*, pages 2023–11, 2023.

[10] Simon Steshin. Lo-hi: Practical ml drug discovery benchmark. *Advances in Neural Information Processing Systems*, 36:64526–64554, 2023.

[11] Prudencio Tossou, Cas Wognum, Michael Craig, Hadrien Mary, and Emmanuel Noutahi. Real-world molecular out-of-distribution: Specification and investigation. *Journal of Chemical Information and Modeling*, 2024.

[12] Csaba Szántai-Kis, István Kövesdi, György Kéri, and László Örfi. Validation subset selections for extrapolation oriented qspar models. *Molecular diversity*, 7:37–43, 2003.

[13] Frank Wilcoxon. Individual comparisons by ranking methods. biom bull 1 (6): 80–83, 1945.

[14] Carlo Bonferroni. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R istituto superiore di scienze economiche e commericiali di firenze*, 8:3–62, 1936.

[15] Wenyi Wang, Fjodor Melnikov, Joe Napoli, and Prashant Desai. Advances in the application of in silico admet models–an industry perspective. *Computational Drug Discovery: Methods and Applications*, 2:495–535, 2024.

[16] Steven M Corsello, Joshua A Bittker, Zihan Liu, Joshua Gould, Patrick McCarren, Jodi E Hirschman, Stephen E Johnston, Anita Vrcic, Bang Wong, Mariya Khan, et al. The drug repurposing hub: a next-generation drug library and information resource. *Nature medicine*, 23(4):405–408, 2017.

[17] 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.

[18] Congying Xu, Feixiong Cheng, Lei Chen, Zheng Du, Weihua Li, Guixia Liu, Philip W Lee, and Yun Tang. In silico prediction of chemical ames mutagenicity. *Journal of chemical information and modeling*, 52(11):2840–2847, 2012.

[19] Ning-Ning Wang, Jie Dong, Yin-Hua Deng, Min-Feng Zhu, Ming Wen, Zhi-Jiang Yao, Ai-Ping Lu, Jian-Bing Wang, and Dong-Sheng Cao. Adme properties evaluation in drug discovery: prediction of caco-2 cell permeability using a combination of nsga-ii and boosting. *Journal of chemical information and modeling*, 56(4):763–773, 2016.

[20] Youjun Xu, Ziwei Dai, Fangjin Chen, Shuaishi Gao, Jianfeng Pei, and Luhua Lai. Deep learning for drug-induced liver injury. *Journal of chemical information and modeling*, 55(10):2085–2093, 2015.

[21] Hao Zhu, Todd M Martin, Lin Ye, Alexander Sedykh, Douglas M Young, and Alexander Tropsha. Quantitative structure- activity relationship modeling of rat acute toxicity by oral exposure. *Chemical research in toxicology*, 22(12):1913–1921, 2009.

[22] R Scott Obach, Franco Lombardo, and Nigel J Waters. Trend analysis of a database of intravenous pharmacokinetic parameters in humans for 670 drug compounds. *Drug Metabolism and Disposition*, 36(7):1385–1405, 2008.

[23] Vinicius M Alves, Eugene Muratov, Denis Fourches, Judy Strickland, Nicole Kleinstreuer, Carolina H Andrade, and Alexander Tropsha. Predicting chemically-induced skin reactions. part i: Qsar models of skin sensitization and their application to identify potentially hazardous compounds. *Toxicology and applied pharmacology*, 284(2):262–272, 2015.

[24] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[25] David J Rogers and Taffee T Tanimoto. A computer program for classifying plants: The computer is programmed to simulate the taxonomic process of comparing each case with every other case. *Science*, 132(3434):1115–1118, 1960.

[26] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[27] Robert R Sokal and Charles D Michener. A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin*, 38(6):1409–38, 1958.

[28] Eugene Rogot and Irving D Goldberg. A proposed index for measuring agreement in test-retest studies. *Journal of chronic diseases*, 19(9):991–1006, 1966.

[29] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[30] Francesca-Zhoufan Li, Ava P Amini, Yisong Yue, Kevin K Yang, and Alex X Lu. Feature reuse and scaling: Understanding transfer learning with protein language models. *bioRxiv*, pages 2024–02, 2024.

