Selection bias in bioinformatics

Benchmarking the effectiveness of domain adaptation techniques in mitigating sample selection bias when leveraging the global domain

Andrei Camil Tociu

TU Delft

Selection bias in bioinformatics

Benchmarking the effectiveness of domain adaptation techniques in mitigating sample selection bias when leveraging the global domain

by

Andrei Camil Tociu

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Friday January 30, 2026 at 15:30.

Student number:	5219183	
Masters programme:	Computer Science	
Faculty:	Electrical Engineering, Mathematics and Computer Science	
Project duration:	February, 2025 – January, 2026	
Thesis committee:	Dr. Joana Gonçalves	TU Delft, supervisor
	Dr. Jorge Martinez Castaneda	TU Delft
	Dr. Yasin Tepeli	TU Delft, daily supervisor

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Preface

This thesis represents the culmination of my long academic journey at the Delft University of Technology towards obtaining my degree of Master of Science in Computer Science. However, my exploration of this topic has begun even before my Masters degree.

In 2023, as part of my Bachelors thesis project, my academic supervisor Joana has introduced me to the concept of tackling sample selection bias through domain adaptation techniques. For three months I had explored this topic from a machine learning perspective with both great interest and success, having presented my work at the BNAIC/BeNeLearn conference. Joana offered me the opportunity to continue our collaboration on this topic during my Masters studies as well and expand the benchmark with new adaptation techniques and evaluation scenarios. Lastly, in 2025, as part of my one-year Masters thesis project, I also explored real-life bioinformatics applications of my research. This last aspect I found particularly interesting because it challenged out of my comfort zone towards biology, a field I had never academically explored before. In summary, this thesis is the result of almost three years of continuous work on this topic, which I found both interesting and rewarding.

I would like to acknowledge the people who made this work possible. First and foremost, I would like to thank Joana Gonçalves, my academic supervisor, for the trust she extended to me and her uninterrupted guidance throughout both my Bachelors and Masters thesis projects. I would also like to thank Yasin Tepeli for his invaluable help, insights and feedback, which elevated the quality of my work. On a personal level, I would like to express my gratitude towards my mother (Dr. ing. Carmen Tociu), uncle (Prof. dr. ing. Gheorghe Maria) and aunt (Dr. ing. Cristina Maria) for their unconditional support throughout my university studies and beyond and for having taught me the value of education. Lastly, I would like to thank all my friends and family for their love and support in this long journey.

Andrei Camil Tociu
Delft, January 2026

Benchmarking the effectiveness of domain adaptation techniques in mitigating sample selection bias when leveraging the global domain

Andrei Camil Tociu,[1,*] Yasin Tepeli[1] and Joana Gonçalves[1]

[1]Pattern Recognition and Bioinformatics, Intelligent Systems Dept., EEMCS Faculty, Delft University of Technology, The Netherlands
[*]Corresponding author. A.C.Tociu@student.tudelft.nl

Abstract

Sample selection bias is a widespread cause of distribution shift between the train and test sets, which can significantly degrade the generalisability and performance of machine learning models. To mitigate distribution shifts, numerous domain adaptation techniques have been developed, which adapt the train set to the test set. However, adapting to a specific test set under sample selection bias might impede the model from properly generalizing across the entire problem domain and requires re-adaptation whenever the test data changes. Therefore, we propose a novel adaptation strategy, called global domain adaptation, in which we instead adapt to a larger (global) domain representative of the distribution from which both the train and test sets originate. We introduce a comprehensive benchmark to investigate the behavior and limitations of domain adaptation techniques when adapting to the global domain, which consists of synthetic datasets and selection biases as well as complex bioinformatics datasets with intrinsic biases. Our benchmark reveals interesting performance patterns across categories of domain adaptation techniques: minimax estimators are very fragile in practice, while deep domain adaptation has lower stability in spite of increased architectural complexity. Lastly, we find that global domain adaptation is a viable approach for certain techniques such as importance weighting, while semi-supervised techniques tend to perform best for existing test set adaptation.

Introduction

For a machine learning model to be useful, it is crucial that it is generalizable. Using biased data, for example due to sample selection bias, may cause a model to learn a view that is not representative of the true patterns. Selection bias occurs when the data are not sampled uniformly from the underlying population, causing certain groups or types of samples to appear more prominently than others. Therefore, selection bias can cause train and test sets to differ substantially, which can lead to serious degradation of a model's generalisability and performance [1]. In practice, sample selection bias occurs in a multitude of situations [1]. For example, clinical studies with non-representative patient recruitment [2], scientific datasets where certain experiments are more feasible or frequently performed than others [3, 4], or bioinformatics where certain proteins are easier to isolate and study than others [5, 6].

In order to mitigate distribution shifts between train and test sets, numerous (unsupervised) domain adaptation techniques have been proposed [1]. They all operate under the assumption that the test set without labels is available at the time of training the model and leverage it in order to adapt the (labeled) train set to the (unlabeled) test set (Figure 1a). However, this approach poses certain limitations under sample selection bias. Adapting the train set to the (biased) test set does not make the model generalizable across the entire distribution of the problem, but only the specific test set. The model will therefore require re-adaptation every time we want it to generalize on new, unseen test data. Furthermore, since domain adaptation techniques are designed to align distributions, they cannot be used when only individual (or extremely few) test samples are available. This motivates

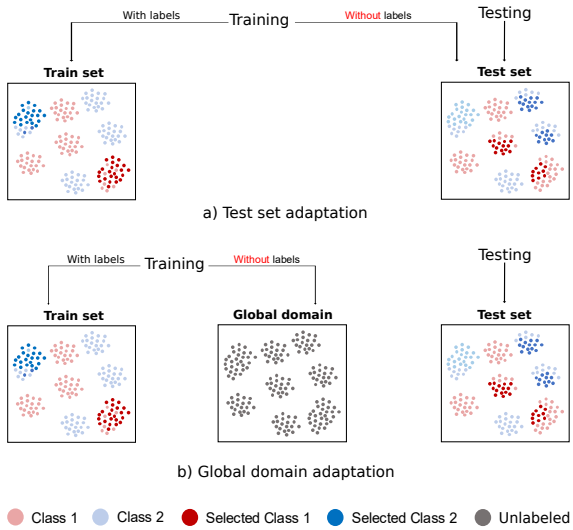

Fig. 1. Domain adaptation approaches. a) Existing approach of using unlabeled data from the test set; b) Proposed approach of using unlabeled data from a wider distribution of the problem domain, here called global domain.

a different, novel perspective: instead of adapting to the test set, one may instead collect unlabeled data from and adapt to the distribution of a broader (here called global) domain from which both the train and test sets are assumed to have been selected with bias (Figure 1b). Intuitively, adapting the train set once to a global domain that is representative enough of the underlying distribution of the classification task should allow a

model to generalize without the need to re-adapt to any new test data that originates from this distribution.

Even though domain adaptation has been widely investigated as a solution to many types of distribution shifts, existing domain adaptation benchmarks (e.g. Office-31 [7], Office-Home [8], DomainNet [9], WILDS [10]) do not investigate the presence of sample selection bias in the datasets they curate. Furthermore, they are also targeted at the existing test set adaptation approach, meaning they do not offer unlabeled global data required for global domain adaptation.

Moreover, studies indicate that the effectiveness of domain adaptation is subject to certain limitations that usually stem from the assumptions that an adaptation technique makes [1]. Some factors widely recognized in literature for potentially influencing their performance are the size of the labeled train set [11–13], the amount of available unlabeled data [14–17] and the severity of the distribution shift [1, 11, 14, 18]. While these factors have been long investigated in the wider context of distribution shifts using the traditional test set adaptation framework, their impact in the context of sample selection bias with the proposed global domain adaptation approach has yet to be studied.

To overcome the current gaps in literature, we propose a benchmark to investigate the effectiveness of domain adaptation techniques in mitigating sample selection bias specifically under the novel global domain adaptation approach. More precisely, we use the benchmark to study:

1. the limiting factors of adaptation techniques that emerge from the dataset characteristics (i.e. global domain sample size, train set sample size, amount of bias in the data);
2. the impact of hyperparameter tuning on the performance of adaptation techniques that use the global domain;
3. the effectiveness of adaptation techniques leveraging the global domain in mitigating intrinsic selection biases;
4. the behavior of adaptation techniques when adapting to the global domain versus test set.

The benchmark encompasses controlled experiments with artificial selection bias that allow to study systematically the limiting factors and the impact of hyperparameter tuning, as well as real-world bioinformatics problems with intrinsic selection bias in the data. Lastly, the bioinformatics datasets are also used to compare the novel global domain adaptation with the existing test set adaptation.

Methodology

Domain adaptation techniques

(Unsupervised) Domain adaptation techniques are very numerous and diverse in terms of the strategy they use to perform the adaptation. While studying all techniques is infeasible, we use the taxonomy proposed by [1] to make a representative selection of techniques. Adaptation techniques are distinguished in [1] based on the approach they use: sample-based techniques correct the distribution shift by adjusting the individual observations in the train set, feature-based techniques transform the feature space such that a classification model trained on the remapped train set will generalize on the unlabeled set, and inference-based techniques incorporate the adaptation directly in the parameter estimation procedure of the classification model. Each of these three approaches is then further split into finer categories based on the adaptation mechanism itself and the assumptions it makes about the adaptation problem. We select for our study 11 adaptation techniques, covering both established and state-of-the-art methods, across five prominent categories (Table 1).

Importance weighting

The importance weighting category belongs to the sample-based approach and is usually employed in clinical applications [1], which makes it relevant to the bioinformatics field as well. It assigns a weight to each sample in the train set, such that the distribution of the weighted train set is more representative of the distribution of unlabeled data. Kernel-Mean Matching (KMM) [11] and Kullback-Leibler Importance Estimation Procedure (KLIEP) [19] are two well known importance weighting techniques that both infer the sample weights by minimizing a distribution discrepancy metric between the train and unlabeled sets. KMM uses the Maximum Mean Discrepancy as metric, while KLIEP employs the Kullback-Leibler divergence.

Semi-supervised

Semi-supervised is an inference-based category that incorporates the unlabeled samples in the training process by pseudo-labeling them, in order to achieve adaptation. Self-training [20] uses a classification model to iteratively assign labels to the unlabeled samples and subsequently selects a subset of them with the highest prediction confidence to add to the train set for the next training iteration. The Co-training approach [21] involves using two different classification models in parallel on the same data [22] to pseudo-label the unlabeled set and then adding some of the highest-confidence predictions from both models to the train set during each iteration.

Subspace mapping

The subspace mapping category follows the feature-based approach and projects the data in both the train and unlabeled sets into a new subspace in which the two are aligned. Subspace Alignment (SA) [23] achieves this by extracting the first d components of the principal component analysis applied to the train set and subsequently aligns them via a linear transformation matrix to the first d components of the unlabeled set. On the other hand, Transfer Component Analysis (TCA) [24] aligns the train and unlabeled sets in an independent subspace by computing a projection kernel matrix that minimizes the Maximum Mean Discrepancy metric between the two sets in this new feature space.

Minimax estimators

Minimax estimators use the inference-based approach. They view domain adaptation as an optimization problem consisting of a classifier that attempts to minimize risk and an adversary that maximizes it by changing the distribution of the unlabeled data from that of the train set. Following on this idea, the Robust Bias-Aware (RBA) classifier [25] assumes an adversary that changes the posterior distribution of the unlabeled data. The Target Contrastive Pessimistic Risk (TCPR) approach [26] focuses on the performance gain that can be obtained by changing the parameters of the classifier, while assuming maximum uncertainty regarding the labels of the unlabeled set.

Deep domain adaptation

Deep domain adaptation is feature-based; it leverages artificial neural networks to extract high-level features that are both common across the train and unlabeled sets, and robust

Table 1. Summary of the benchmark domain adaptation techniques.

