Diverse Self-Training Via Metric Learning

Mitigating selection bias in synthetic lethality prediction using metric learning

Mathijs de Wolf

Delft University of Technology

TU Delft

Diverse Self-Training Via Metric Learning

Mitigating selection bias in synthetic lethality prediction using metric learning

by

Mathijs de Wolf

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday, June 23, 2023, at 10:00 AM.

Student number:	4707133	
Master's programme:	Computer Science, Bioinformatics specialization	
Faculty:	Electrical Engineering, Mathematics and Computer Science	
Project duration:	May, 2022 – June, 2023	
Thesis committee:	Dr. Joana Gonçalves,	TU Delft, supervisor
	Dr. Pradeep Murukanaiah,	TU Delft
	MSc. Yasin Tepeli,	TU Delft, daily supervisor

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

TU Delft

Preface

Ever since I decided to pursue my studies in computer science, I have been amazed by all the applications and possibilities it has. When I got the choice to combine the field of computer science with the field of biology, another interest of mine, I knew it would be the right choice. Over the past few years, I have learned not only how to develop and use computer science models but also how the human body works and is able to adapt. It has been interesting to learn how much is already known about the inner workings of individual cells and how the field of bioinformatics contributes to improving our understanding of the world.

The last two years have not been easy. In the aftermath of covid, I struggled a lot with finding the motivation, drive and energy to finish my Master's degree. It has been a series of ups and downs, but now that the end is near, I'm excited to see what the future holds. Completing my Masters's has not been possible without the support of all the people around me.

Firstly I would like to thank Joana Gonçalves for all her support, guidance and feedback during the thesis project. It has been inspiring to work around such a driven person, and I value all the insights she has given. I would also like to express my gratitude to Yasin Tepeli, my daily supervisor, for being available whenever I had any questions during these last two years. I would also like to thank Pradeep Murukanaiah for his interest in my project and for agreeing to take place in my defence committee. Additionally, I would like to thank everyone from the Gonçalves Lab for all the interesting meetings and lunch discussions. Lastly, I would like to thank my parents, my brother, and the rest of my family for the unconditional love they have shared with me over all these years.

Mathijs de Wolf
Delft, June 2023

Mitigating selection bias in synthetic lethality prediction using metric learning

Mathijs de Wolf,[1] Yasin Tepeli[1] and Joana Gonçalves[1,*]

[1]Pattern Recognition and Bioinformatics, Intelligent Systems Dept., EEMCS Faculty, Delft University of Technology, Netherlands
[*]Corresponding author. Joana.Goncalves@tudelft.nl

Abstract

Synthetic lethality (SL) is a relationship between two genes, exploited for targeted anti-cancer therapy, whereby functional loss of both genes induces cell death, but the functional loss of either gene alone is non-lethal. Computational prediction of SL gene pairs is sought after because it is expensive to do lab screening for SL. Existing SL labeled pairs from wet-lab experiments often focus on specific genes or pathways, resulting in notable selection bias. Current SL prediction methods ignore this bias when training on available SL labels, and fail to generalize if test sets follow a different selection bias. One way to mitigate bias is to incorporate unlabeled pairs during model learning. However, conventional semi-supervised methods such as self-training can reinforce bias by adding confidently pseudolabeled pairs, which tend to be most similar to previously included samples. We present DBST, a self-training strategy that addresses the issue by promoting diversity in the selection of pseudolabeled samples. This is achieved using metric learning to find a class-contrastive representation of the feature space, based on which DBST selects diverse (or dissimilar) pseudolabeled pairs. In results for five cancer types, semi-supervised models, including DBST, delivered improved SL prediction performance over the supervised model. Additionally, DBST successfully incorporated unlabeled samples that were more dissimilar among them compared to standard self-training. In experiments with differing biases between train and test sets, DBST showed a slight improvement in performance compared to the supervised model.

Introduction

Synthetic lethality (SL) is a relationship between two genes where the functional loss of either gene alone is not harmful to the cells, but the functional loss of both genes causes the cells to die. It is an active area of research in the quest to find targeted cancer treatment. Specific gene functions are disrupted in tumor cells due to direct deleterious mutation or indirect alterations. Targeted drug therapies can target synthetic lethal genes so that the mutual inhibition of both genes in cancer cells causes cell death while healthy cells remain viable (1; 2). The most prominent example is the SL interaction between the PARP and BRCA genes. Specific PARP-inhibitor therapy for cancers with deleterious mutations in the breast cancer susceptibility genes BRCA1 and BRCA2 was created (3; 4), tested in clinical trials (5), and later approved by the FDA and EMA (6). Since then, many more targeted therapies that leverage SL interactions have been researched with applications in different cancer types (7).

Finding robust SL gene pairs is an extensive process, as humans have between 20.000 to 25.000 protein-coding genes (8), making the number of possible gene pairs extremely large. Besides that, it is believed that only a small number of all possible gene pairs have a synthetic lethal interaction. As testing every gene combination for SL in wet-lab experiments is nearly impossible, computational approaches are being developed to find potentially interesting gene pairs for experimental screening.

There are various methods to predict new synthetic lethal interactions. These methods include statistical-based methods, network-based methods, and classic feature-based machine learning methods (9).

Statistical-based methods form hypotheses on gene pairs having statistical properties of being synthetic lethal. One example is the DAISY method (10), which forms three statistical tests based on the assumptions that SL gene pairs are not coinactivated, individual genes in SL pairs are important, and SL gene pairs are often coexpressed. The advantage is that statistical-based methods do not need pre-existing SL data to form predictions, and the models are more comprehensible to biologists (9). However, statistical-based methods often lack predictive power.

Network-based models exploit various biological networks, such as PPI networks, signaling networks, metabolic networks, and existing SL networks, to find a structure indicating synthetic lethality. A disadvantage, however, is that these networks are largely incomplete, making these models less effective. Another disadvantage is that existing SL labels contain selection bias (11; 12), and network-based models usually exploit and follow the bias present in the data. Wet-lab experiments often focus on a subset of genes, which causes the available SL data to lean towards specific genes rather than the data being uniformly sampled and representing the true distribution. This selection in what data point we have labels for is called selection bias (13). It has been shown that especially network-based models are sensitive to this selection bias (11).

Machine learning methods typically include supervised learning methods where models are trained based on pre-existing SL data as labels and other biological data as features. Examples of this are an SVM-based model that uses PPI networks to predict SL (14) and random forest-based models DiscoverSL, SBSL, and ELISL that predict SL interaction based on multiomics cancer data (15; 11; 12). However, many more

gene pairs are available without their respective SL label, which cannot be used in a supervised setting.

We can notably leverage unlabeled gene pairs in the training process by using semi-supervised learning (16). The unlabeled samples allow us to learn the underlying distribution over the entire space besides the known SL data that carries the selection bias. There are already SL prediction models that make use of semi-supervised learning. One example is Exp2SL (17), which can use cell-line-specific gene expression profiles where it can use a loss function that not only measures the error for labeled gene pairs but also tries to increase the margin between labeled and unlabeled samples.