[31] Gang Li, Filip Buric, Jan Zrimec, Sandra Viknander, Jens Nielsen, Aleksej Zelezniak, and Martin KM Engqvist. Learning deep representations of enzyme thermal adaptation. *Protein Science*, 31(12):e4480, 2022.

[32] Bo Chen, Xingyi Cheng, Pan Li, Yangli-ao Geng, Jing Gong, Shen Li, Zhilei Bei, Xu Tan, Boyan Wang, Xin Zeng, et al. xtrimopglm: unified 100b-scale pre-trained transformer for deciphering the language of protein. *arXiv preprint arXiv:2401.06199*, 2024.

[33] UniProt Consortium. Uniprot: a worldwide hub of protein knowledge. *Nucleic acids research*, 47(D1):D506–D515, 2019.

[34] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022:500902, 2022.

[35] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

[36] Ahmed Elnaggar, Hazem Essam, Wafaa Salah-Eldin, Walid Moustafa, Mohamed Elkerdawy, Charlotte Rochereau, and Burkhard Rost. Ankh: Optimized protein language model unlocks general-purpose modelling. *arXiv preprint arXiv:2301.06568*, 2023.

[37] Serbulent Unsal, Heval Atas, Muammer Albayrak, Kemal Turhan, Aybar C Acar, and Tunca Doğan. Learning functional properties of proteins with language models. *Nature Machine Intelligence*, 4(3):227–245, 2022.

[38] Temple F Smith, Michael S Waterman, et al. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.

[39] Peter Rice, Ian Longden, and Alan Bleasby. Emboss: the european molecular biology open software suite. *Trends in genetics*, 16(6):276–277, 2000.

[40] 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.

[41] 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*, 6:1–12, 2011.

[42] Alex CW May. Percent sequence identity: the need to be explicit. *Structure*, 12(5):737–738, 2004.

[43] 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.

[44] 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.

[45] Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. Reoptimization of mdl keys for use in drug discovery. *Journal of chemical information and computer sciences*, 42(6):1273–1280, 2002.

[46] 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.

[47] Oliver Kramer and Oliver Kramer. Scikit-learn. *Machine learning for evolution strategies*, pages 45–53, 2016.

[48] 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.

[49] Aric Hagberg and Drew Conway. Networkx: Network analysis with python. *URL: https://networkx. github. io*, 2020.

[50] 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.

[51] Walid Ahmad, Elana Simon, Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta-2: Towards chemical foundation models. *arXiv preprint arXiv:2209.01712*, 2022.

A Hestia computational suite

A.1 Similarity calculation

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

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

Biological sequences. The similarity function calculated for protein sequences and nucleic acids is the sequence identity in pairwise alignments. Local alignments are calculated using the MMSeqs2 implementation [5] of the Waterman-Smith algorithm [38] with or without prior k-mer prefiltering. Global alignments are calculated using the EMBOSS implementation [39] of the Needleman-Wunch algorithm [40]. More information regarding the empirical differences between local and global alignments can be found in [41, 8]. 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 [42].

Protein structures. The similarity function calculated for a pair of protein structures is the probability that they belong to the same SCOPe family [43]. This probability is approximated using the Foldseek alignment algorithm [44] 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: $s^* := max[s(x, y), s(y, x)]$.

Small molecules. Similarity between small molecules can be calculated using a variety of similarity functions including: Tanimoto [25], Dice [26], Rogot-Goldberg [28], Sokal [27], euclidean, manhattan, or cosine similarity between various types of fingerprints including extended connectivity fingerprints (ECFP) [6], MACCS [45], or MAPc [7].

Representation learning pre-trained model embeddings. Similarity between embeddings includes traditional geometrical distance functions like cosine, euclidean, manhattan, as well as offering the option to use any custom similarity function. The distance functions d are transformed into similarities with the following expression: $s(x, y) = \frac{1}{1+d(x,y)}$

A.2 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 $s(N_i, N_j)$ with values above a threshold λ, thus, $E = \{(n_1, n_2)\ \forall\, n_1, n_2 \in N$ such that $s(n_1, n_2) > \lambda\}$. 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) [46] and MMSeqs (both) [5]. 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 [47] 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 [8] that generalises it to any similarity function 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` [48] and `Networkx` [49].