Category	Technique	Type	Requires base model
Importance weighting	KMM	sample-based	yes
	KLIEP	sample-based	yes
Semi-supervised	Self-training	inference-based	yes
	Co-training	inference-based	yes (2 models)
Subspace mapping	SA	feature-based	yes
	TCA	feature-based	yes
Minimax estimators	RBA	inference-based	no
	TCPR	inference-based	no
Deep domain adaptation	DANN	feature-based	yes (neural network)
	WDGRL	feature-based	yes (neural network)
	MDD	feature-based	yes (neural network)

against the distribution dissimilarity between the two. Due to the inclusion of neural networks, deep domain adaptation is particularly suitable for high-dimensional applications [1], such as bioinformatics. The Domain Adversarial Neural Network (DANN) [27] is an established method that uses a feature encoder with two loss layers: the first classifies the train samples based on their known labels, while the second classifies the train and unlabeled samples based on their domain such that they cannot be distinguished from each other. Wasserstein Distance Guided Representation Learning (WDGRL) [28] is architecturally similar to DANN, but the second loss layer instead minimizes the Wasserstein distance between the train and unlabeled sets. Lastly, Margin Disparity Discrepancy (MDD) [29] also leverages a feature encoder, but it aligns domains by minimizing the difference in margins between a primary classifier and an auxiliary adversarial classifier, both applied to the extracted feature representation. All three deep domain adaptation techniques follow an adversarial approach, however, DANN and WDGRL rely on domain discriminators, while MDD instead focuses on aligning decision boundaries directly by minimizing the difference in classification margins.

Selection of base classification model

When using an adaptation technique it is important to consider whether or not a base classification model needs to be selected to use in the adaptation process. Certain techniques align the train and unlabeled domains independently of the classification model used on the task and can therefore be applied to (almost) any choice of base model. In this case a decision must be made on which classifier to use with the adaptation technique. Other adaptation approaches incorporate the classification task in the inner workings of the technique itself and therefore require no base model. In this study we use logistic regression as the base model because it is suitable for binary classification tasks and easily interpretable due to its linearity. It also allows for weighting samples during training, which is a prerequisite of importance weighting adaptation. We use the logistic regression implementation from scikit-learn[1] with its default hyperparameter values: L2 regularization and 100 maximum iterations. When dealing with unbalanced train sets we also enable the balancing class weighting function. For Co-training, which uses two classification models, we combine it with linear discriminant analysis (scikit-learn[1]; default hyperparameters) because this is also linear and intuitively explainable. Lastly, deep domain adaptation cannot be straightforwardly applied to any base model because of the neural network architecture. To ensure a fair evaluation, we use as task head in the three techniques a single fully connected layer with sigmoid activation that replicates the behavior of the logistic regression base model used by the other techniques.

[1] https://scikit-learn.org (version 1.7.2)

Controlled experiments with artificially introduced sample selection bias

All adaptation techniques incorporate assumptions in their adaptation mechanism that strongly influence in which situations they succeed or fail [1]. We identified some common factors to which adaptation techniques have been studied to potentially be sensitive to: the sample size of the train set [12, 13], the number of unlabeled samples [15–17], and the degree of distribution dissimilarity between the train and test sets [11, 14, 18]. Therefore, we examine how domain adaptation techniques perform in these situations when they adapt the train set to the global domain. In order to be able to evaluate each of these factors individually, we focus on experiments in which we craft the train, test and global sets ourselves and have full control over the sample selection bias present in the data.

Data and sample selection bias

We evaluate the adaptation techniques on five binary classification tasks (Table 2), which are diverse in terms of sample size (900 - 30,000), number of features (7 - 23) and feature types in order to increase the generalisability of our analysis. First, to mitigate potential confounding factors, we eliminate class imbalance by randomly subsampling the majority class to the sample size of the minority class. As such, the tasks used in the experiments have between 900 and 13,272 balanced samples and between 7 and 23 features split over categorical (between 0 and 8), binary (between 0 and 3) and numerical (between 7 and 20). Afterwards, the data of each of the five tasks is randomly divided, stratified by class labels to maintain class balance, into train (40%), global (50%) and test (10%) sets. At this stage, the data in the three splits is assumed to have a similar distribution due to the random split. Also, we did not yet introduce any artificial sample selection bias in the data.

Table 2. Binary classification tasks used in the controlled experiments.

Task	#Samples	Class balance	#Samples balanced	#Features (cat./bin./num.)
Raisin [31]	900	50% - 50%	900	7 (0/0/7)
Twonorm [32]	7,400	50% - 50%	7,400	20 (0/0/20)
Ringnorm [33]	7,400	50% - 50%	7,400	20 (0/0/20)
Diabetic [34]	1,151	47% - 53%	1,082	19 (0/3/16)
Credit card [35]	30,000	78% - 22%	13,272	23 (8/1/14)

To be able to effectively evaluate the adaptation techniques, there needs to be a distribution difference present between the train and test sets, in the form of sample selection bias. We achieve this by introducing an artificial type of sample selection bias in the train set via the hierarchy bias approach [30]. This approach consists of identifying clusters of samples within each of the two classes in the train set and then favoring samples from a specific, randomly chosen cluster. The bias ratio parameter of the hierarchy bias dictates the percentage of selected samples that originates from this one cluster, with the remaining samples being randomly and uniformly chosen from the rest of the clusters. The selection process maintains an equal number of samples in each class. Ultimately, by being able to control both the number of samples selected with bias and the bias ratio parameter, we manage to control how much bias we introduce. After hierarchy bias is introduced in the train set, it follows a different distribution than that of the global and test sets.

Training of domain adaptation techniques and baselines

Each of the domain adaptation techniques is evaluated by first training it on the (biased) train set alongside the unlabeled global domain and subsequently evaluating it on the (unbiased) test set. During the training process, we tune the hyperparameters of the adaptation technique by performing grid search five-fold cross validation (see Appendix A for the search space); four folds of the train set alongside the global domain are used for training, while one fold is put apart for validation. Since all datasets are balanced, we use accuracy as evaluation metric.

We employ a number of baselines in our study in order to better contextualize the performance of the adaptation techniques. The *No bias* approach trains the base classification model (logistic regression; Section 2.1.6) used by the adaptation techniques on the complete and unbiased train set, therefore giving an indication of what the maximum achievable score for the task can be when no sample selection bias is present. *Bias* trains the base model on the same biased train set used by the adaptation techniques as well, indicating what the classification score is when selection bias is present and no adaptation is performed. Lastly, we use *Random* to validate whether the performance decrease when sample selection bias is used is due to the bias itself and not the diminished sample size. It trains the base model using the same number of selected train samples as *Bias*, however they have been picked randomly instead of with bias. All three baselines are evaluated on the test set and no domain adaptation is performed for any of them.

Varying sample sizes of the train set

Due to the train set sample size affecting some adaptation techniques (importance weighting [11, 12], subspace mapping [1, 13]), we also evaluate how the number of train samples alone influences the performance of the adaptation techniques on the five tasks when they adapt to the global domain. Therefore, we fix the hierarchy bias ratio parameter to 80% across all experiments and then select from the total number of train samples of each task, 60 and 100 balanced data points, respectively. All other experimental settings, including the data splits, are identical for the two sample sizes.

Varying sample sizes of the global domain

We study the impact that the sample size of the unlabeled dataset has on the adaptation performance because it has been recognized as an influential factor in the unsupervised domain adaptation process (importance weighting [1, 14], semi-supervised [15], deep domain adaptation [16, 17]). In order to evaluate its impact without other effects, we fix the parameters of the hierarchy bias to 80% bias ratio and 100 train set selections across all five tasks. Afterwards, we evaluate the adaptation techniques when they leverage all (100%) unlabeled samples in the global domain versus only 10% of them, obtained through random subsampling in order to have minimal data distribution changes.

Varying amount of sample selection bias

We identified that the degree of distribution dissimilarity between the train and test sets has an important impact on the performance of many adaptation techniques [1, 11, 14, 18]. Sample selection bias can potentially cause dissimilarity to increase, which prompts us to study how adaptation techniques perform for a varying amount of sample selection bias in the data. We choose the bias ratio parameter of the hierarchy bias as a means to control the amount of bias we introduce in the train set. The bias ratio influences how much the original train set distribution changes and implicitly its dissimilarity from the test set. Therefore, we fix the number of selections from the train set to 100 and examine the performance of the adaptation techniques when the bias ratio increases from 60% to 80%.

Varying hyperparameter tuning approaches for the domain adaptation techniques

Like all machine learning models, the domain adaptation techniques have hyperparameters as well that need to be tuned. Traditionally, the hyperparameters of machine learning models are tuned using a validation set that originates from the train set. The validation set gives a good approximation of the performance on the test set because the data distribution in the train and test sets is not drastically different. However, this approach can be inadequate in the presence of sample selection bias because it might skew the data distributions. In our experiments, the hierarchy bias we introduce in the train set makes it unrepresentative of both the test and unlabeled sets. Therefore, tuning the hyperparameters of the adaptation techniques on a validation set that originates from the train set could nudge them to fit to the biased train set better

rather than being more generalizable. Therefore, we verify whether in our experiments the adaptation techniques are indeed properly adapting to the global domain by focusing on their choice of hyperparameters. This also allows us to investigate how sensitive are the adaptation techniques to the choice of hyperparameters.

We propose to use a subset of the global domain (together with its labels) as validation set to be able to quantify the effect of setting proper hyperparameters for the adaptation techniques. Even though this approach is unrealistic and cannot be used in experiments or as a means to validate how an adaptation technique performs, it allows us mimic in our controlled experimental setup the scenario of adaptation with perfect hyperparameters to the global domain. Our approach (called Global validation) consists of five-fold cross-validation on the global set. The adaptation techniques are trained on the full train set and adapted to four (unlabeled) folds of the global domain, while a separate fifth fold is used (together with its labels) as validation set. We compare the original train set cross-validation approach described in Section 2.2.2 (which we call Source validation) to the Global validation approach using one of the earlier problem setups with 80% hierarchy bias ratio and 100 train set selections.

Bioinformatics problems with intrinsic sample selection bias

Even though crafting our own train, test and global sets and artificially introducing selection bias gives us a sense of control over the experimental conditions, it is unlikely that the controlled experiments, with limited sample and feature sizes, match the complexity of real-world applications in which sample selection bias is often already inherently present in the data. We therefore turn our attention to the field of bioinformatics, more specifically the established problem of protein function prediction, which is suitable for unsupervised domain adaptation for two main reasons. Firstly, both the train and test samples are often already sourced with natural selection bias present in them due to the constraints in data collection and annotation that arise from limitations in cost, time and experimental feasibility. Secondly, unlabeled proteins are available in abundance, making it easy to collect a global domain. Therefore, we evaluate the adaptation techniques on two established tasks of sequence-based protein function prediction, namely protein solubility and the Gene Ontology(GO) terms. These tasks and their associated datasets have already been investigated in several works [36–40] which will be foundation to set up our problem.

Protein solubility prediction

Knowing whether a protein will be soluble or not is essential because soluble proteins can be more easily studied to understand their structure and functions [40, 41]. This is crucial in many industry areas, for example food processing [42], and production of therapeutic proteins like antibodies and hormones [43]. Traditionally, proteins are produced in standard host cells like *Escherichia* (*E.*) *coli* due to their ability to make large amounts of proteins [41]. However, many proteins made in *E. coli* turn out to be insoluble [44], which prompted the development of machine learning models that predict solubility directly from the amino-acid sequence of the protein [39, 40]. While the binary classification task of predicting whether a protein is soluble or not based on its amino-acid sequence is already well-researched, it can also serve as benchmark for sample selection bias mitigation. Many of the solubility datasets used in experiments contain natural biases from how the proteins were sourced, annotated or pre-processed [5, 6, 41]. This makes them suitable for evaluating adaptation techniques that aim to combat sample selection bias in machine learning.

For this experiment we use the protein solubility dataset [40] from the PEER benchmark [36] because it inherently contains selection bias stemming from the way its data was collected. The train and test sets originate from different sources and have each been subject to various pre-processing steps, a practice that is common for bioinformatics experiments and that makes this type of selection bias relevant. The dataset consists of 62,478 train (class balance: 42-58%), 6,942 validation (42-58%) and 2,000 test (50-50%) protein sequences, all expressed in *E. coli*. Train data was originally compiled in [39] by merging data from the pepcDB [45] and Protein Data Bank (PDB) [46] databases, and subsequently preprocessed by [40] to decrease sequence redundancy to a maximum sequence identity of 90% and then prune out all sequences with a sequence identity >30% compared to the test set. Lastly, 10% of data was randomly selected and put aside to form the validation set. Test data was collected by [44] by combining sequences from three different studies [38, 39, 47], subsequently reducing their redundancy to 30% sequence identity level and then randomly subsampling 2000 balanced samples.

As this benchmark lacks an unlabeled global domain, we create it by selecting the sequences expressed specifically in *E. coli* from the Protein Data Bank (PDB) [46]. This yields around 160,519 sequences from which we subsequently remove the ones already present in the train, validation and test sets. Similarly to [21], we then randomly subsample this unlabeled data and choose 50,000 sequences to form the global domain.