A common framework for semi-supervised learning is self-training: a model is progressively built using supervised learning, initially based only on labeled samples; this model is then subsequently retrained or refined by pseudolabeling (or predicting) unlabeled samples, some of which are selected and incorporated into further training based on the confidence of pseudolabeling by the model (16; 18). However, self-training can follow and even reinforce the bias already present in the data (19) because self-training suffers from confirmation bias (20). In selecting the highest confident samples self-training is more likely to select samples similar to those already present in the training data, which causes self-training to lean more towards the biased data. To use self-training without reinforcing the bias, one must more carefully consider how pseudosamples are selected to prevent the selection of similar samples as the samples in the training data (20; 21).

To mitigate selection bias, many ideas have been proposed. One example of these ideas is aligning the predicted unlabeled distribution to the labeled distribution (22). Here they assume that the true distribution can be extrapolated from the labeled samples, a problem they do not address further. Another method is to try and reconstruct the missing data by introducing artificial data points (23). By comparing where the artificial data points are created, you can measure if your data suffers from bias and use the artificial points as a complete dataset. The downside of this method is the use of synthetic data which might not exist of follow a different bias. A third method combines active learning with semi-supervised learning (24). They try to mitigate the bias by leveraging active learning to find informative unlabeled samples and then pseudolabel those informative samples in an SSL approach. The downside is that informative unlabeled samples might not be predicted well by the SSL method.

In this paper, we propose a model that combines metric learning with a known semi-supervised learning method called self-training to promote diversity in the selection of pseudosamples. Semi-supervised approaches often focus on adding high-confidence pseudosamples and are, therefore, likely to select similar samples. This results in models reinforcing the selection bias present in the data. We propose DBST, a method that can select diverse pseudosamples that are less similar to labeled SL samples. The aim is that the selected diverse samples allow the model to generalize better over the actual distribution and not follow the bias present in labeled SL data. This is achieved using metric learning to find a class-contrastive representation of the feature space, based on which DBST selects pseudolabeled samples.

Methods

Our aim is to learn a model that can use unlabeled data in SL prediction to mitigate sample selection bias by selecting diverse samples.

Concept

To protect the class balance and prevent following further bias, we use metric learning in combination with self-training while protecting the class balance and adding diverse confident samples. Metric learning allows us to learn an embedding space where the distance between samples indicates their dissimilarity (25). This embedding space creates the opportunity to find diverse samples based on the distances between them.

Definitions

We define the input matrix of featurized representations of gene pairs as $\boldsymbol{X} \in \mathbb{R}^{n \times m}$, where n is the number of samples (or gene pairs) and m is the number of features. We define a vector containing the SL labels of all samples (gene pairs) as $\boldsymbol{y} \in \{0,1\}^n$, where 1 indicates a synthetic lethal relation between genes of the pair and 0 indicates no synthetic lethal relation. We further define the embedding matrix as $\boldsymbol{Z} \in [0,1]^{n \times q}$, where q is the dimension of each sample embedding vector.

Problem formulation

We aim to learn a model that can predict SL labels $\boldsymbol{y}$ for gene pairs in $\boldsymbol{X}$ while incorporating both labeled and unlabeled data, and where included unlabeled samples are chosen to be diverse among each other. For this purpose, we would like to learn a non-linear transformation $f_\theta : \boldsymbol{X} \rightarrow \boldsymbol{Z}$, where θ represents the model weights, such that samples with identical SL class label are closer to each other in the embedding space while samples with distinct SL class labels are further apart. Once this model is learned, it can be used to (i) predict the SL label $\hat{y}$ for an unknown sample based on the proximity to known samples in the latent space, and (ii) use a distance metric in the embedding space to inform the selection of diverse unlabeled samples to incorporate in the semi-supervised self-training process, with the goal of mitigating the effect of selection bias.

Solution

The overview of our model can be found in Figure 1. The first step is to train a transformation f_θ based on the labeled samples $\boldsymbol{X}_L$ and their SL labels $\boldsymbol{y}_L$ using metric learning with a neural network. This trained model provides the transformation f_θ from the original feature matrix $\boldsymbol{X}_L$ to its embedding matrix $\boldsymbol{Z}_L$ with corresponding labels $\boldsymbol{y}_L$, where samples from the same class are closer together in the embedding space, and samples from different classes are farther apart. The same transformation model f_θ trained on the labeled samples is used to transform the unlabeled samples $\boldsymbol{X}_U$ into their corresponding embeddings $\boldsymbol{Z}_U$. For each unlabeled sample, a (pseudo)label is predicted by looking at the labels of its k nearest neighbors in the latent space from the labeled set. After pseudolabeling, a final selection is made to choose which pseudolabeled samples to add to the labeled set for a subsequent iteration of the training procedure.

Metric learning

To learn the transformation f_θ, we use metric learning. Metric learning is a method where the prediction task is not optimized directly. Instead, we learn a transformation into an embedding

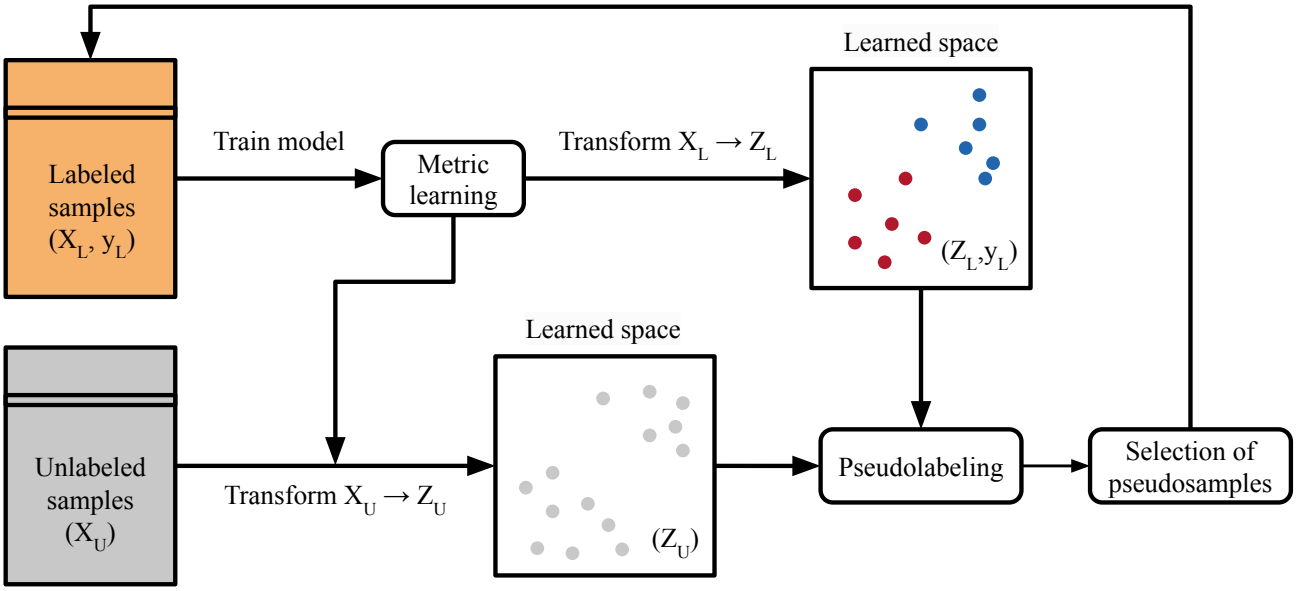

Fig. 1. Overview of our proposed method

space based on the classes of samples and the distances between them. The idea is that in the embedding space, samples from the same class are close to one another, while samples from different classes have greater distances between them. In other words, metric learning is learning an embedding function such that we can say something about the dissimilarity of the samples based on the Euclidean distance between those samples in the resulting embedding space (26). Metric learning relies on three parts: informative input samples, an underlying neural network architecture, and a loss function (25). All three can impact the discriminating power and convergence of the model.