Connected components partitioning.

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 subgraphs (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 subgraphs in ascending order of number of members, i.e., smaller subgraphs 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.

Algorithm 1 CCPart (Connected Components Partitioning) algorithm

```
Define G(𝒟, λ_s)
U ← Find all unconnected subgraphs of G
Sort U in order of ascending number of elements
𝒯, ℰ ← ∅, ∅
while |ℰ| ≤ 0.185 × |𝒟| do
    Add to ℰ first element of U
    Remove first element from U
end while
𝒯 ← U
return 𝒯, ℰ
```

A.3 GOOD curve slope calculation

The calculation of the GOOD curve slope is performed by linear regression on the performance of the model as a function of the similarity threshold (λ_s). The linear regression is performed using the Levenberg-Marquadt algorithm for linear least-squares regression. The linear regression is performed both with slope and intercept.

B Molecular fingerprints

ECFP (Extended-connectivity fingerprints) [6] are binary fingerprints where an on bit represents the presence of a substructure and an off bit, its absence. The radius determines the number of hops consider in the molecular graph to define each substructure, whereas the number of bits determines the level of information compression applied with smaller numbers leading to higher compression and increasing the likelihood of collisions, i.e., two substructures being assigned the same bit.

We have chosen the values of radius 2 and 2,048 bits as it is their most common configuration. Figure S1 shows how different that radius 2 and number of bits 2,048 consistently perform robustly across the different datasets. They also perform well when the features used for small molecule featurization are Molecular Language Models (MolFormer-XL [50] and ChemBERTa-2 [51]).

MACCS (Molecular ACCess System) [45] are binary fingerprints, as well, with 166 bits. They differ from ECFPs in that there is a direct mapping between the bits and specific, pre-defined, chemical substructures.

MAPc [7] fingerprints are hash-based fingerprints, as ECFP, but they are not binary, each position can be occupied by different hash-values. They have been shown to outperform any other fingerprint in similarity searches with unique