All protein sequences, labeled and unlabeled, are each encoded into 640 numerical features using ESM2 [48], a state-of-the-art pre-trained protein language model. Subsequently, the adaptation techniques are trained using the train set and global domain, and have their hyperparameters tuned on the validation set. Similarly to the previous controlled experiments, we employ a *Bias* baseline, which consists of a logistic classifier trained on the train set and evaluated on the test set without leveraging domain adaptation. However, because the sample selection bias is not artificially introduced in the data but already naturally present in it, we do not have the *No bias* and *Random* baselines. Lastly, since the datasets are imbalanced, we use the F1-score as evaluation metric in order to adequately capture how the models perform.

Gene Ontology(GO) terms prediction

Gene Ontology (GO) represents a standardized vocabulary in the field of bioinformatics for representing the functions of proteins, labeled as GO terms. GO terms span three domains: the Cellular Components (CC) where a protein resides, the Molecular Functions (MF) that it fulfills and Biological Processes (BP) in which it is involved. Considerable efforts have been made for effectively predicting whether a protein has a particular GO term or not, especially utilizing protein sequences [37]. However, sample selection bias remains naturally prevalent in many of the datasets used for GO term prediction mainly due to the annotation practices. For example, mass-annotation methods tend to produce more general GO terms, while single-protein experiments yield more specific annotations [4]. It was also shown that a small group of proteins tends to concentrate most annotations [3]. This makes

Table 3. Tasks used in the bioinformatics experiments.

	Protein solubility	CAFA3 - CC			CAFA3 - MF			Data2017 - MF		
		GO:0044444	GO:0043231	GO:0005737	GO:0003824	GO:0005488	GO:0005515	GO:0030165	GO:0035251	GO:0004540
Dimensions	640		551			677			135	
Train samples	62,478		50,596			36,110			32,280	
Train balance	42-58%	49-51%	54-46%	57-43%	54-46%	58-42%	34-66%	2-98%	2-98%	2-98%
Test samples	2,000		1,265			1,137			3,132	
Test balance	50-50%	38-62%	40-60%	45-55%	33-67%	56-44%	37-63%	2-98%	4-96%	4-96%
Global samples	50,000		50,000			50,000			50,000	

GO terms prediction benchmarks also attractive for evaluating adaptation techniques on sample selection bias.

The first datasets we use are Cellular Component (CC) and Molecular Function (MF) from the CAFA3 [49] benchmark, a global competition for the computational annotation of protein functions. The selection bias originates in this case from the temporal nature of the data collection and annotation process. Specifically, it was discovered that "the distribution of GO categories changes over time as a result of strong biases in the annotation process" [50]. The train samples in CAFA3 have been experimentally collected and annotated before September 2016, the submission deadline, while the test set acquired annotations between September 2016 and November 2017. Therefore, since the train and test sets are split by collection date, they inherently have different distributions and also selection bias in them.

In order to investigate bias mitigation further, we also utilized the dataset Molecular Function (MF) from Data2017 [51] because its data is naturally biased to include only well-studied proteins. The data consists of annotated proteins from UniProtKB [52], but only includes as prediction targets those GO terms supported by at least 200 proteins. As such, the proteins considered for study are those that fulfill popular, well-documented functions represented here as GO terms with many ($\geq$200) annotations. Proteins that fulfill either less studied or more niche functions, thus having only a few related GO terms, are automatically discarded. The authors then randomly split the proteins into train and test sets and pre-processed the data such that only the test proteins that share less than 50% sequence similarity to the train set were kept. While reducing sequence similarity is a standard practice in order to avoid leakage, it inherently introduces selection bias in the test set and increases its distribution shift from the train set [53, 54].

We formulate our problems as proteins being related to a specific GO term or not. For each dataset (CAFA3-CC, CAFA3-MF, Data2017-MF), we select the top three GO terms with the highest train set balance ratio for our benchmark study to mitigate for the potential effect of class imbalance on the adaptation performance. This process yields nine evaluation tasks in total, for which the class imbalance ratios are reported in Table 3. Since the test sets are still visibly imbalanced, we use the F1-score as evaluation metric.

The unlabeled global domain is collected from the SwissProt [55] database, which contains curated protein sequences across all species. For each task in part, the train and test sequences are first discarded from the unlabeled set, which is subsequently subsampled to 50,000 sequences that we use for adaptation.

All protein sequences, labeled and unlabeled, are encoded into numerical features using Lite-SeqCNN [37], a state-of-the-art convolutional neural network engineered specifically for the task of sequence-based protein function prediction. Subsequently, the adaptation techniques are trained and evaluated on the encoded sequences. Since no designated validation set is made available like in the protein solubility task, we tune the hyperparameters of the adaptation techniques using grid search five-fold cross-validation on the train set, identically to the controlled experiments (Section 2.2.2). Lastly, a *Bias* baseline is also used to help contextualize the adaptation performance of the techniques.

Adapting to the global domain versus test set

One of the key components of unsupervised domain adaptation is represented by the unlabeled data, which traditionally is sourced from the test set. In this regard, we purposefully choose in this study to use unlabeled data from a more widespread (here called global) domain than that of the test set. Intuitively, if we adapt to a global dataset that is representative of the problem, the resulting classification model should be better informed, more robust and be able to perform on any test set for the specific problem. Therefore, we check experimentally the validity of this assumption in a realistic setup as well. We repeat all previous bioinformatics experiments but instead of using the global domain, we use the test set without labels to adapt; the original unlabeled global set collected in the experiments is fully excluded. All other experimental settings are kept identical to the original setup.

Results and discussion

To evaluate the ability of the domain adaptation techniques to mitigate selection bias when adapting to the global domain, we focused on a wide range of scenarios that spanned both controlled experiments and realistic bioinformatics problems. Firstly, we investigated some caveats that are specific to adaptation techniques using controlled experiments where we generate the datasets and induce selection bias artificially in them. Secondly, we checked the hyperparameter tuning for the adaptation techniques to verify whether they are properly adapting to the global domain. Thirdly, we focused on evaluating the bias mitigation ability of adaptation techniques for more complex bioinformatics problems in which selection bias is inherently present. Lastly, we used the bioinformatics problems to also verify whether leveraging unlabeled data

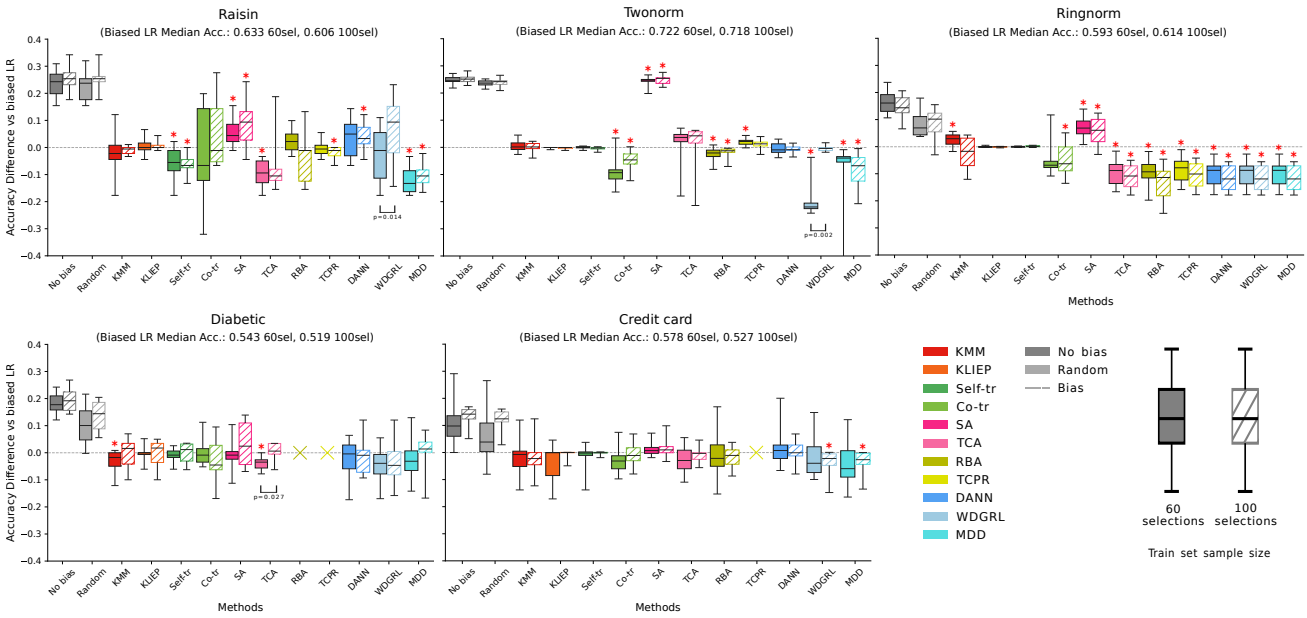

Fig. 2. Performance of the domain adaptation techniques for different sample sizes of the train set in the controlled experiments. Results obtained across 10 runs, with all methods evaluated using the same folds (train/test/global splits). Methods included: 11 adaptation techniques, supervised model trained on unbiased data (No bias), on biased selection of data without adaptation (Bias), and on randomly selected samples (Random, same number as Bias). Adaptation techniques that did not manage to run are marked with X. Red stars indicate significant difference ($p < 0.05$) between the performances of the adaptation technique and the biased supervised method (double-sided Wilcoxon signed-rank test). P: significant difference between the performances of the adaptation techniques on the two sample sizes.

from the global domain instead of the test set yields a better informed classifier that can generalize better on the problem. We evaluated 11 adaptation techniques spanning five categories (Table 1): KMM and KLIEP from importance weighting, Self-training and Co-training from semi-supervised, SA and TCA from subspace mapping, RBA and TCPR from minimax estimators, and DANN, WDGRL and MDD from deep domain adaptation.

For easier interpretation, we rescaled the scores of all adaptation techniques alongside the baselines *No bias* (base model without sample selection bias) and *Random* (base model trained with randomly subsampled samples) by subtracting the score of *Bias* (base model with selection bias and no adaptation) for each experimental run. Therefore, a positive score difference means that the model performed better than *Bias*, while a negative score indicates the opposite.

Investigation of factors influencing the effectiveness of domain adaptation techniques

As discussed in Section 2.2, we investigated how adaptation techniques that leverage the global domain are impacted by factors widely recognized in the literature as affecting their performance, namely the number of samples in the train [11–13] and unlabeled [14–17] sets, alongside the amount of sample selection bias present in the data [1, 11, 14, 18]. We evaluated the adaptation techniques on five real-world datasets (Table 2) in which samples were selected with hierarchy bias from the train set, while the unlabeled and test sets maintained the original distribution.

Varying sample sizes of the train set

In Figure 2, *Bias* performance was lower than both *No bias* and *Random* in all datasets. Compared to the biased model, the median accuracy difference for *No bias* ranged between 0.098 (Credit card) and 0.247 (Twonorm) for 60 selections, and between 0.142 (Credit card) and 0.256 (Raisin) for 100 selections; for *Random* it ranged between 0.039 (Credit card) and 0.239 (Raisin) for 60 selections, and between 0.102 (Ringnorm) and 0.256 (Raisin) for 100 selections. The lower *Bias* performance compared to *No bias* and *Random* indicates that hierarchy bias was effective in introducing selection bias in the train set and that the decrease in performance was due to the bias and not the sample size reduction.

The adaptation techniques did not show uniform behavior patterns across the datasets. Most notably, the majority of techniques struggled to significantly improve the score of *Bias*, even when the sample size increased. SA is the only technique that consistently surpassed *Bias* significantly, in three of the five datasets: Raisin ($p = 0.011$ for 60sel, $p = 0.018$ for 100sel), Twonorm ($p = 0.002$ for 60sel, $p = 0.002$ for 100sel) and Ringnorm ($p = 0.002$ for 60sel, $p = 0.009$ for 100sel). However, there were also adaptation techniques that performed significantly ($p < 0.05$) worse than *Bias* for both 60 and 100 selections, particularly in datasets Raisin, Twonorm and Ringnorm: Self-training and MDD in Raisin, Co-training, RBA and MDD in Twonorm, and TCA, RBA, TCPR, DANN, WDGRL and MDD in Ringnorm. In these three datasets *Bias* had larger median score drops from *No bias* than in Diabetic and Credit card, which indicates that the problem setup created by introducing bias was harder in these instances.