The neural network used for metric learning to transform the original feature representations into class-contrastive embeddings is a feedforward neural network with one 8-node hidden layer and a 2-node output layer. After each layer, a sigmoid activation function is used, which makes the neural network a non-linear transformation. The network is trained in batches with an Adam optimizer to guide the learning rate.

As for the loss function, we opted for the contrastive loss because it is easier to understand and because this is a binary classification problem (either SL or non-SL). The contrastive loss increases when samples of the same class are not within a positive margin m_{pos} or when the distance between different classes is not greater than a negative margin m_{neg}. This comparison is made between all possible combinations of two samples within a batch. The losses for each batch are then accumulated, considering each batch's size. The final loss value is then used to update the model weights. The contrastive loss is defined as per Eq. 1:

$$\mathcal{L}_{contrastive} = \sum_{(i,j)\in P} \mathbb{1}_{y_i=y_j}[d_{i,j} - m_{pos}]_+ + \mathbb{1}_{y_i\neq y_j}[m_{neg} - d_{i,j}]_+, \quad (1)$$

where i and j are samples, P is the set with all selected samples, which in our case is every combination between samples in the same batch. Additionally, $\mathbb{1}_{condition}$ is an indicator function which equals 1 if the condition is true and 0 if the condition is false, and m_{pos} and m_{neg} are the positive margin and negative margin, respectively. Moreover, $d_{i,j}$ denotes the distance between samples i and j, which is shorthand for $d(z_i, z_j)$ and corresponds to the Euclidean distance between samples i and j in the embedding space.

Predicting the label (and pseudolabeling)

To infer the class label of each unlabeled sample, we can utilize the k nearest neighbors (kNN) training samples of the unlabeled samples in the learnt latent space $\boldsymbol{Z}$. This is done by taking an unlabeled sample i, transforming its original representation x_i into its embedding vector z_i, and then using the labeled samples as a reference to find the k (10) labeled samples that are closest to sample i in the embedding space. Based on the labels of those nearest neighbors, a class label is predicted for an unlabeled sample i. To reward the proximity of the k nearest neighbors to sample i and penalize distance, a weighted kNN model is used where the contribution of the label of each nearest neighbor k is inversely proportional to the distance between the unlabeled sample i and the nearest neighbor. The weighted average over the k nearest neighbors forms an SL score as per Eq. 2:

$$SLscore(i) = \frac{\sum_k y_k \times (1 - d_{i,k}) + (1 - y_k) \times d_{i,k}}{|k|}. \quad (2)$$

This SL score is in the interval $[0, 1]$. The closer the score is to 1, the more confidence that the sample is positive and, therefore, synthetic lethal. The closer the score is to 0, the more confidence that the sample is negative and not synthetic lethal. If the SL score is at 0.5, we are indifferent to whether the sample is synthetic lethal. To come to a final classification $\hat{y}$, we use 0.5 as a threshold as per Eq. 3:

$$\hat{y}_i = \begin{cases} 1, & \text{if } SLscore(i) > 0.5 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Since the confidence of a sample depends on how close the SL score is to either 0 or 1 we define the confidence of a sample as follows: a sample i has a confidence of at least c if $SLscore(i) \geq c$ or $SLscore(i) \leq 1 - c$.

Selection of pseudolabeled samples

After the pseudolabeling of unlabeled samples, a selection is made of which pseudolabeled samples to include for the next training iteration. For a given training iteration t, we first train the model on the labeled set L_t. At the end of iteration t, we select a subset of pseudosamples $U_{ts} \in U_t$ to add to the labeled set for the next iteration $L_{t+1} = L_t \cup U_{ts}$. The selected pseudosamples U_{ts} are then also removed from the unlabeled set U_t for the next iteration such that $U_{t+1} = U_t \setminus U_{ts}$. At every iteration t, we add a total of p pseudosamples using one of three different strategies: standard self-training (ST), balanced self-training (BST) and diverse balanced self-training (DBST).

Standard self-training

The ST approach is a semi-supervised method of incorporating unlabeled samples into the model learning and prediction task (18). To select the pseudosamples, all unlabeled samples are ranked based on their confidence. The p highest confident samples are then selected as pseudosamples in each iteration. This selection is independent of the class label, so after the first iteration, there is no longer a guarantee on the class balance in the labeled set.

Balanced self-training

The BST strategy works similarly to ST but considers the prediction probability of the pseudosamples. The unlabeled samples are therefore not ranked based on their confidence but instead based on $SLscore$. Half of the pseudosamples $p/2$ come from the highest-scoring samples, and the other $p/2$ pseudosamples come from the lowest-scoring samples. This method makes it more likely that the class balance is maintained throughout all training iterations, as the upper and lower halves are more likely populated by predicted positives and negatives, respectively.

Diverse and balanced self-training

The DBST strategy is fundamentally different from the ST and BST approaches described above in that it uses the metric learning embeddings for the selection of pseudolabeled samples. The idea is to pick random pseudosamples in the embedding space that are above a certain confidence threshold μ. By picking random samples, the goal is to select samples that are different from each other and do not follow the same inherent bias in the SL labels. In order to select a random pseudosample, we generate a random real number between 0 and 1 for each dimension or coordinate of the embedding space. This vector of random coordinates represents a random data point in the embedding space. Using the Euclidean distance, we find the closest unlabeled sample from this random data point in the same embedding space. The closest unlabeled sample is our candidate c. If the pseudolabeling confidence of candidate c exceeds the confidence threshold μ, it is selected to be included as a pseudolabeled sample for the next iteration. Finding samples is done until we have selected $p/2$ positives and $p/2$ negatives or if a given maximum number of iterations $(50 \times p)$ is reached. If the selection procedure is stopped prematurely, the already found pseudosamples are balanced by undersampling the majority class to ensure the same number of positives and negatives in the labeled set.

Data

To evaluate the proposed self-training framework and its ST, BST, and DBST variants, we used both SL labeled and unlabeled sets of gene pairs, together with their feature vector representations, for multiple cancer types.

Table 1. Numbers of synthetic lethality labeled and unlabeled samples or gene pairs per cancer type.

Cancer	Total	Positives	Negatives	Unlabeled
BRCA	2453	1443	1010	151888
OV	805	253	552	151972
CESC	4900	144	4756	150964
SKCM	18407	107	18300	151545
LUAD	6103	594	5509	150944

Synthetic lethality labels and gene pairs

In the context of the prediction task, each sample corresponds to a gene pair with a label denoting if the pair is synthetic lethal or not. We used a set of SL labeled gene pairs combining data from four studies: Exp2SL (17), DiscoverSL (dSL) (15), Lu et al. (27), and ISLE (28). The SL labels were aggregated per cancer type, removing gene pairs with different labels in 2 or more studies. We opted to use data for five different cancer types: breast (BRCA), ovarian (OV), cervix (CESC), skin (SKCM), and lung (LUAD). BRCA was selected because it contains a lot of labeled samples and is the most complete. OV was selected for its heavy selection bias. CESC was used because it does not have many samples in general. SKCM was interesting because it has few samples and a heavy selection bias. And finally, LUAD was selected because it also has many labeled samples. The numbers of labeled samples or gene pairs per cancer type can be found in Table 1.