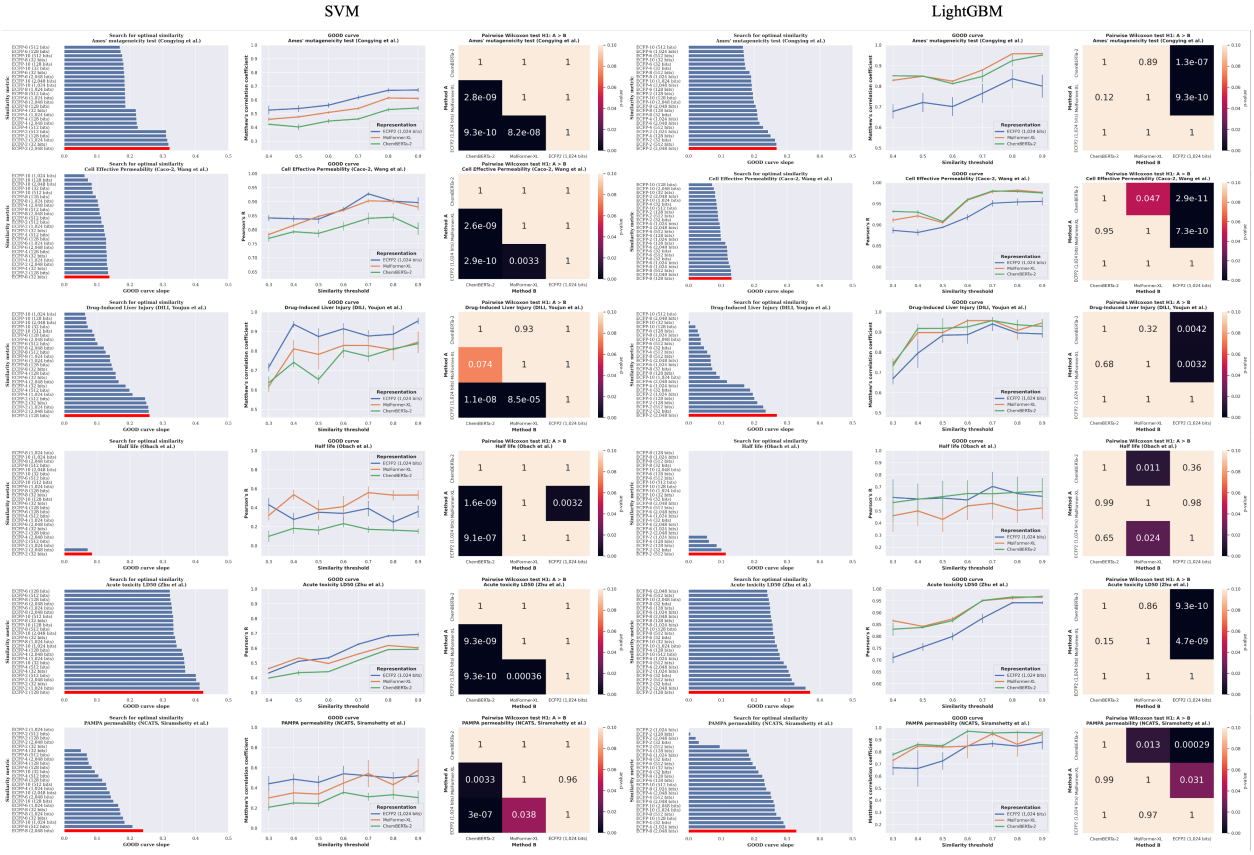

Figure 1: Experiments with ECFP fingerprints with varying radii from 2 to 12 and increasing number of bits from 32 to 4,096. similarity function is Tanimoto. Results are shown with two different downstream models: SVM and LightGBM. Error bars represent standard error of the mean across 10 runs with different seeds.

properties such as being able to discriminate between different stereoisomers of the same molecule. Number of bits and radius was selected based on the best configuration reported by its authors.

C Small molecules choice of similarity function

Figure S2 display detailed results per dataset. It is clear that similarity functions with small GOOD curve dynamic ranges, like all involving MACCS fingerprints, also demonstrate the biggest variance in their GOOD curve slopes. This is completely reasonable, as the smaller the dynamic range, the smaller the number of points comprising the GOOD curve and the more sensitive that the slope calculation will be to random noise.

Overall, the similarity function with MAPc is consistently the one with the largest dynamic range, but also among the highest slopes with low variance. There are certain datasets, like PAMPA permeability, where alternative similarity functions like Sokal or Tanimoto index with ECFP2, might have been more indicated due to their higher slopes, but still, the magnitude of the difference is negligible.