When comparing scores between the train samples sizes, we expected that more samples would ideally translate into a more accurately represented training set and consequently an improved adaptation performance. However, the adaptation techniques did not show a pattern in this regard. The only instances in which increasing the sample size from 60 to

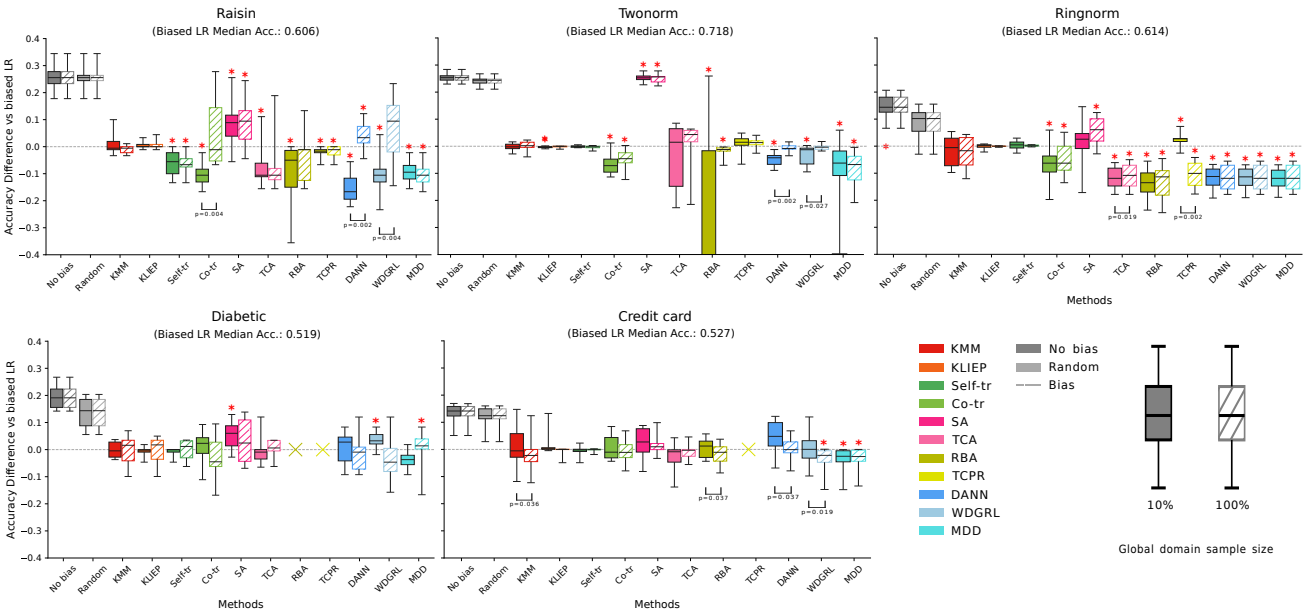

Fig. 3. Performance of the domain adaptation techniques for different sample sizes of the unlabeled global domain in the controlled experiments. Results obtained across 10 runs, with all methods evaluated using the same folds (train/test/global splits). Methods included: 11 adaptation techniques, supervised model trained on unbiased data (No bias), on biased selection of data without adaptation (Bias), and on randomly selected samples (Random, same number as Bias). Adaptation techniques that did not manage to run are marked with X. Red stars indicate significant difference ($p < 0.05$) between the performances of the adaptation technique and the biased supervised model (double-sided Wilcoxon signed-rank test). P: significant difference between the performances of the adaptation techniques on the two sample sizes.

100 selections led to a significant improvement in adaptation performance were WDGRL in Raisin ($p = 0.014$) and Twonorm ($p = 0.002$) alongside TCA in Diabetic ($p = 0.027$). In these instances, while the adaptation techniques performed well with more train samples, they did not significantly improve over *Bias*. In fact, they caused performance degradation when used with 60 samples. A possible explanation for the lack of more consistent improvements is that the 40 extra added samples might have been too few given the complexity of some datasets, which had up to 23 dimensions. However, we refrained from selecting more than 100 train samples because it would have possibly made introducing bias harder as more samples can mean better representation of the original distribution after some point.

Varying sample sizes of the global domain

We investigated another factor that adaptation techniques are sensitive to, namely the sample size of the unlabeled global domain. We kept the selection bias fixed (80% ratio and 100 selections) while we changed the amount of samples in the global domain (Section 2.2.4). Regardless of the global domain size, the introduction of hierarchy bias in the train set caused considerable performance drops for *Bias* in all datasets when compared to both *No bias* (median between 0.142 and 0.256) and *Random* (median between 0.102 and 0.256) (Figure 3). This indicates that the sample selection bias caused a clear distribution dissimilarity that the adaptation techniques can tackle.

We expected the behavior of adaptation techniques to change with varying sizes of the unlabeled set because this would also impact the amount of noise, outliers and the representation of the data, which are all important factors in domain adaptation. We observed some significant improvements when 100% of the global domain was used instead of only 10%: Co-training ($p = 0.004$), DANN ($p = 0.002$), WDGRL ($p = 0.004$) in Raisin, DANN ($p = 0.002$) and WDGRL ($p = 0.027$) in Twonorm, and TCA in Ringnorm ($p = 0.019$). There were also occurrences when adaptation techniques performed better with less unlabeled samples: TCPR in Ringnorm ($p = 0.002$), and KMM ($p = 0.036$), RBA ($p = 0.037$), DANN ($p = 0.037$) and WDGRL ($p = 0.019$) in Credit card. Feeding the adaptation techniques with an excessive amount of unlabeled samples when the underlying class distribution is complex might make it hard to correctly identify the distribution per class in the unlabeled set and consequently confound the targets. It could also introduce more noise and outliers in the unlabeled set. In the case of Credit card, it had the lowest *No bias* median accuracy (0.668) out of all datasets to attest to its complexity, while increasing its unlabeled sample size from 10% to 100% resulted in 5972 new samples, significantly more compared to only 100 train samples.

Lastly, the majority of adaptation techniques did not show significant performance differences between the two unlabeled sample sizes and many of them did also not visibly perform better than *Bias* (Figure 3). Guided by our previous observation on the interplay of the unlabeled sample size and the complexity of the data distribution, we hypothesize that each adaptation technique might have an optimal number of unlabeled data it requires, dependent on the task complexity, for which it performs well. Leveraging too few unlabeled samples does not paint an informative enough picture of the data distribution, while too many samples confuse the technique. The difference between the two unlabeled sample sizes we probed (10% versus 100%) is very large, which makes us believe the techniques did not perform better because their optimal number of unlabeled samples might lay somewhere in between the two values, which requires extensive analysis on every dataset they are applied to.

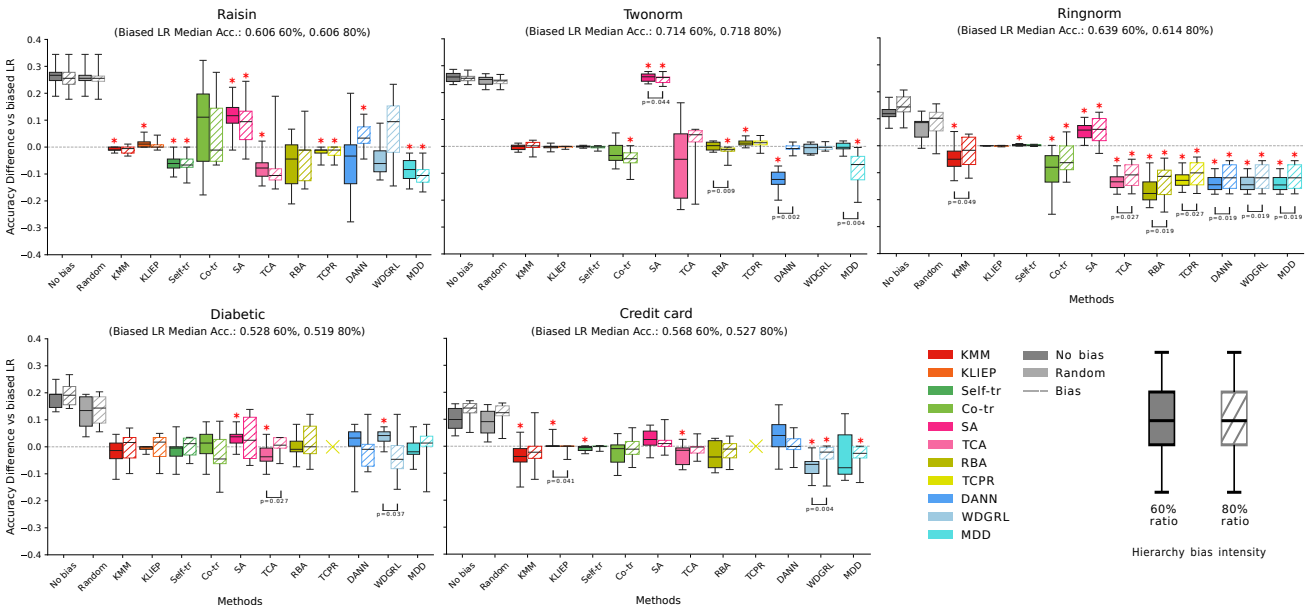

Fig. 4. Performance of the domain adaptation techniques for different hierarchy bias intensities in the controlled experiments. Results obtained across 10 runs, with all methods evaluated using the same folds (train/test/global splits). Methods included: 11 adaptation techniques, supervised model trained on unbiased data (No bias), on biased selection of data without adaptation (Bias), and on randomly selected samples (Random, same number as Bias). Adaptation techniques that did not manage to run are marked with X. Red stars indicate significant difference ($p < 0.05$) between the performances of the adaptation technique and the biased supervised model (double-sided Wilcoxon signed-rank test). P: significant difference between the performances of the adaptation techniques on the two bias intensities.

Varying amount of sample selection bias

We also investigated how the amount of selection bias impacts the adaptation techniques when they adapt to the unlabeled global domain. We kept the number of selected train samples fixed (100) and varied the ratio parameter of hierarchy bias. We expected that increasing the bias ratio from 60% to 80% would skew the original train set distribution more and consequently result in a worse classification performance for *Bias*. The median accuracy difference from *No bias* and *Random* to *Bias* did indeed increase in Ringnorm and Credit card. However, it decreased in Raisin and Twonorm. These results highlight how difficult it is to anticipate the effect of sample selection bias on the data distribution of complex datasets, even in carefully controlled experimental setups. Nevertheless, *Bias* performed visibly lower than both *No bias* and *Random* in all datasets, meaning there was a distribution dissimilarity present for the adaptation techniques to align.

Domain adaptation behavior to the increase in bias ratio was in general very fluctuating. Effectiveness (performance increase compared to *Bias*) of some techniques which were able to mitigate bias for 60% ratio significantly dropped when bias intensity increased: SA ($p = 0.044$), RBA ($p = 0.009$) and MDD ($p = 0.004$) in Twonorm, WDGRL ($p = 0.037$) in Diabetic, and KLIEP ($p = 0.041$) in Credit card. In other cases, even though the technique worked significantly better with more bias intensity, it decreased performance compared to *Bias* regardless of the intensity: DANN ($p = 0.002$) in Twonorm, TCA ($p = 0.027$) in Diabetic, and WDGRL ($p = 0.004$) in Credit card. In Ringnorm in particular, 7 techniques were significantly better with stronger bias, but they all underperformed *Bias* for both intensities. This behavior highlights the complexity of the interaction between the selection bias and the data distribution. The outcome of hierarchy bias in particular is affected by the dataset cluster structure and the number of samples selected. Although hierarchy bias is expected to produce a stronger effect with increased bias ratio, the opposite appears to have occurred for domain adaptation in Ringnorm.

Overall, none of the adaptation techniques proved robust to variations of the amount of bias we introduced via the bias ratio parameter. Techniques that did not often lose effectiveness when increasing bias ratio from 60% to 80% (e.g., KMM, KLIEP, Self-training) usually scored comparable to *Bias* in the first place. The techniques that did show variations in performance (e.g. Co-training, SA) sometimes showed a slight decrease in effectiveness for the higher bias ratio setting; however, this was dependent on the dataset.

The effect of hyperparameter tuning on the adaptation performance

We wanted to evaluate the adaptation techniques when their hyperparameters are chosen optimally for the global domain. Therefore, we repeated the evaluation for one of the earlier experimental setups (80% bias ratio, 100 selections) by tuning their hyperparameters using a labeled subset of the global domain (Global validation) instead of the train set (Source validation) (Section 2). The labels of the global domain should ideally not be used, but they allowed us in this specific setup to estimate the potential of adaptation techniques when their hyperparameters are tuned with an independent unbiased set instead of the biased train set. As shown in Figure 10, introducing sample selection bias produced a visible performance drop in all datasets. The median accuracy difference between *No bias* and *Bias* was in the range 0.142 (Credit card) and 0.256 (Raisin), and between *Random* and *Bias* in the range 0.102 (Ringnorm) and 0.256 (Raisin).