Besides the SL labeled gene pairs, we also made use of a set of unlabeled samples made available by Tepeli et al. (12). The pairs were obtained by generating all pairwise combinations of 572 genes present in cancer and DNA repair pathways from KEGG, Reactome, and PID. From these pairwise combinations, pairs present in the labeled set were removed. The final numbers of unlabeled samples per cancer type can be found in Table 1

Features of gene pairs

For each gene pair, we obtained a sequence-based embedding considering the amino acid sequences of the corresponding proteins as a proxy for functional relatedness (12). For each gene, the amino acid sequence context was first encoded using the SeqVec method (29) to create an embedding of 1024 dimensions. Principal component analysis was then applied to reduce the embedding dimension to the 128 most impactful principal components. To create the final feature vector, the embeddings of the individual genes were combined by taking the absolute value of the element-wise difference. This resulted in a feature of 128 dimensions for each gene pair.

Experimental setup

To evaluate the proposed self-training framework and its variants, we performed three different experiments. The first experiment investigated the ability of the proposed models to predict SL. The final two experiments examined the ability of the proposed models to mitigate selection bias in training data.

All three experiments were performed for ten runs. At the start of each run, the train, validation and test sets were balanced by random undersampling of the majority class. The model was trained on the train set until the validation loss did not decrease for five consecutive rounds. The final performance

was measured on the test set. How the train, validation and test sets were constructed differs between the three experiments.

Randomized split

In the first experiment, the data was split randomly into test, train, and validation sets. This data partitioning resulted in a standard experiment where the test and training sets came from the same distribution and therefore had the same bias. This meant that this experiment was not informative for the question about mitigating the bias. However, this partitioning did allow us to evaluate the performance of the proposed models on predicting SL without yet setting additional constraints.

The test set was created by taking a split of 20% stratified by the SL class label. At the start of each run, the remaining data was split further, with the validation set as a stratified split of 20% and the remaining 80% as the train set.

Double holdout

To assess the performance of the proposed methods when the train and test set follow different biases, we performed an experiment where the gene pairs in the train and test sets did not have any genes in common. By decoupling the genes in the train set from the test set, we constructed an experiment where the two sets do not originate from the same distribution and do not follow the same sample selection bias. In this experiment, we could evaluate the ability of the methods to transfer knowledge learned on one distribution to data with a different bias. We divided the set of all individual genes into two sets, a test and a train set. Then all pairwise combinations of genes were generated within each set, and finally, we kept only the pairs for which we had labels in each set. This separation ensured that there was no overlap between the two sets of gene pairs.

To divide the data, we iteratively selected a random gene and added this gene to either the train or test set, depending on the number of labeled samples in each set. If there were more than four times as many labeled gene pairs in the train set, the randomly selected gene was added to the test set, and if there were less than four times as many labeled gene pairs in the train set, the randomly selected gene was added to the train set. In the end, we tested whether the resulting train and test set contained enough labeled samples according to a threshold (BRCA: 280, 100; OV: 85, 20; CESC: 55, 12; LUAD: 200, 50, the sizes of train and test set per cancer, respectively). We continued this process until we had ten different partitions for each cancer.

The SKCM dataset was particularly challenging to split, as most available samples or gene pairs contained the gene MYC. This gene was so dominant that only 60 samples did not contain the gene. Therefore, we split the data for this cancer type differently: we took the 60 samples that did not contain the gene MYC as one set and removed all samples from the other set that had any gene overlap with these 60 samples.

Multiple SL label sources

To further investigate the performance of our proposed methods when the train and test set do not follow the same bias, we performed a second experiment. In this experiment, we constructed the train set from the SL labels of one study and the test set based on the SL labels of another study. By using two different datasets as the train and test set, the idea is that we are more accurately measuring the capabilities of the models to mitigate selection bias. The two datasets will likely have a different selection bias, so this experiment is less strict than the double holdout.

For this experiment, we used three different partitions, namely:

1. The ISLE dataset and the dSL dataset on BRCA cancer.
2. The ISLE dataset and the dSL dataset on LUAD cancer.
3. The dSL dataset and the Exp2SL dataset on LUAD cancer.

For each combination of studies, the experiment was performed twice, with one of the datasets as the train set and the other as the test set and vice-versa, resulting in six experiments overall.

For each experiment, it was possible that the datasets from the two studies shared gene pairs between them. To guarantee an independent test set, we excluded the shared gene pairs from the test set.

Evaluation

To evaluate the models, we measured their performance over ten runs on the test set using the area under the precision-recall curve (AUPRC), calculated as the average precision. The confidences for the test samples were calculated as stated in the section *Predicting the label* by using the training samples as a reference (including pseudolabeled samples), and treating the test samples as the unlabeled samples. With the confidence of the test samples as well as their actual labels, we calculated the AUPRC.

To assess the significance of the results, we used the non-parametric two-sided Wilcoxon signed ranked test with a p-value threshold of 0.05. We tested for the significance of the change in performance between each semi-supervised method and a supervised learning baseline. To account for model variation in these comparisons, we ran all experiments for ten runs with differently undersampled train, validation, and test sets. If the p-value is lower than the 0.05 threshold, we consider that the results of the two models show a different distribution, which can either mean an increase or decrease in performance. If the p-value is at least 0.05, we can not conclusively determine if it is a significant performance difference.

Hyperparameter selection

To select the hyperparameters p (number of pseudosamples each iteration) and μ (confidence threshold), we performed a grid search for each experiment and dataset. The settings with the lowest validation loss were selected for the model. The used parameters are described in the Supplementary Figures 2, 3, 4. For the other model parameters, we initially tested different configurations, but as they are described above is what seemed to work best for this problem. These parameters include the size of the neural network, the number of output dimensions, and the positive and negative margins.

Results and discussion

To evaluate the proposed model, we focus on two main questions. The first question is, "What is the effect of incorporating unlabeled samples in metric learning for synthetic lethality prediction?". This question focuses mainly on the performance of the metric learning model and whether metric learning as a concept is suitable for SL prediction. The second question is, "How does metric learning perform to mitigate bias?". This question focuses on whether our proposed DBST method can mitigate bias in SL prediction.

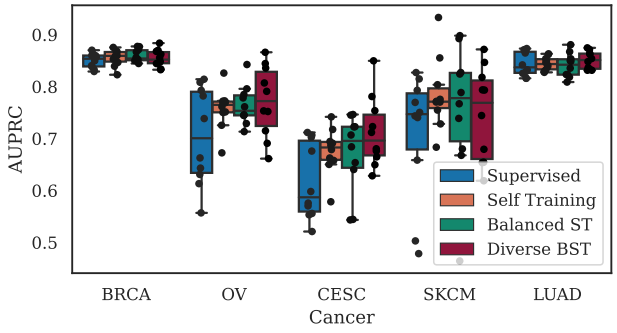

Fig. 2. Performance of the random split experiment across ten iterations. The experiments include five different cancer types (BRCA, OV, CESC, SKCM, LUAD) and are highlighted according to the four different methods (supervised, ST, BST and DBST). The dots represent the AUPRC of an individual run.