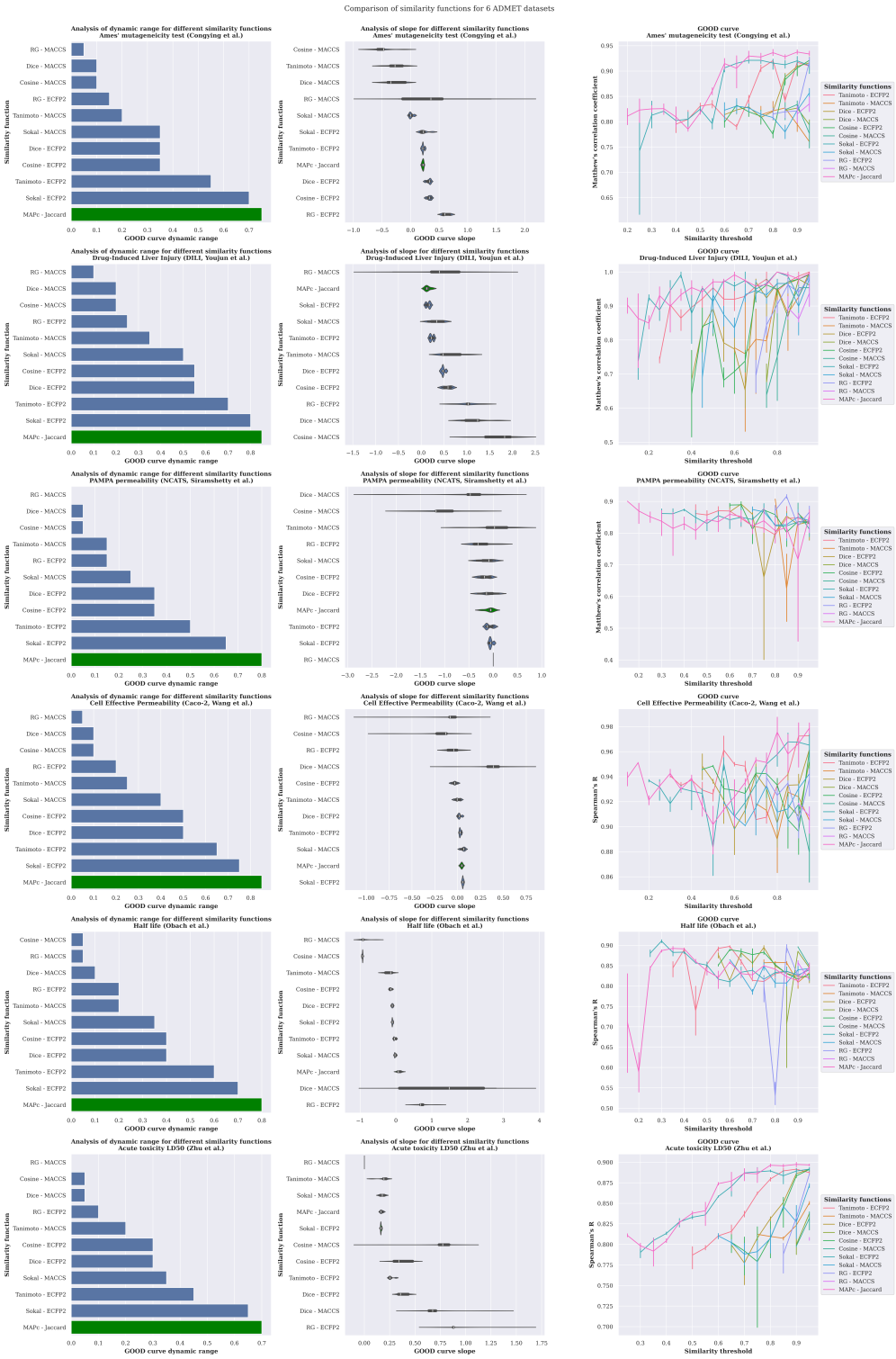

Figure 2: A: Analysis of dynamic range for different similarity functions. Error bars correspond to the standard error of the mean across 5 runs. B: Analysis of slope for different similarity functions. Violin plot show the dispersion in the GOOD curve slope across 5 runs.

D Hyperparameter Search for HPO

Table S1 describes the hyperparameter space defined for all experiments.

Table 1: Hyperparameter search space for each learning algorithm.

Model	Trials	Hyperparameter search space			
		Name	**Type**	**Range**	**Log-scale**
SVM	200	C	float	1×10^{-3} - 10^{3}	Yes
		kernel	categorical	linear, poly, rbf, sigmoid	NA
		degree (only kernel poly)	integer	2-5	No
		coef0 (only with poly or sigmoid)	float	10^{-8} - 1	Yes
		epsilon (only regression)	float	10^{-5} - 1	Yes
KNN	200	K	integer	1-30	No
		Weights	categorical	uniform, distance	NA
		algorithm	categorical	ball tree, kd tree, brute	NA
		leaf size (only with ball or kd tree)	integer	5 - 100	No
		power of Minkowski metric	integer	1 - 3	No
RF	200	number of estimators	integer	10-5,000	No
		criterion	categorical	gini*, entropy*, log loss*, MSE**, MAE**, friedman MSE**	NA
		minimum samples per split	integer	2 - 100	No
		maximum features	categorical	log2, sqrt	NA
		complexity parameter (`ccp_alpha`)	float	10^{-10} - 10^{-3}	Yes
LightGBM	200	number of estimators	integer	10-5,000	No
		minimum child samples	integer	10 - 500	No
		minimum split gain	float	10^{-10} - 10^{-3}	Yes
		regularization α	10^{-10} - 10^{-3}	Yes	
		learning rate	float	10^{-7} - 10^{-1}	Yes

E Visualisation of the chemical space

This appendix contains the visualisation of the level of overlap between the training and testing partitions of the dataset in the chemical space defined by the ECFP fingerprints.