As expected, domain adaptation techniques achieved in general better performance when their hyperparameters were tuned using Global validation. Most notably, the adaptation

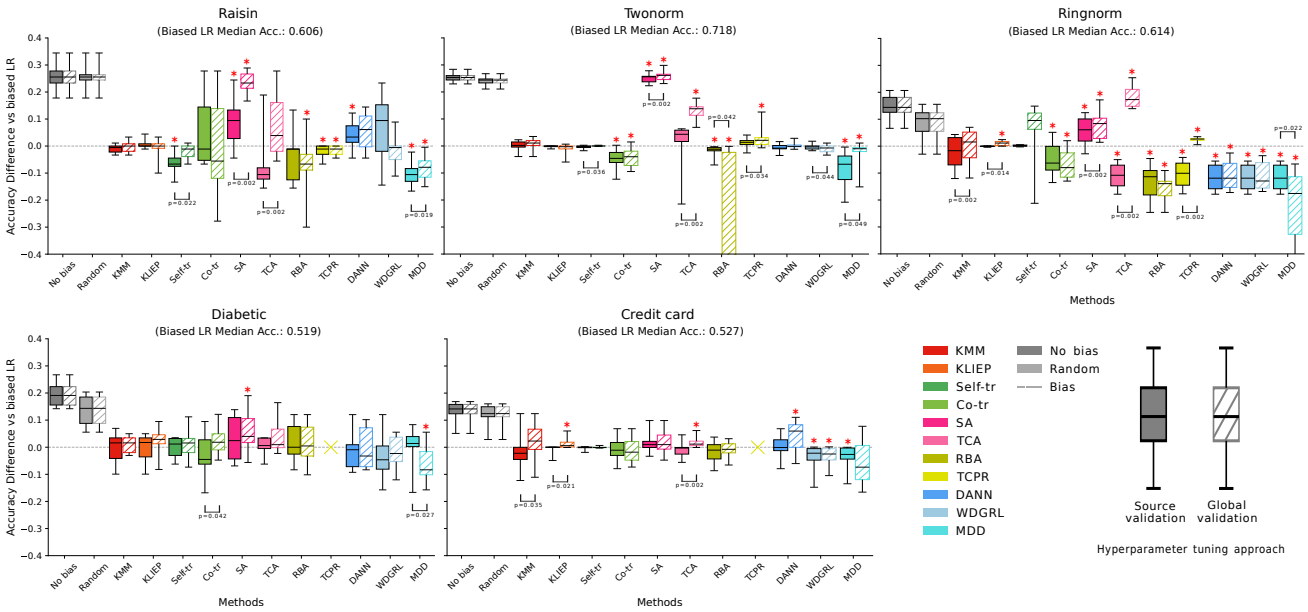

Fig. 5. Performance of the domain adaptation techniques for different hyperparameter tuning approaches in the controlled experiments. Results obtained across 10 runs, with all methods evaluated using the same folds (train/test/global splits). Methods included: 11 adaptation techniques, supervised model trained on unbiased data (No bias), on biased selection of data without adaptation (Bias), and on randomly selected samples (Random, same number as Bias). Adaptation techniques that did not manage to run are marked with X. Red stars indicate significant difference ($p < 0.05$) between the performances of the adaptation technique and the biased supervised model (double-sided Wilcoxon signed-rank test). P: significant difference between the performances of the adaptation techniques on the two hyperparameter tuning approaches.

techniques using Global validation managed in a considerable number of instances to both outperform *Bias* and significantly improve the score for Source validation: SA ($p = 0.002$) in Raisin, SA ($p = 0.002$), TCA ($p = 0.002$) and TCPR ($p = 0.034$) in Twonorm, KLIEP ($p = 0.014$), SA ($p = 0.002$), TCA ($p = 0.002$), TCPR ($p = 0.002$) in Ringnorm, and KLIEP ($p = 0.021$) and TCA ($p = 0.002$) in Credit card. Furthermore, there were also many instances in which the adaptation performance for Global validation significantly outperformed that for Source validation and scored comparably to *Bias*: Self-training and TCA in Raisin, Self-training, WDGRL and MDD in Twonorm, KMM in Ringnorm, Co-training in Diabetic, and KMM in Credit card. Therefore, the current approach of tuning hyperparameters using a validation set that originates from the biased train set can represent a considerable performance bottleneck for the domain adaptation techniques. SA, TCA, KMM and KLIEP in particular showed potential to outperform *Bias*, but failed to do so due to inadequate hyperparameter tuning. Unfortunately, while the usage of the Global validation approach in our experiments did highlight a shortcoming of current practices, it does not represent a solution. The lack of access to the labels of the data to which we adapt is intrinsically specific to unsupervised domain adaptation and that makes it difficult to properly tune the hyperparameters of the adaptation techniques in a straight-forward way.

Lastly, the fact that the scores obtained for Source and Global validation were significantly different from each other on numerous occasions attests to the sensitivity of adaptation techniques to the choice of hyperparameters in general. This was noticeable even for the grid search strategy with a limited hyperparameter search space that we used in the experiments.

Complex bioinformatics problems with intrinsic selection bias

To better characterize the behavior of the adaptation techniques when they adapt to the global domain, we also evaluated them on four different bioinformatics datasets that have naturally occurring selection bias in them (Section 2.3). First, we looked at a protein solubility dataset in which selection bias stems the data collection process. Secondly, we analyzed three GO term prediction datasets: CAFA3-CC and CAFA3-MF containing selection bias that originates from the temporal collection of the train and test sets, and Data2017-MF for which only well studied proteins have been selected. For each dataset we predict three GO terms. For convenience, we refer to a problem as [dataset][GO term], which is predicting whether a protein carries the [GO term] term in the [dataset] dataset.

For each repetition in the bioinformatics experiments, the train and test sets were fixed because they originate from benchmarks whereas the unlabeled set was randomly subsampled anew from the global domain. Because the train and test sets did not change between the repetitions, the variance of the unadapted base model (*Bias*) was low (Supplementary Figure 11). The variance of most domain adaptation techniques was also noticeably low (Figure 6), which indicates that the number of unlabeled sequences (50,000) we subsampled was adequate and the sequences themselves were representative between repetitions. On the other hand, deep domain adaptation techniques in particular tended to have higher variance. This might be explained by the adversarial optimization approach they use [56], for example a domain classifier in DANN, the Wasserstein critic in WDGRL or an adversarial margin classifier in MDD. Unlike the objectives of the other adaptation techniques, these are non-convex and

therefore highly sensitive to initialization settings and choice of hyperparameters [27, 57], which might increase variance.

Adaptation techniques tended to improve the classification score the most when the performance of *Bias* was low. *Bias* registered the lowest median score in Data2017-MF GO:0030165 (0.273), while it scored above 0.5 in all other experiments. For this task in particular, all adaptation techniques except Self-training significantly outperformed *Bias* with robust median differences (0.115 KMM, 0.233 KLIEP, 0.238 Co-tr, 0.044 SA, 0.206 DANN, 0.187 WDGRL, 0.281 MDD). Similarly, for Data2017-MF GO:0004540, in which *Bias* also had one of the lowest median scores (0.534), three techniques significantly improved *Bias* by visible margins (0.051 Co-tr, 0.065 DANN, 0.099 WDGRL), while KMM, KLIEP and MDD showed a tendency to improve as well. Nevertheless, we also noticed that the adaptation techniques had a slightly higher variance in these two datasets in particular. Therefore, more complex tasks with more selection bias represented a better opportunity for adaptation techniques to improve the score, but they also made the adaptation more unstable.

Although semi-supervised techniques successfully improved performance in a few cases (e.g. Co-training in Data2017-MF GO:0030165 and GO:0004540), they struggled in general to adapt and often underperformed *Bias*. Co-training performed significantly worse than *Bias*, sometimes by large margins, in six tasks (median differences: -0.032 Prot. sol., -0.07 CAFA3-CC GO:0043231, -0.245 CAFA3-MF GO:0003824, -0.175 CAFA3-MF GO:0005488, -0.012 CAFA3-MF GO:003016504, -0.046 Data2017-MF GO:0035251). Self-training significantly underperformed *Bias* in four tasks, albeit by smaller margins (median differences: -0.009 Prot. sol., -0.005 CAFA3-CC GO:0044444, -0.005 CAFA3-MF GO:0003824, -0.006 Data2017-MF GO:0030165). The occurrences were spread across all datasets and types of sample selection bias, which makes us hypothesize that the weak performance was mainly the result of the adaptation mechanism itself. Because semi-supervised techniques incorporate pseudo-labeling, they can start hallucinating when the unlabeled samples are low-confidence [58] or at the edge of the distribution that the technique is familiar with [18, 20]. The distribution of the unlabeled data might have been too broad in our bioinformatics experiments, which caused the semi-supervised techniques to learn the wrong decision boundary.

Minimax estimators RBA and TCPR failed to run to completion and instead threw errors for all tasks except RBA in Prot. sol. (Figure 6). Both techniques integrate the base classification model in their adaptation mechanism (see Table 1) and we believe that the increased number of features and the complexity this has introduced in the data might have caused their solver to fail to find a decision boundary. The optimization problem used by minimax estimators to adapt relies on worst-case labeling assumptions on the unlabeled set, which impose highly conservative constraints on the potential decision boundary [1]. These constraints make the feasible solution space very small, especially when the data is high-dimensional or the overlap between the train and unlabeled sets is limited [26]. The bioinformatics experiments had a considerably larger feature space (135-640 dimensions) than the controlled experiments (7-23 dimensions), in which RBA and TCPR mostly managed to run successfully. Furthermore, the few instances in the controlled experiments in which the minimax estimators did struggle to run occurred exclusively in the datasets with high dimensionality or high data complexity (i.e., Diabetic and Credit card). Therefore, while theoretically robust, these adaptation techniques can be numerically fragile in practice.

Lastly, there were also techniques that performed promising when adapting to the global domain (Figure 6). In particular, SA showed consistent robustness and scored significantly better than *Bias* in 6 out of 10 experiments, worse in 1 and did not run in 2. SA also had the largest median improvement out of all techniques in all CAFA3-CC experiments alongside CAFA3-MF GO:0043231. Furthermore, importance weighting techniques KMM and KLIEP each significantly improved *Bias* performance in 6 out of 10 experiments, but KMM performed significantly worse in 3 cases and KLIEP in 1 case. Deep domain adaptation techniques also showed potential, albeit to a lesser extent possibly due to the increased variance in their scores. DANN and WDGRL each significantly outperformed *Bias* in 5 out of 10 experiments, underperformed in 2 and scored comparable to it in 3. MDD however scored mostly comparable to *Bias*, in 6 cases.

Comparing adaptation to the global domain with the test set

We also investigated how our proposed approach of adapting to a global domain that is representative (as much as possible) of the original distribution compare with the existing approach of adapting to the test set. We repeated the bioinformatics experiments when the test set (without its labels) is leveraged as the unlabeled set for adaptation (Figure 6).

When comparing the variance of the scores for global domain adaptation against test set adaptation (Figure 6), we found no conclusive evidence of them being different. This demonstrates that global adaptation can remain as stable as test set adaptation, while still potentially including more information in the unlabeled data.

None of the two adaptation approaches emerged as a distinguishably superior in our experiments. Both global domain adaptation and test set adaptation significantly outperformed the other approach in exactly 23 out of 46 instances. These instances were spread relatively uniformly among the experiments, which indicates that the suitability of the adaptation approach was not dependent on either the dataset or the type of selection bias present in the data. We also analyzed each of these significant instances to reveal patterns among the adaptation techniques. Important weighting techniques usually performed better for global domain adaptation. From the 10 experiments, KMM scored significantly better in 6 and worse in 1, and KLIEP better in 5 and worse in 2. Using the wider global domain likely provided broader and more representative coverage of the test set distribution, enabling KMM and KLIEP to estimate more accurate and stable weights. This is consistent with prior findings that limited support of the test set leads to unstable or high-variance importance weights and therefore an improper adaptation [11, 14, 59]. Relying on unlabeled data exclusively from the test set might have offered limited coverage of the complete test set data distribution, therefore limiting adaptation effectiveness [14, 59].