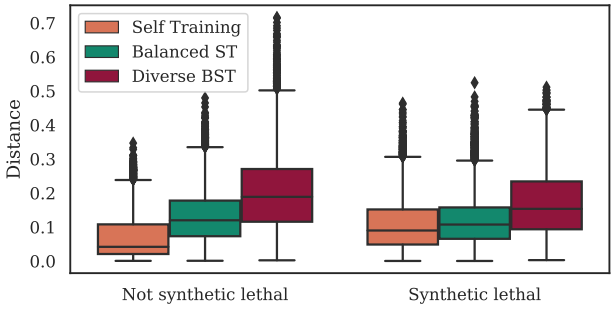

Fig. 3. Euclidean distances in the latent space between synthetic lethal and between not synthetic lethal pseudosamples added during training. These are the results from the first iteration of training on the BRCA dataset in the randomized split experiment.

What is the effect of incorporating unlabeled samples in metric learning for synthetic lethality prediction

We performed the randomized split experiment described in the methods to quantify the effect of incorporating unlabeled samples in metric learning. We first trained a supervised model using only labeled samples. Then we also experimented with the three semi-supervised methods ST, BST and DBST, each incorporating unlabeled samples differently (see Methods).

We compared the supervised and semi-supervised methods on five different cancer types. The median performance of the baseline supervised model for the BRCA and LUAD cancer types was relatively high (0.854 and 0.837, respectively). The three semi-supervised methods using different pseudosample selection approaches did not improve performance over the baseline (median AUPRC BRCA ST 0.859, BST 0.854, DBST 0.852; and LUAD ST 0.843, BST 0.842, DBST 0.851) significantly (p-values BRCA ST 0.375, BST 0.233, DBST 0.557; and LUAD ST 0.770, BST 0.557, DBST 0.160) (Fig. 2). An explanation for this behaviour might be that the starting performance of the supervised method was already high and that it would be difficult to achieve further improvement by incorporating unlabeled samples. We have possibly reached the maximum performance that could be obtained based on the sequence data we used, because BRCA and LUAD were the two cancer types with the largest datasets. To further improve prediction, we might need additional features such as PPI networks, as it has been suggested that pairs of genes of robust SL interactions tend to be closely connected in PPI networks (30). By incorporating more different types of data related to the prediction task, we will be able to increase the discriminating power of the samples to better differentiate between SL and non-SL gene pairs.

The AUPRC of the semi-supervised methods on the SKCM dataset (median AUPRC ST 0.771, BST 0.778, DBST 0.769) did not improve the predictive performance over the supervised approach (median AUPRC 0.747) significantly (p-values ST 0.105, BST 0.131, DBST 0.432). The performance on ovarian cancer (OV) showed an improvement over the supervised model (median AUPRC 0.701) for the three different methods of pseudolabel selection (median AUPRC ST 0.765, BST 0.753, DBST 0.772). However, these improvements were not significant across the ten runs (p-values ST 0.232, BST 0.232, DBST 0.232). Similarly, the semi-supervised methods (median AUPRC ST 0.683, BST 0.695, DBST 0.696) performed better than the supervised model for CESC (median AUPRC 0.587). For ST and BST, the improvement was not significant (p-values ST 0.131, BST 0.131), but for DBST, the improvement was significant (p-value 0.014). The labeled sets for these two cancer types were more limited in size compared to BRCA and LUAD, especially after balancing the classes. That is why there might be more room for improvement due to the supervised approach needing more informative training samples to make an accurate prediction, which the semi-supervised models provide.

In these experiments, the train and test set follow the same bias. Therefore, even if the methods mitigate the bias, there might be no performance improvement. Since the train set has the same distribution as the test set, it would be sufficient to learn a model that is predictive of the train set. One thing to note, however, is that the semi-supervised methods at least do not perform worse than the supervised model. This indicates that the semi-supervised models perform as well or better than the supervised model, and that the expected benefit of the models in mitigating the bias is not tested.

Another noteworthy observation is that there was no real performance difference between the diversity approach (DBST) and the other two semi-supervised methods (ST and BST), and between the standard self-training (ST) and the balanced self-training (BST) methods. Balancing the classes in BST compared to not balancing the classes in ST had no significant effect. One explanation for this is that metric learning is robust against class imbalance. Since the contrastive loss compares samples in pairs, an excess of samples for one class will not lead the model to prefer that class. That metric learning performs better on class imbalanced data is also supported in literature from the contrastive loss (31) to custom loss functions that perform better on long-tailed class distributions (32; 33), these findings were reported on multi-class classification. However, the class imbalance in ST in our experiments was not very large (BRCA 55% $\pm$ 2 for the majority class, Supplementary Table 5). It has also been shown that as the imbalance drastically increases, the performance of the minority class decreases for metric learning in binary classification (34).

To examine the differences between the three semi-supervised methods and see whether the diversity approach adds diversity to the samples, we plotted the distances between similarly labeled selected pseudosamples in the final embedding space (Fig. 3). We grouped all selected pseudosamples that were added during training based on their pseudolabel (not synthetic lethal or synthetic lethal). Then we measured the distance between every pair of sample in the embedding space.

The median distance between non-SL samples and the median distance between SL samples was higher in the DBST