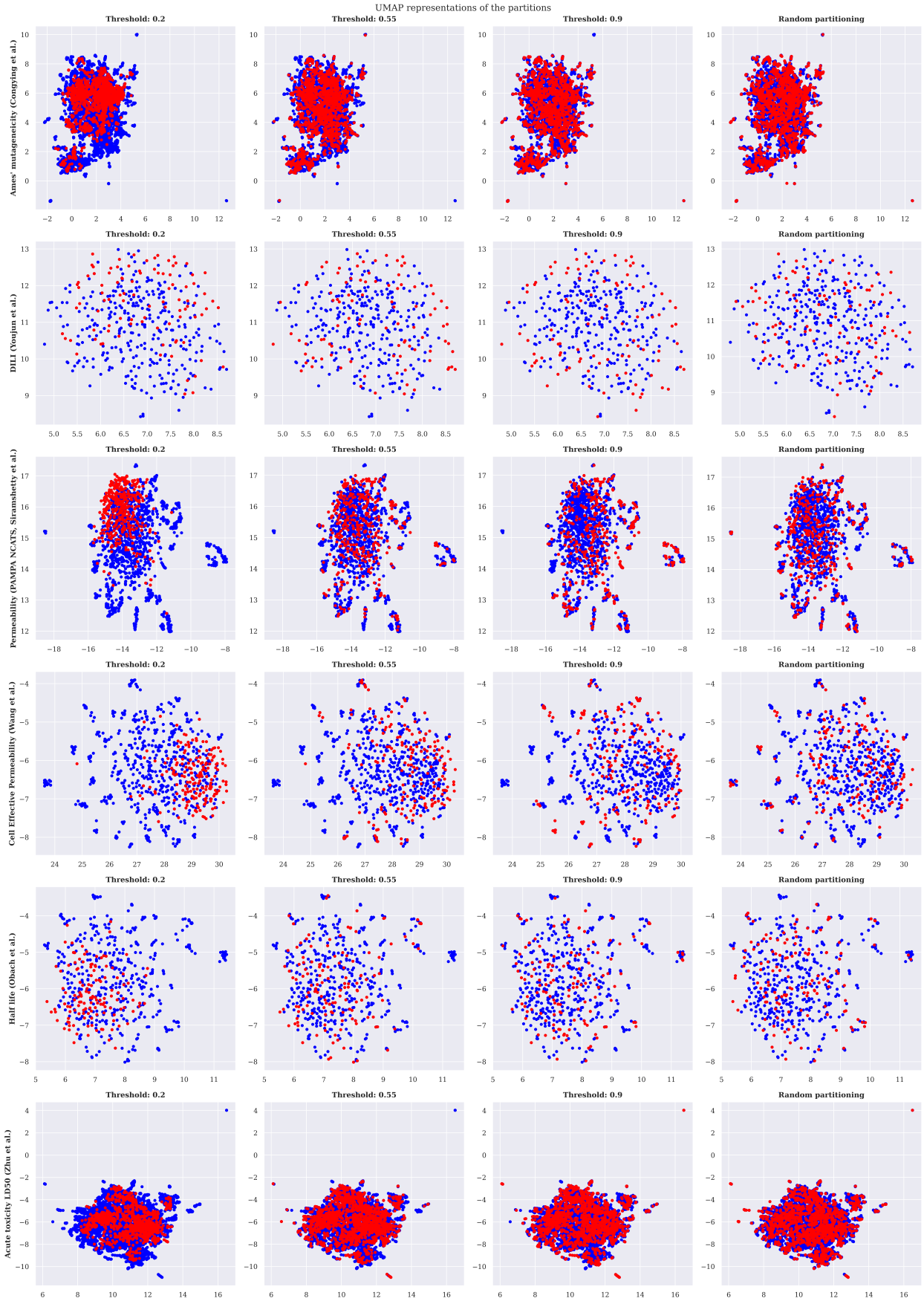

Figure 3: UMAP representation of the chemical space covered by the training (blue) and testing (red) partitions of all molecular datasets considered during this study.