The semi-supervised approaches performed visibly better when leveraging test set adaptation. Self-training performed significantly better 6 times for test set adaptation and only 3 times for global domain adaptation. Co-training performed significantly better 7 times for test set adaptation and no times for global domain adaptation across the experiments. Our

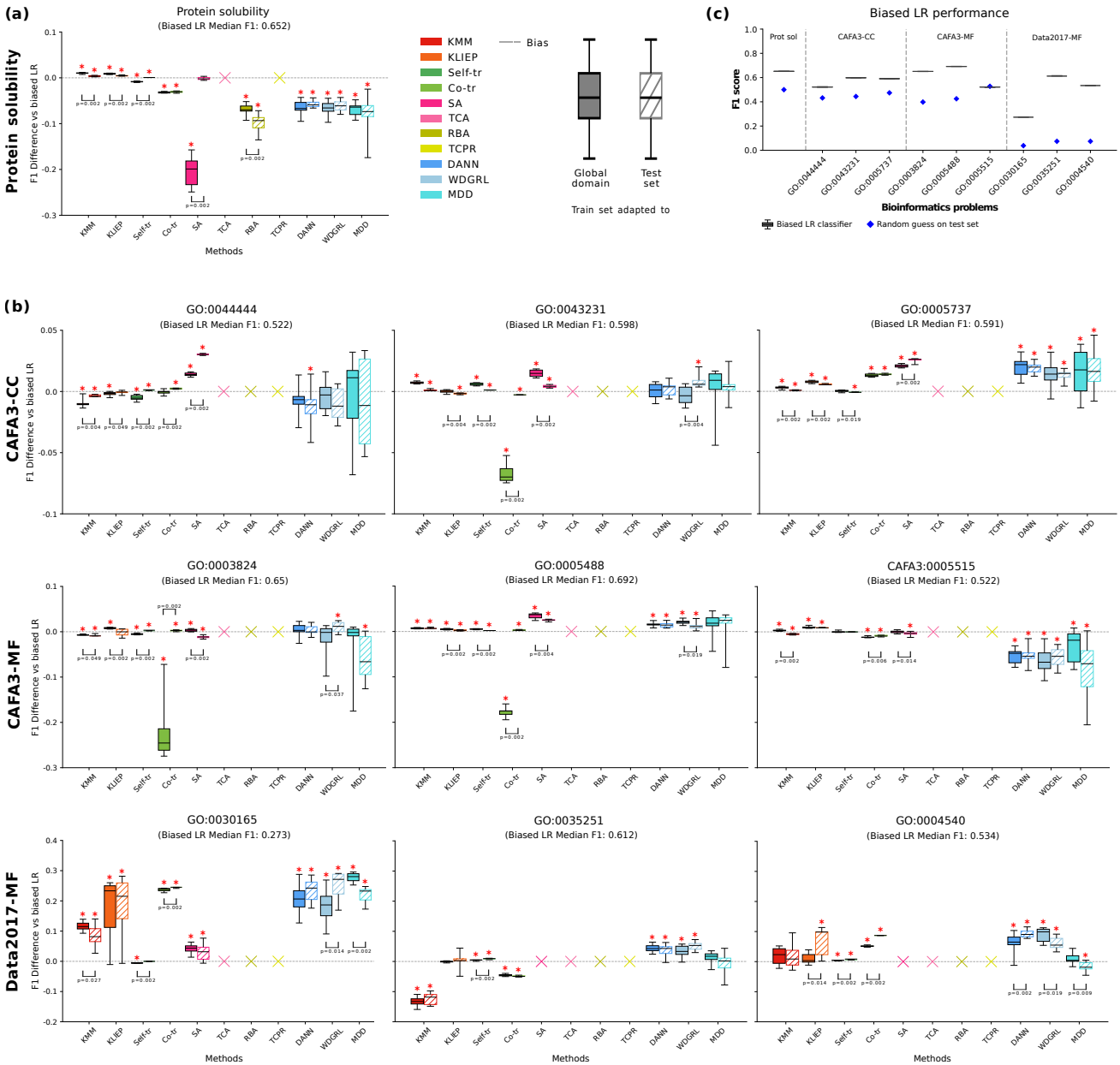

Fig. 6. Performance of the domain adaptation techniques on bioinformatics problems with naturally biased data: (a) protein solubility prediction, and **(b)** prediction of 9 Gene Ontology(GO) terms across 3 datasets. Results obtained across 10 runs, with all methods evaluated using the same train/test/global datasets. Methods included: 11 adaptation techniques and supervised model trained and evaluated on the biased data without domain adaptation (Bias). Adaptation techniques that did not manage to run are marked with X. Red stars indicate significant difference ($p < 0.05$) between the performances of the adaptation technique and the biased supervised model (double-sided Wilcoxon signed-rank test). P: significant difference between the performances of the adaptation techniques using global domain versus test set adaptation. **(c)** Performance of the biased supervised model (Bias) compared to random guess on the test set.

hypothesis is that using unlabeled samples from the broader global domain distribution might have introduced many low-confidence or out-of-distribution instances during the training of the semi-supervised techniques, which increased pseudo-label noise and thereby harmed their performance [58, 60]. It was shown that Self-training performs best when its unadapted base model already performs decently on the test set, and the unlabeled data closely matches the test set distribution [18, 61]. Since *Bias* did score on the test set visibly better than random guess in almost all experiments (6c), we believe that incorporating unlabeled data from a (much) wider distribution than that of the test set is the reason why global adaptation underperformed.

There were also adaptation techniques that did not show a clear performance advantage for either of the adaptation approaches. SA from the subspace mapping category performed significantly better 4 and 3 times, respectively, for global domain versus test set adaptation. The same was observed for the deep domain adaptation category (DANN, WDGRL, MDD) as well, which performed significantly better exactly 4 times for

each adaptation approach. Deep domain adaptation also had a low number of significant differences when compared to the other categories. Both findings could be explained by the higher variance present in the scores of DANN, WDGRL and MDD for both the global domain and test set adaptation scenarios, which made it harder to establish with confidence a better approach overall.

Conclusion

In this study we analyse the performance of (unsupervised) domain adaptation techniques when they mitigate data distribution dissimilarities caused by sample selection bias. Whereas unlabeled data used for performing domain adaptation originates traditionally from the test set, we use a novel approach in which the unlabeled data is procured from the distribution of a larger (here called global) domain representative of the problem. We benchmarked 11 adaptation techniques across both controlled experiments with artificially introduced selection bias and two bioinformatics problems with intrinsic selection bias in the data.

We found that no domain adaptation technique is universally suited for all applications. The properties of the dataset (e.g., sample size, dimensionality, data distribution), the type of sample selection bias, and the approach used by the adaptation technique are all important aspects to consider. In particular, minimax estimators are fragile in high-dimensional tasks in which the distribution difference tends to be inherently more complex, such as protein function prediction. In our experiments, minimax estimators RBA and TCPR failed to run on almost all bioinformatics tasks and sometimes struggled even in the less complex, controlled experiments. Furthermore, deep domain adaptation tends to be less stable compared to other adaptation approaches, especially for more complex problems. Its distinctive neural network architecture combined with an adversarial approach makes the optimization non-convex and sensitive to initialization settings. The novel neural network architecture was also suggested to make deep domain adaptation particularly suited for high-dimensional applications [1], but we found no strong evidence of it outperforming more traditional adaptation approaches in such scenarios. Furthermore, it is particularly ill-advised to employ semi-supervised adaptation approaches when the train and unlabeled sets have little overlap. They tend to easily mislabel samples from outside the learned train set distribution, as prior research also suggests [18, 20, 58]. This can also be the case if the unlabeled set has a large sample size or captures a (very) complex distribution. Lastly, the sample size of the unlabeled set is a factor that affects all adaptation techniques, not only the semi-supervised ones. Intuitively, more unlabeled samples should provide a better estimation of the underlying distribution and aid the adaptation. Nevertheless, our controlled experiments showed that too many unlabeled samples, especially for complex problems, can confuse the adaptation techniques about the correct class distribution and decision boundary.

Our novel approach of adapting the train set to the global domain instead of the test set can be advantageous in certain situations. Most importantly, our bioinformatics experiments showed that global adaptation retains a level of stability similar to that of test set adaptation, while potentially encoding more information about the distribution of data in the problem. Some adaptation techniques, particularly importance weighting ones, benefit from this additional information and often significantly outperformed test set adaptation in our study. Semi-supervised approaches however perform much better when adapted to the test set. Leveraging a wider, more complex unlabeled distribution can cause them to mislabel samples as explained earlier. Techniques from other categories (e.g., deep domain adaptation, subspace mapping) performed similarly for both adaptation approaches and we recommend to be further investigated.

Similarly to other machine learning models, domain adaptation techniques also have hyperparameters that require tuning. Our study provides empirical evidence that they are sensitive to the tuning approach and the subsequent choice of hyperparameters like previous studies suggest [62]. We add to mounting evidence [62–64] that the standard approach of tuning their hyperparameters by using a validation set originating from the train set is inadequate when the train set is unrepresentative of the true distribution, for example due to sample selection bias. This prevents the adaptation techniques from properly adapting to the unlabeled data and represents a significant performance bottleneck as our study shows. While a few alternative hyperparameter tuning approaches have been proposed, such as reverse cross-validation [64] and C-Ent [63], none of them was evaluated on distribution dissimilarities caused explicitly by sample selection bias, which we believe represents a worthwhile future research effort.

Lastly, we reflect on the composition of our benchmark. Controlled experiments with artificially introduced sample selection bias are a very useful tool for providing insights especially into aspects that are not (easily) observable in real-world problems. For example, they allowed us to study empirically the shortcomings of current hyperparameter tuning approaches and to measure the amount of sample selection bias present in the data. Nevertheless, controlled experiments fall short of the complexity of real-world sample selection biases and the bioinformatics problems proved a much better tool for identifying performance patterns across the adaptation techniques. In our study we focused on the bioinformatics field due to the fact that biases are usually intrinsically caused by experimental limitations in cost, time and other resources, while collecting a global domain is feasible because the experimental space is known. However, exposing the adaptation techniques exclusively to selection biases specific to this field can represent a limitation. WILDS2.0 [65] is a recently proposed collection of diverse real-world datasets for domain adaptation that also contain unlabeled data beyond the test set. However, it was not investigated whether the distribution shifts are the result of selection bias or the unlabeled data adequately captures the whole problem domain. We recommend that future studies leverage such initiatives in order to evaluate global domain adaptation on selection biases from more diverse contexts.

To conclude, we consolidated the existing body of knowledge on unsupervised domain adaptation and explored the more specific problem of sample selection bias. We proposed a novel adaptation approach that leverages unlabeled data from a global domain instead of the test set and showed the relevance of our research to the field of bioinformatics by applying our approach to protein function prediction. We hope our findings encourage researchers to further study and use global domain adaptation in more diverse applications.

References

1. W. M. Kouw and M. Loog. A review of domain adaptation without target labels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3):766–785, March 2021.
2. M. A. Hernán and J. M. Robins. Estimating causal effects from epidemiological data. *Journal of Epidemiology & Community Health*, 60(7):578–586, 2006.
3. L. Mowatt et al. Longitudinal analysis of function annotations of the human proteome reveals consistently high biases. *Database (Oxford)*, 2023, 2023.
4. A. M. Schnoes, D. C. Ream, A. W. Thorman, P. C. Babbitt, and I. Friedberg. Biases in the experimental annotations of protein function and their effect on our understanding of protein function space. *PLoS Computational Biology*, 9(5):e1003063, 2013.
5. J. Hon, B. Bhandari, S. Perdue, S. Tian, A. Ray, Z. Fung, Z. Li, P. Kim, and C. R. Buell. NetSolP: predicting protein solubility in *Escherichia coli* using language models. *Bioinformatics*, 38(4):941–948, 2021.
6. D. N. Ivankov and D. B. Lukatsky. A review of machine learning methods to predict the solubility of overexpressed recombinant proteins in *Escherichia coli*. *BMC Bioinformatics*, 15:134, 2014.
7. K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Proceedings of the 11th European Conference on Computer Vision (ECCV) Workshops*, pages 213–226, 2010.
8. H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5018–5027, 2017.
9. X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1406–1415, 2019.
10. P. W. Koh et al. Wilds: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 563–574, 2021.
11. J. Huang, A. Gretton, K. Borgwardt, B. Schölkopf, and A. Smola. Correcting sample selection bias by unlabeled data. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006.
12. W. M. Kouw and M. Loog. An introduction to domain adaptation and transfer learning, 2019.
13. S. Chen, L. Han, X. Liu, Z. He, and X. Yang. Subspace distribution adaptation frameworks for domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, 31(12):5204–5218, 2020.
14. H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.
15. A. Dayal et al. Improving unsupervised domain adaptation: A pseudo-candidate set approach. In *Computer Vision – ECCV 2024*, pages 127–144, Cham, 2025. Springer Nature Switzerland.
16. M. HassanPour Zonoozi and V. Seydi. A survey on adversarial domain adaptation. *Neural Processing Letters*, 55:2429–2469, 2023.
17. A. Ramponi and B. Plank. Neural unsupervised domain adaptation in NLP—A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855. International Committee on Computational Linguistics, 2020.
18. X. J. Zhu. Semi-supervised learning literature survey. Technical Report 1530, University of Wisconsin-Madison, 2005.
19. M. Sugiyama, S. Nakajima, H. Kashima, P. Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.
20. D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, ACL '95, page 189–196. Association for Computational Linguistics, 1995.
21. A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, page 92–100, New York, NY, USA, 1998. Association for Computing Machinery.
22. S. A. Goldman and Y. Zhou. Enhancing supervised learning with unlabeled data. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, page 327–334, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
23. B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *2013 IEEE International Conference on Computer Vision*, pages 2960–2967, 2013.
24. S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.
25. A. Liu and B. Ziebart. Robust classification under sample selection bias. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.
26. W. M. Kouw and M. Loog. Robust domain-adaptive discriminant analysis. *Pattern Recognition Letters*, 148:107–113, August 2021.
27. Y. Ganin et al. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.
28. J. Shen, Y. Qu, W. Zhang, and Y. Yu. Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018.
29. Y. Zhang, T. Liu, M. Long, and M. Jordan. Bridging theory and algorithm for domain adaptation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7404–7413. PMLR, 09–15 Jun 2019.