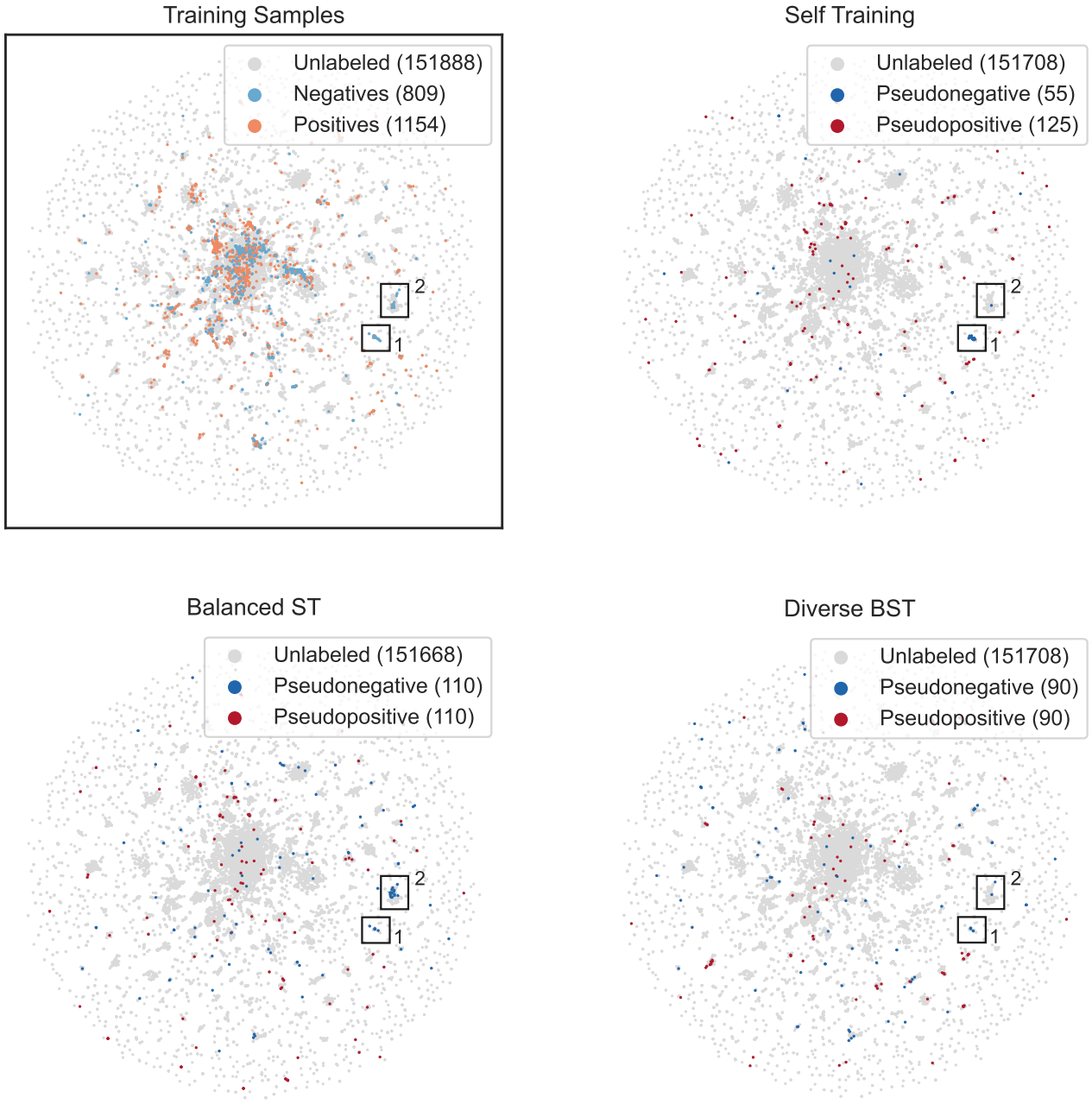

Fig. 4. UMAP projections of the BRCA dataset. In the top left, the training samples are highlighted. The top right plot shows the pseudosamples selected during the self-training approach. The bottom left contains pseudosamples selected during the balanced self-training approach. The bottom right contains pseudosamples selected during the diversity approach. The numbers behind the labels represent the numbers of samples of each class in the figure. The boxes with number one highlight a cluster dominated by gene pairs containing the gene CDH1. The boxes with number two highlight gene pairs dominated by the genes COL4A1, COL4A2, COL4A4 and COL4A6

method compared to the ST and BST methods. This increase was likely caused due to the self-training method selecting pseudosamples close to the biased samples in the training set, while we can select more diverse samples in the embedding space with the diversity approach.

Additionally, the distances of selected pseudosamples in BST were higher than in ST, with a more drastic increase in the distances between the non-SL samples. In the ST method, there were only 55 negative and 125 positive selected pseudosamples, while in the BST method, there were 110 negative and 110 positive selected pseudosamples. This difference means that while selecting pseudosamples, BST had to choose more negatives even if the confidence of those negatives might have been slightly lower than the confidence of the positives selected in ST because BST has to select as many positives as negatives. So when adding these confident negatives, even if they were not the most confident samples overall, increased the distance between the pseudosamples.

To further examine the differences between the three semi-supervised methods, we examined the distribution of

selected pseudosamples in a UMAP projection of the original feature space onto two dimensions. For this, we performed a UMAP projection of the labeled and unlabeled samples, and highlighted the selected pseudosamples according to their assigned pseudolabel (Fig. 4). The highlighted gene pairs in the top left represent the original labeled samples.

The first note that can be made is that there is no clear separation between the positives and negatives in the training samples. This scattered distribution shows the complexity of the problem and that there is no obvious linear decision boundary. Another note is that many clusters in the UMAP projection contain no or few training samples. This lack of representation further illustrates the bias problem in synthetic lethality, as these clusters are not represented in the train set and, therefore, can not be considered while training.

During self-training, the distribution of pseudolabels changed to only 55 negatives added compared to 125 positives. Of these 55 negatives, 29 were in a cluster dominated by the gene CDH1 (Fig. 4 highlighted in box 1). This cluster contained 507 labeled and unlabeled samples, of which 501 samples contained the gene CDH1. The fact that the method focused so heavily on one cluster during self-training demonstrates the main drawback of using this approach to mitigate the effect of selection bias. When we examined this cluster in the BST and DBST approaches, we saw that they added 4 and 6 pseudonegatives from this cluster, respectively. Both of these methods included more pseudonegatives and yet had fewer pseudosamples in this cluster, demonstrating that BST and DBST were better at including less similar samples and did not strengthening the bias by adding samples from the biased part of the distribution.

We further examined the UMAP and noticed the cluster highlighted by the second box (Fig. 4). This cluster contained 1059 gene pairs, all of which containedone of the following genes: COL4A1, COL4A2, COL4A4 or COL4A6. Using ST and DBST, only one pseudosample was selected from this cluster, while BST selected 18 pseudosamples from this cluster. This illustrates the same drawback as ST in the CDH1 cluster in that BST selects pseudosamples from high-confidence regions. By balancing the number of negatives and positives that can be selected, we limit the number of negatives that can be selected and force the model to focus on high-confidence positives in this case.

How does metric learning perform when train and test set have different biases

Double holdout

The first experiment was the double holdout experiment. As explained in the methods, for this experiment there was no overlap in the genes between the test and train sets to ensure that both sets were forming a different distribution. The downside of splitting the data in such a strict condition is that the train set became relatively small. The smallest train set was CESC, with only around 90 samples.

The AUPRC across all pseudolabeling methods was lower than in the random split experiment (Fig. 5). This outcome was expected, as the double holdout experiment is more restrictive in size and distribution than the randomized experiment. Four cancer types (BRCA, OV, SKCM, LUAD) did not improve by adding pseudolabels. For one cancer type, CESC, we did see an improvement in AUPRC when comparing the supervised approach (median AUPRC 0.560) with the BST and DBST methods (median AUPRC BST 0.584, DBST 0.596), where

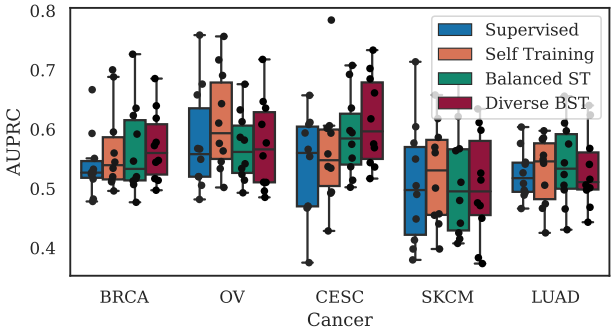

Fig. 5. Performance of the double holdout experiment across ten iterations. The experiments include five different cancer types (BRCA, OV, CESC, SKCM, LUAD) and are highlighted according to the four different methods (supervised, ST, BST and DBST). The dots represent the AUPRC of an individual run.