F Detailed Hestia analyses for all ADMET datasets

This appendix contains the detailed Hestia analyses for all ADMET datasets considered in this study.

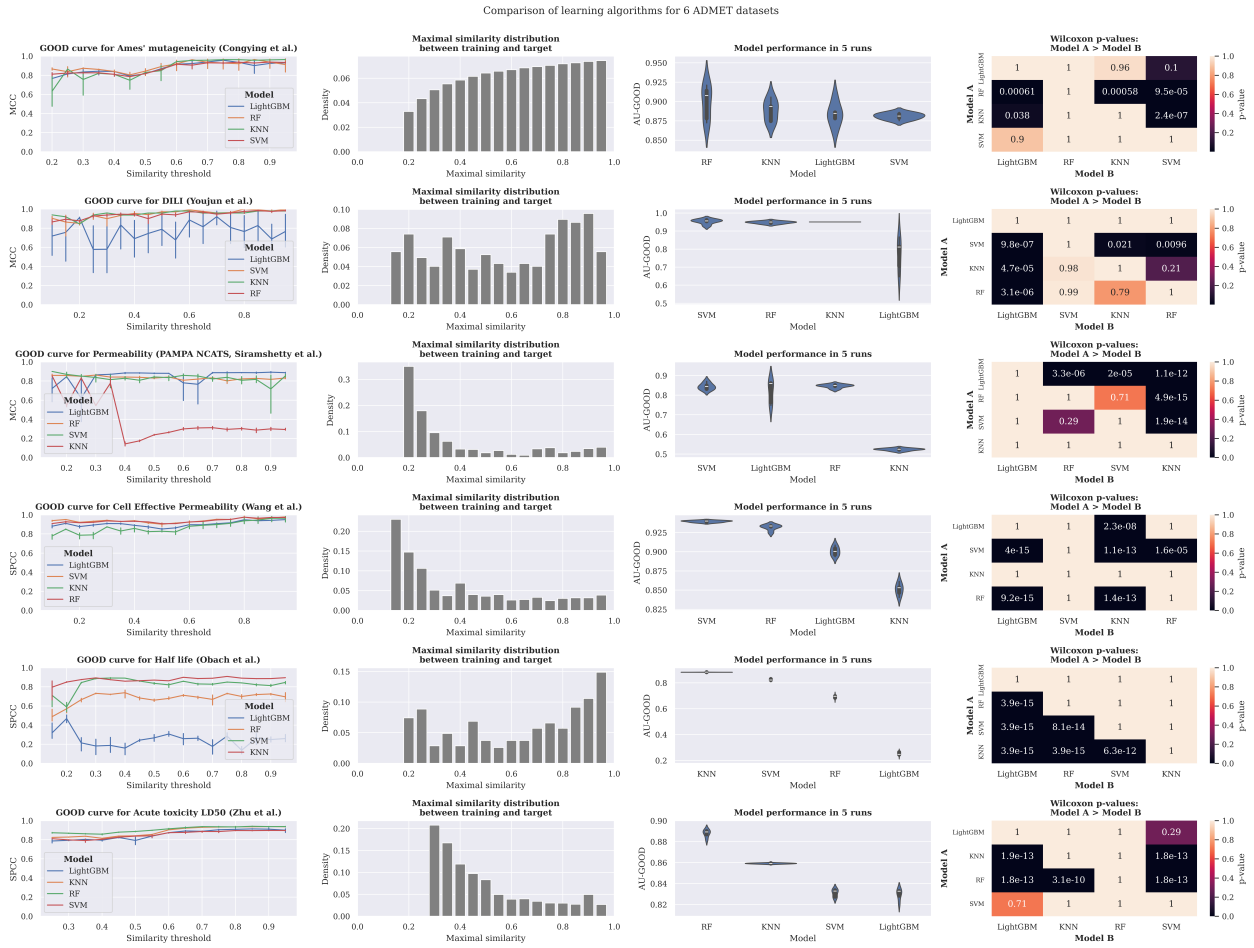

Figure 4: Hestia analyses for the comparison of learning algorithms for the 6 ADMET datasets considered in this study.