30. Y. I. Tepeli and J. P. Gonçalves. Dcast: Diverse class-aware self-training mitigates selection bias for fairer learning, 2024.
31. İ. Çinar, M. Koklu, and S. Tasdemir. Raisin. UCI Machine Learning Repository, 2020.
32. M. Revow. Twonorm. Open Machine Learning (OpenML), 1996.
33. M. Revow. Ringnorm. Open Machine Learning (OpenML), 1996.
34. B. Antal and A. Hajdu. Diabetic retinopathy debrecen. UCI Machine Learning Repository, 2014.
35. I.-C. Yeh. Default of credit card clients. UCI Machine Learning Repository, 2009.
36. M. Xu et al. Peer: A comprehensive and multi-task benchmark for protein sequence understanding. *arXiv preprint arXiv:2206.02096*, 2022.
37. V. Kumar, A. Deepak, A. Ranjan, and A. Prakash. Lite-seqcnn: A light-weight deep cnn architecture for protein function prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(3):2242–2253, 2023.
38. P. Smialowski, A. J. Martin-Galiano, A. Mikolajka, T. Girschick, T. A. Holak, and D. Frishman. Protein solubility: sequence based prediction and experimental verification. *Bioinformatics*, 23(19):2536–2542, 12 2006.
39. P. Smialowski, G. Doose, P. Torkler, S. Kaufmann, and D. Frishman. Proso ii – a new method for protein solubility prediction. *The FEBS Journal*, 279, 2012.
40. S. Khurana et al. Deepsol: a deep learning framework for sequence-based protein solubility prediction. *Bioinformatics*, 34(15):2605–2613, 03 2018.
41. P. Baldi. Solpro: accurate sequence-based prediction of protein solubility. *Bioinformatics*, 25(17):2200–2207, 2009.
42. F. Boukid, S. Ganeshan, Y. Wang, M. Ç. Tülbek, and M. T. Nickerson. Bioengineered enzymes and precision fermentation in the food industry. *International Journal of Molecular Sciences*, 24(12), 2023.
43. B. David. Recombinant protein bioprocessing. *Pharmaceutical Bioprocessing*, 11(2):19–21, 2023.
44. C. C. H. Chang, J. Song, B. T. Tey, and R. N. Ramanan. Bioinformatics approaches for improved recombinant protein production in escherichia coli: protein solubility prediction. *Briefings in Bioinformatics*, 15(6):953–962, 08 2013.
45. H. M. Berman, J. D. Westbrook, M. J. Gabanyi, W. Tao, R. Shah, A. Kouranov, T. Schwede, K. Arnold, F. Kiefer, L. Bordoli, J. Kopp, M. Podvinec, P. Adams, L. Carter, W. Minor, R. Nair, and J. Baer. The protein structure initiative structural genomics knowledgebase. *Nucleic acids research*, 37(SUPPL. 1):D365–D368, 2009.
46. H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 01 2000.
47. S. Idicula-Thomas and P. V. Balaji. Understanding the relationship between the primary structure of proteins and its propensity to be soluble on overexpression in escherichia coli. *Protein Science*, 14(3):582–592, 2005.
48. Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, and A. Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.
49. N. Zhou et al. The cafa challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biology*, 20(24), 2019.
50. I. Kahanda, C. S. Funk, F. Ullah, K. M. Verspoor, and A. Ben-Hur. A close look at protein function prediction evaluation protocols. *GigaScience*, 4(1):s13742–015–0082–5, 09 2015.
51. A. Ranjan, A. Tiwari, and A. Deepak. A subsequence based approach to protein function prediction via multi-attention based multi-aspect network. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(1):94–105, 2023.
52. The UniProt Consortium. Uniprot: the universal protein knowledgebase in 2025. *Nucleic Acids Research*, 53(D1):D609–D614, 2024.
53. R. Joeres, D. B. Blumenthal, and O. V. Kalinina. Data splitting to avoid information leakage with datasail. *Nature Communications*, 16:3337, 2025.
54. M. AlQuraishi. Proteinnet: a standardized data set for machine learning of protein structure. *BMC Bioinformatics*, 20(1):311, 2019.
55. P. Gane et al. Uniprot: A hub for protein information. *Nucleic Acids Research*, 43:D204–D212, 11 2014.
56. M. HassanPour Zonoozi and V. Seydi. A survey on adversarial domain adaptation. *Neural Processing Letters*, 55(6):2429–2469, 2022.
57. Y. Zhang, T. Liu, M. Long, and M. I. Jordan. Bridging theory and algorithm for domain adaptation. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pages 7404–7413. PMLR, June 2019.
58. O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, 2006.
59. M. Sugiyama, M. Krauledat, and K.-R. Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8:985–1005, 2007.
60. Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, volume 17, pages 529–536, 2005.
61. D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.
62. K. Saito, D. Kim, P. Teterwak, S. Sclaroff, T. Darrell, and K. Saenko. Tune it the right way: Unsupervised validation of domain adaptation via soft neighborhood density. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9184–9193, October 2021.
63. P. Morerio, J. Cavazza, and V. Murino. Minimal-entropy correlation alignment for unsupervised deep domain adaptation. In *International Conference on Learning Representations (ICLR) 2018*, 2018. poster.
64. E. Zhong, W. Fan, Q. Yang, O. Verscheure, and J. Ren. Cross validation framework to choose amongst models and datasets for transfer learning. In *Machine Learning and Knowledge Discovery in Databases, European Conference*, pages 547–562, 09 2010.
65. S. Sagawa et al. Extending the wilds benchmark for unsupervised adaptation. In *International Conference on Learning Representations (ICLR) 2022*, 2022. Oral.

Supplementary: Hyperparameter search space for the domain adaptation techniques

Table 4. Hyperparameter search space for the validation of the domain adaptation techniques.

Technique	Hyperparameter	Search space
KMM	kernel	RBF
	gamma (kernel parameter)	0.0001, 0.001, 0.01, 0.1, 1, 10
	max. iterations	100
KLIEP	kernel	RBF
	gamma (kernel parameter)	0.001, 0.01, 0.1, 1, 10
	max. centers	50, 100, 200
	max. iterations	100
Self-training	criterion	threshold, K-best
	threshold (for threshold criterion)	0.5, 0.7, 0.85
	K (for criterion K-best)	10, 20, 30, 50, 100
	max. iterations	100
Co-training	unlabeled pool size	5, 10, 20, 50
	max. iterations	100
SA	nr. components	25, 50, 75 % of nr. features
TCA	nr. components	25, 50, 75 % of nr. features
	mu (regularization parameter)	0.001, 0.01, 0.1, 1
	kernel	RBF
	gamma (kernel parameter)	0.001, 0.01, 0.1, 1, 10
RBA	L2 regularization	0.001, 0.01, 0.1]
	gamma (decaying learning rate)	0.01, 0.1, 1, 10
	max. iterations	100
TCPR	L2 regularization	0.001, 0.01, 0.1
	learning rate	0.01, 0.1, 1, 10
	max. iterations	100
DANN	lambda (trade-off parameter)	0.01, 0.1, 1, 10
WDGRL	lambda (trade-off parameter)	0.01, 0.1, 1, 10
	gamma (gradient penalization parameter)	0.1, 1, 10
MDD	lambda (trade-off parameter)	0.01, 0.1, 1, 10
	gamma (margin parameter)	0.1, 1, 4, 10

Supplementary: Original scores for the controlled experiments

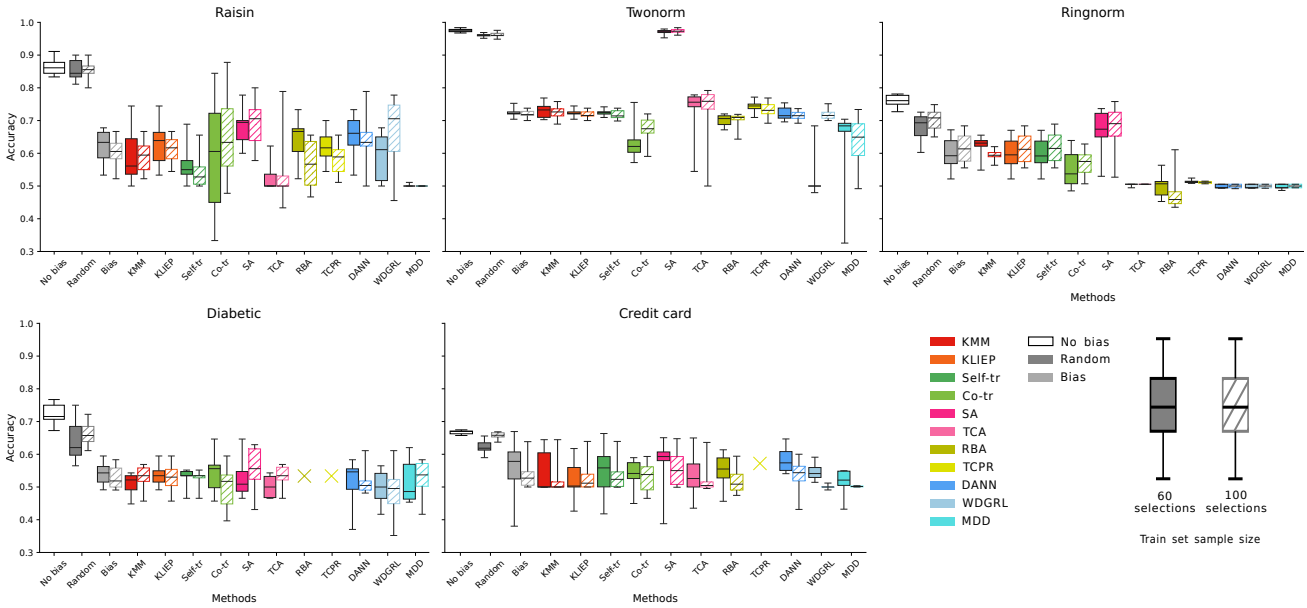

Fig. 7. Performance of the domain adaptation techniques for different sample sizes of the train set in the controlled experiments. Results obtained across 10 runs, with all methods evaluated using the same folds (train/test/global splits). Methods included: 11 adaptation techniques, supervised model trained on unbiased data (No bias), on biased selection of data without adaptation (Bias), and on randomly selected samples (Random, same number as Bias). Adaptation techniques that did not manage to run are marked with X.