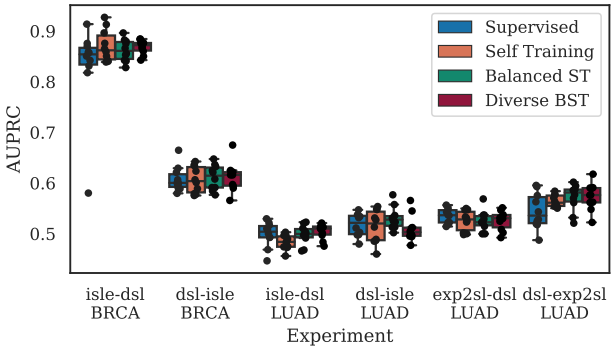

Fig. 6. Performance of the multiple studies experiment across ten iterations. The experiments include six combinations of different studies and cancer types. The first study is the training set, and the second study is used as the test set. Each is highlighted according to the four methods (Supervised, ST, BST, DBST). The dots represent the AUPRC of an individual run.

the DBST was the only experiment that showed a significant improvement (p-values BST 0.105, DBST 0.010). However, it is necessary to note that the performance of the pseudolabeling methods again did not decrease the performance compared to the supervised approach.

The low performance for the double holdout experiment has two potential causes. The first potential cause is that the double holdout experiment is too restrictive. The train and test sets not only follow different biases, but they also have little in common. This makes it incredibly difficult for the model to find an improvement. The second potential cause is that the train set is very small. The number of samples available decreased drastically by adding the double holdout constraint. A combination of the two may have had the effect that the performance of the baseline supervised method already dropped significantly (median AUPRC Randomized split BRCA 0.853, OV 0.701, CESC 0.587, SKCM 0.747, LUAD 0.837; Double holdout BRCA 0.527, OV 0.558, CESC 0.560, SKCM 0.497, LUAD 0.517), making it difficult for semi-supervised methods to improve since these methods use the initial model to find new samples to add.

Multiple SL label sources

The second experiment to test our proposed model's ability to generalize the actual distribution was to train on data

with SL labels available from one study and test with data containing SL labels from another study (Fig. 6). This was done for combinations of three different studies where each study functioned as either train set or test set, resulting in six experiments. In future reference, the first study represents the train set, the second study represents the test set.

We saw similar results to the double holdout experiments. Five of the six experiments across SL label sources showed a low AUPRC across all pseudolabeling methods (median AUPRC dsl-isle BRCA ST 0.604, BST 0.614, DBST 0.616; isle-dsl LUAD ST 0.484, BST 0.499, DBST 0.510; dsl-isle LUAD ST 0.520, BST 0.528, DBST 0.502; exp2sl-dsl LUAD ST 0.528, BST 0.522, DBST 0.530; dsl-exp2sl LUAD ST 0.561, BST 0.581, DBST 0.576). This low AUPRC was also likely caused by the low starting performance of the supervised approach (median AUPRC dsl-isle BRCA 0.600, isle-dsl LUAD 0.504, dsl-isle LUAD 0.521, exp2sl-dsl LUAD 0.536, dsl-exp2sl LUAD 0.536). However, we saw a high AUPRC for the model trained on samples with SL labels from the isle BRCA dataset and tested on the dSL BRCA dataset. This specific experiment showed the same behaviour as the random split experiments on BRCA and LUAD, where the supervised performance was already between 0.8 and 0.9 (median AUPRC 0.854), and the semi-supervised approaches did not seem to improve the AUPRC (median AUPRC ST 0.862, BST 0.861, DBST 0.872)(p-values ST 0.105, BST 0.375, DBST 0.105).

In the experiment where the model was trained on the LUAD dSL dataset and evaluated on the LUAD Exp2SL dataset, the AUPRC of the three semi-supervised methods (median AUPRC ST 0.561, BST 0.581, DBST 0.576) showed an improvement over the supervised method (median AUPRC 0.536). The performance increase of the models was significant for this experiment (p-values ST 0.049, BST 0.006, DBST 0.014). This was the only experiment that significantly improved when trained on one dataset and tested on another.

The other four experiments did not show the improvement we expected to see and might be caused due to a limited training size, also in this experiment. Another possibility is that our model does not work well to generalize the actual distribution and that it does not work well to mitigate the effect of selection bias. However, this is still not a conclusion that can be made with certainty.

Conclusion

In this paper, we proposed a model that combines both semi-supervised learning and metric learning to address sample selection bias in synthetic lethality prediction. We proposed a novel method to add diverse samples during self-training to mitigate the bias present in SL labels. This is achieved using metric learning to find a class-contrastive representation of the feature space, based on which distances between samples can be calculated to inform the selection of diverse or dissimilar samples.

In an experiment where we randomly split the data into train and test sets, semi-supervised learning with metric learning improved the performance over supervised metric learning in specific cases. We were able to improve the performance on ovarian (OV) and cervical (CESC) cancers. On other cancer types, the performance of the semi-supervised models was similar to the supervised performance. Due to the test and train set following the same bias, this experiment did not answer the question of mitigating sample selection bias directly. However, it did show that adding pseudosamples could help in some cases and that performance did not drop while attempting to generalize over the true distribution.

We further showed that our DBST method did select more diverse samples compared to ST and BST. The distances between selected pseudosamples were higher in DBST compared to ST and BST. DBST was also less likely to select similar pseudosamples in the original feature space. ST was more likely to reinforce the bias by selecting samples from high-confidence regions that were more similar in the original feature space. This shows the danger of using self-training directly without making proper modifications for diversity for biased problems, where the model is likely to reinforce this bias rather than mitigate it.

In the cases where the train and test set followed a different bias, we saw some improvements of our proposed models, but the experiments might have been too restrictive to conclusively determine the performance of our proposed models with regard to mitigating selection bias. By restricting the train and test set to have no overlap in the genes present in the gene pairs, we severely limited the number of training samples, which resulted in a low starting or baseline performance. This also made it difficult for the subsequent pseudolabeling iterations to find informative samples.

For future work, DBST does need more testing concerning the selection bias problem. We need experiments where we can simultaneously induce a different bias in the test set while keeping the size of the train set as large as possible. This will be a more fair comparison as now current experiments induced a strong bias as well as limited the train set size, which could both have a significant impact on the results on their own. Comparing the performance between our proposed models to the performance of other selection bias mitigation methods would also be interesting since we only examined the capabilities of different semi-supervised versions of metric learning. Also, comparing our method to other SL prediction models such as EXP2SL (17) could be interesting to measure the impact of the bias mitigation more accurately.

The model could be further improved by using another form of hyperparameter optimization. In the current model, grid search was used for simplicity, but another more advanced tuning, such as gradient-based optimization or evolutionary algorithms, could be used. This would also allow for tuning across more hyperparameters. It could also be interesting to use other machine learning algorithms to learn a classifier, but keep metric learning as a method to add diversity in the selection of pseudosamples. This would allow other classifiers to add diversity to their selected pseudosamples since most methods do not create a latent space like metric learning. A limitation of the current method is the way pseudosamples are selected. This is based on a random coordinate in the possible embedding space. However, it is not guaranteed that the samples, including the unlabeled samples, span the entire space. So a more direct method where only the possible samples are considered, and not the space where no samples are projected to, could further enhance this model.

To conclude, we introduced a new method to mitigate sample selection bias. We did not conclusively prove its potential. However, we hope this method gives new insights into reducing the effects of sample selection bias.