G Detailes Hestia analyses for all protein datasets

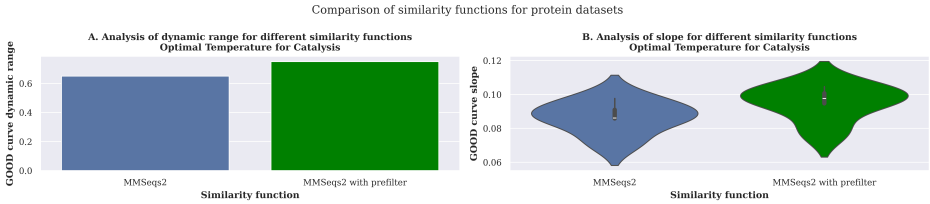

Figure 5: A: Analysis of dynamic range for different similarity functions. Error bars correspond to the standard error of the mean across 2 datasets. B: Analysis of slope for different similarity functions. Violin plot show the dispersion in the GOOD curve slope across 2 datasets and 8 alternative representation methods and 5 runs per dataset and representation (total of 80 experiments)

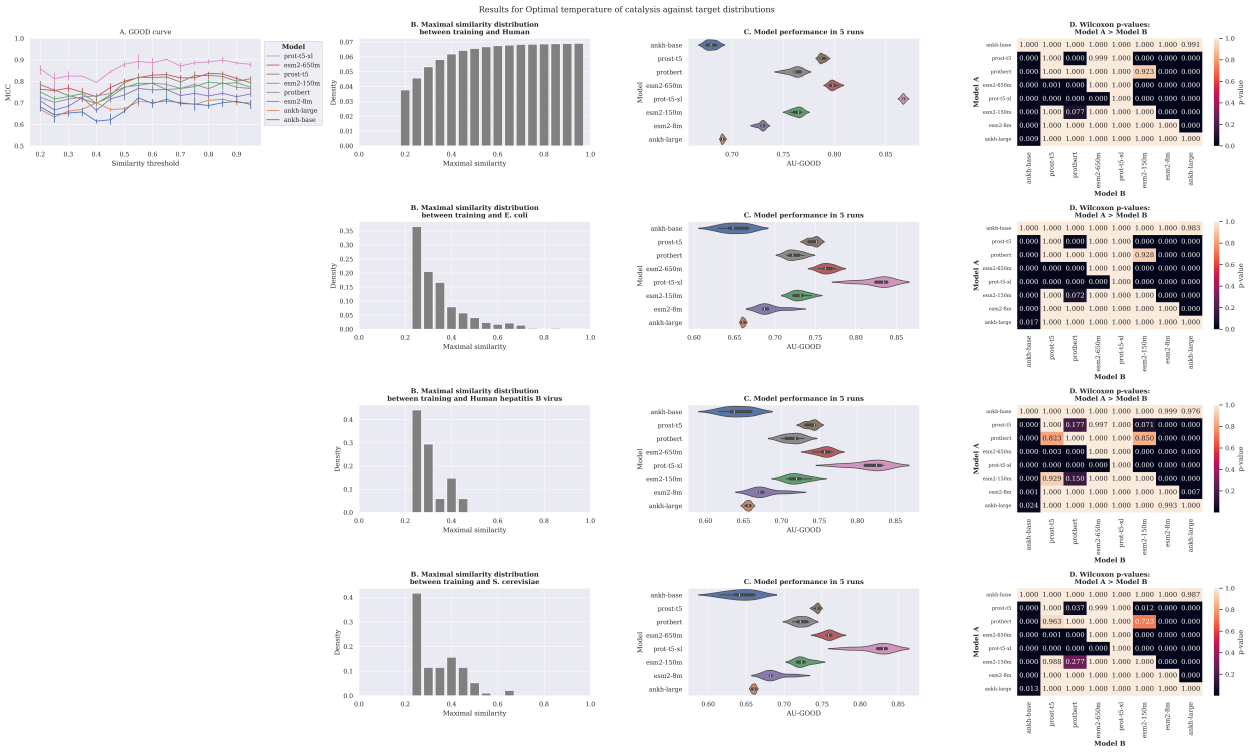

Figure 6: Hestia analyses for the comparison of PLM embeddings algorithms for the optimal temperature for catalysis.