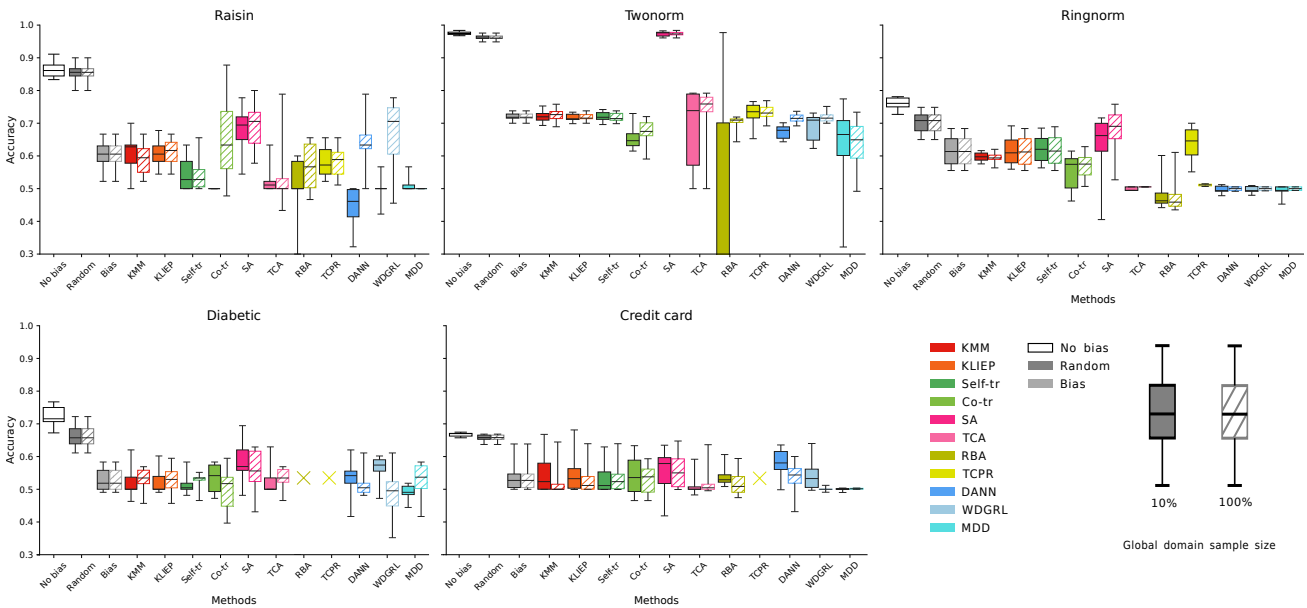

Fig. 8. Performance of the domain adaptation techniques for different sample sizes of the unlabeled global domain in the controlled experiments. Results obtained across 10 runs, with all methods evaluated using the same folds (train/test/global splits). Methods included: 11 adaptation techniques, supervised model trained on unbiased data (No bias), on biased selection of data without adaptation (Bias), and on randomly selected samples (Random, same number as Bias). Adaptation techniques that did not manage to run are marked with X.

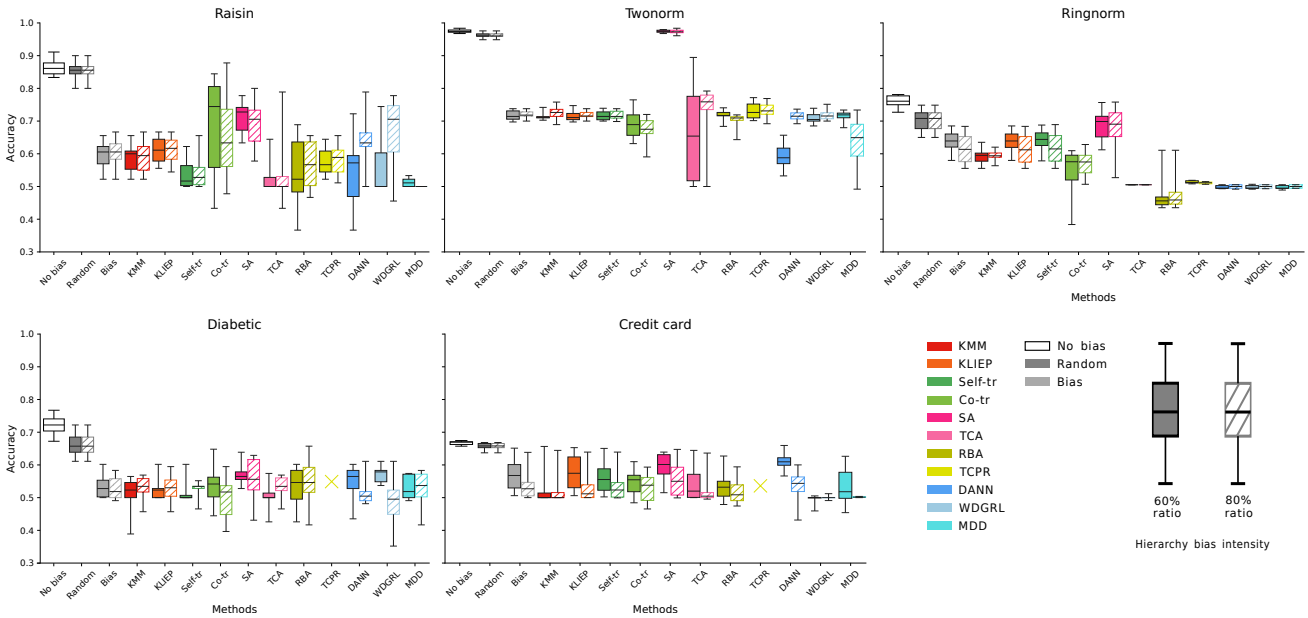

Fig. 9. Performance of the domain adaptation techniques for different hierarchy bias intensities in the controlled experiments. Results obtained across 10 runs, with all methods evaluated using the same folds (train/test/global splits). Methods included: 11 adaptation techniques, supervised model trained on unbiased data (No bias), on biased selection of data without adaptation (Bias), and on randomly selected samples (Random, same number as Bias). Adaptation techniques that did not manage to run are marked with X.

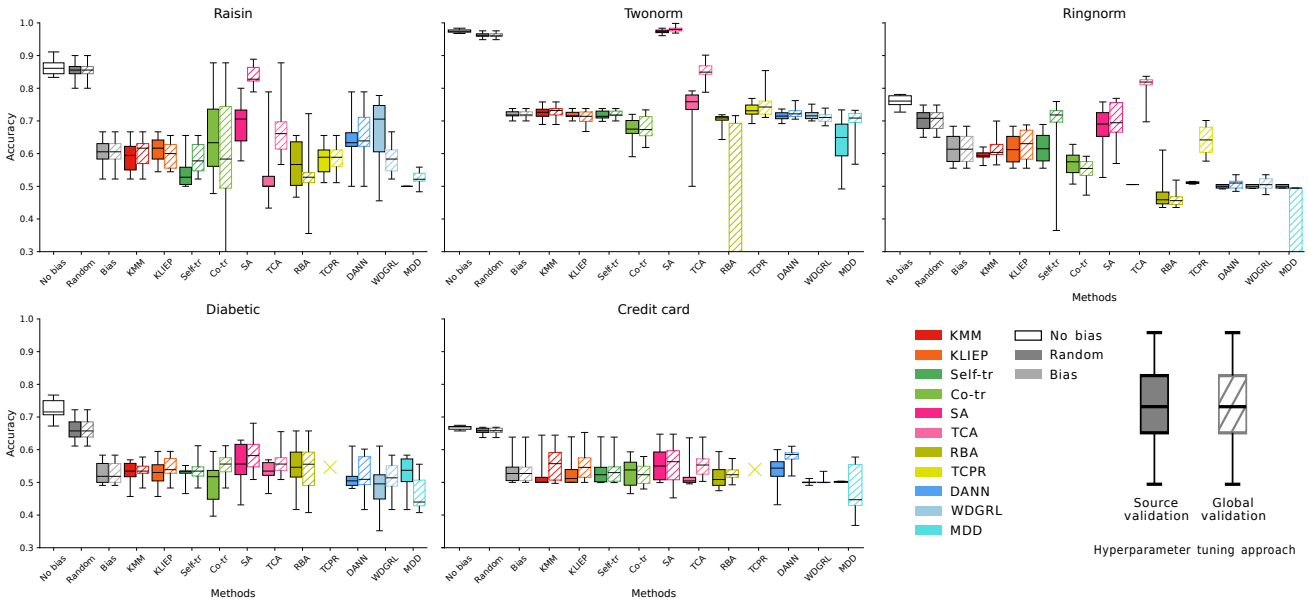

Fig. 10. Performance of the domain adaptation techniques for different hyperparameter tuning approaches in the controlled experiments. Results obtained across 10 runs, with all methods evaluated using the same folds (train/test/global splits). Methods included: 11 adaptation techniques, supervised model trained on unbiased data (No bias), on biased selection of data without adaptation (Bias), and on randomly selected samples (Random, same number as Bias). Adaptation techniques that did not manage to run are marked with X.

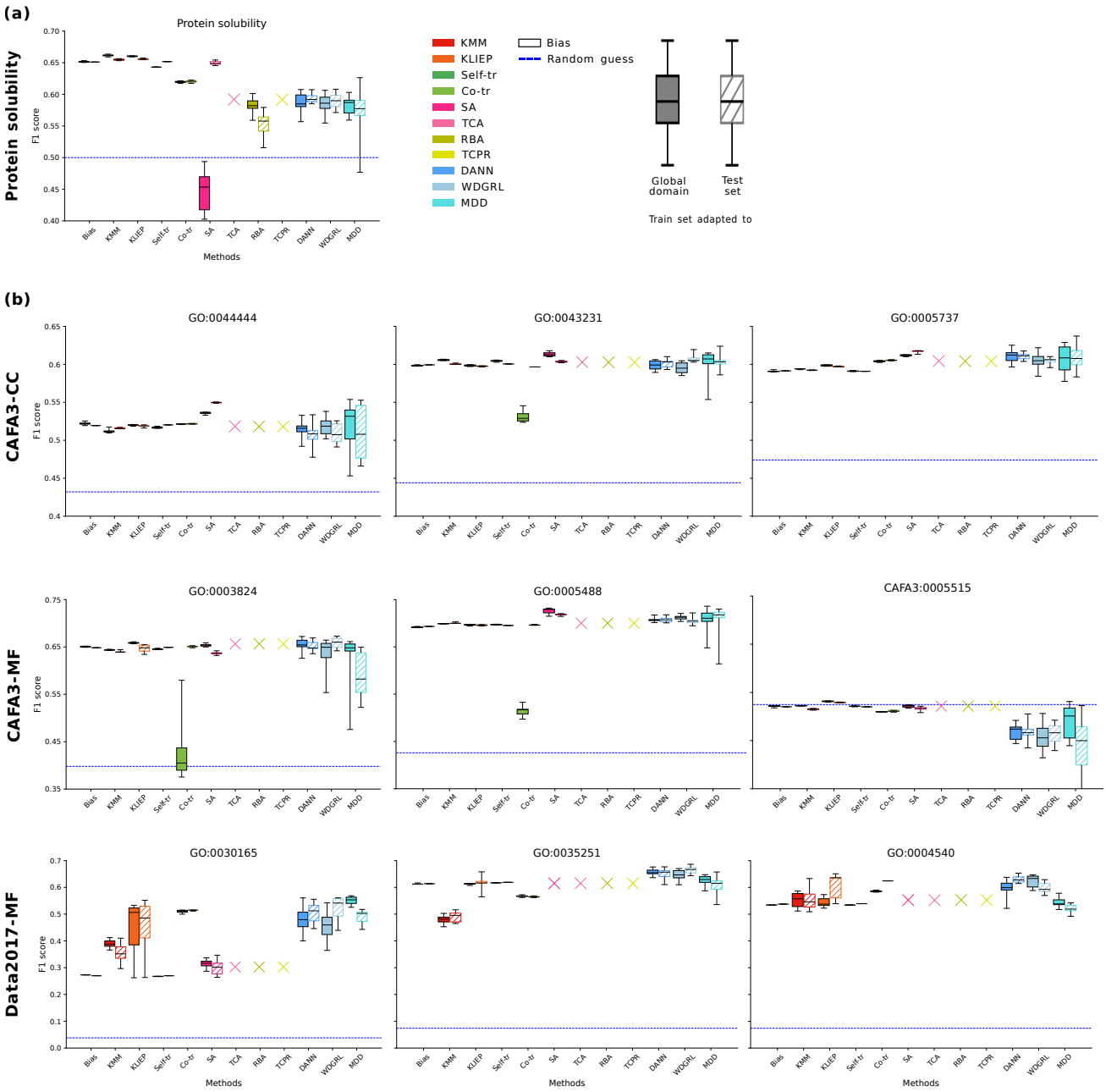

Fig. 11. Performance of the domain adaptation techniques on bioinformatics problems with naturally biased data: (a) protein solubility prediction, and **(b)** prediction of 9 Gene Ontology(GO) terms across 3 datasets. Results obtained across 10 runs, with all methods evaluated using the same train/test/global datasets. Methods included: 11 adaptation techniques and supervised model trained and evaluated on the biased data without domain adaptation (Bias). Adaptation techniques that did not manage to run are marked with X.