References

1. B. Zhang, C. Tang, Y. Yao, X. Chen, C. Zhou, Z. Wei, F. Xing, L. Chen, X. Cai, Z. Zhang, S. Sun, and Q. Liu, "The tumor therapy landscape of synthetic lethality," *Nature Communications*, vol. 12, Feb. 2021.
2. N. J. O'Neil, M. L. Bailey, and P. Hieter, "Synthetic lethality and cancer," *Nature Reviews Genetics*, vol. 18, pp. 613–623, June 2017.
3. H. E. Bryant, N. Schultz, H. D. Thomas, K. M. Parker, D. Flower, E. Lopez, S. Kyle, M. Meuth, N. J. Curtin, and T. Helleday, "Specific killing of BRCA2-deficient tumours with inhibitors of poly(ADP-ribose) polymerase," *Nature*, vol. 434, pp. 913–917, Apr. 2005.
4. H. Farmer, N. McCabe, C. J. Lord, A. N. J. Tutt, D. A. Johnson, T. B. Richardson, M. Santarosa, K. J. Dillon, I. Hickson, C. Knights, N. M. B. Martin, S. P. Jackson, G. C. M. Smith, and A. Ashworth, "Targeting the DNA repair defect in BRCA mutant cells as a therapeutic strategy," *Nature*, vol. 434, pp. 917–921, Apr. 2005.
5. P. C. Fong, D. S. Boss, T. A. Yap, A. Tutt, P. Wu, M. Mergui-Roelvink, P. Mortimer, H. Swaisland, A. Lau, M. J. O'Connor, A. Ashworth, J. Carmichael, S. B. Kaye, J. H. Schellens, and J. S. de Bono, "Inhibition of poly(ADP-ribose) polymerase in tumors fromiBRCA/imutation carriers," *New England Journal of Medicine*, vol. 361, pp. 123–134, July 2009.
6. C. J. Lord and A. Ashworth, "PARP inhibitors: Synthetic lethality in the clinic," *Science*, vol. 355, pp. 1152–1158, Mar. 2017.
7. S. Li, W. Topatana, S. Juengpanich, J. Cao, J. Hu, B. Zhang, D. Ma, X. Cai, and M. Chen, "Development of synthetic lethality in cancer: molecular and cellular classification," *Signal Transduction and Targeted Therapy*, vol. 5, Oct. 2020.
8. S. L. Salzberg, "Open questions: How many genes do we have?," *BMC Biology*, vol. 16, Aug. 2018.
9. J. Wang, Q. Zhang, J. Han, Y. Zhao, C. Zhao, B. Yan, C. Dai, L. Wu, Y. Wen, Y. Zhang, D. Leng, Z. Wang, X. Yang, S. He, and X. Bo, "Computational methods, databases and tools for synthetic lethality prediction," *Briefings in Bioinformatics*, vol. 23, Mar. 2022.
10. L. Jerby-Arnon, N. Pfetzer, Y. Y. Waldman, L. McGarry, D. James, E. Shanks, B. Seashore-Ludlow, A. Weinstock, T. Geiger, P. A. Clemons, E. Gottlieb, and E. Ruppin, "Predicting cancer-specific vulnerability via data-driven detection of synthetic lethality," *Cell*, vol. 158, pp. 1199–1209, Aug. 2014.
11. C. Seale, Y. Tepeli, and J. P. Gonçalves, "Overcoming selection bias in synthetic lethality prediction," *Bioinformatics*, vol. 38, pp. 4360–4368, July 2022.
12. Y. Tepeli, C. Seale, and J. Gonçalves, "ELISL: Early-late integrated synthetic lethality prediction in cancer," Sept. 2022.
13. N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," 2019.
14. S. R. Paladugu, S. Zhao, A. Ray, and A. Raval, "Mining protein networks for synthetic genetic interactions," *BMC Bioinformatics*, vol. 9, Oct. 2008.
15. S. Das, X. Deng, K. Camphausen, and U. Shankavaram, "DiscoverSL: an r package for multi-omic data driven prediction of synthetic lethality in cancers," *Bioinformatics*, vol. 35, pp. 701–702, July 2018.
16. J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, pp. 373–440, Nov. 2019.
17. F. Wan, S. Li, T. Tian, Y. Lei, D. Zhao, and J. Zeng, "EXP2sl: A machine learning framework for cell-line-specific synthetic lethality prediction," *Frontiers in Pharmacology*, vol. 11, Feb. 2020.
18. D.-H. Lee, "Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks," *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013.
19. A. Radhakrishnan, J. Davis, Z. Rabin, B. Lewis, M. Scherreik, and R. Ilin, "Enhancing self-training methods," 2023.
20. E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning," 2019.
21. B. Chen, J. Jiang, X. Wang, P. Wan, J. Wang, and M. Long, "Debiased self-training for semi-supervised learning," 2022.
22. D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, "Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring," 2019.
23. K. Dost, K. Taskova, P. Riddle, and J. Wicker, "Your best guess when you know nothing: Identification and mitigation of selection bias," in *2020 IEEE International Conference on Data Mining (ICDM)*, IEEE, Nov. 2020.
24. P. K. Rhee, E. Erdenee, S. D. Kyun, M. U. Ahmed, and S. Jin, "Active and semi-supervised learning for object detection with imperfect data," *Cognitive Systems Research*, vol. 45, pp. 109–123, Oct. 2017.
25. Kaya and Bilge, "Deep metric learning: A survey," *Symmetry*, vol. 11, p. 1066, Aug. 2019.
26. S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE.
27. X. Lu, W. Megchelenbrink, R. A. Notebaart, and M. A. Huynen, "Predicting human genetic interactions from cancer genome evolution," *PLOS ONE*, vol. 10, p. e0125795, May 2015.
28. J. S. Lee, A. Das, L. Jerby-Arnon, R. Arafeh, N. Auslander, M. Davidson, L. McGarry, D. James, A. Amzallag, S. G. Park, K. Cheng, W. Robinson, D. Atias, C. Stossel, E. Buzhor, G. Stein, J. J. Waterfall, P. S. Meltzer, T. Golan, S. Hannenhalli, E. Gottlieb, C. H. Benes, Y. Samuels, E. Shanks, and E. Ruppin, "Harnessing synthetic lethality to predict the response to cancer treatment," *Nature Communications*, vol. 9, June 2018.
29. M. Heinzinger, A. Elnaggar, Y. Wang, C. Dallago, D. Nechaev, F. Matthes, and B. Rost, "Modeling aspects of the language of life through transfer-learning protein sequences," *BMC Bioinformatics*, vol. 20, Dec. 2019.
30. J. Campbell, C. J. Ryan, R. Brough, I. Bajrami, H. N. Pemberton, I. Y. Chong, S. Costa-Cabral, J. Frankum, A. Gulati, H. Holme, R. Miller, S. Postel-Vinay, R. Rafiq, W. Wei, C. T. Williamson, D. A. Quigley, J. Tym, B. Al-Lazikani, T. Fenton, R. Natrajan, S. J. Strauss, A. Ashworth, and C. J. Lord, "Large-scale profiling of kinase dependencies in cancer cell lines," *Cell Reports*, vol. 14, pp. 2490–2501, Mar. 2016.
31. Y. Marrakchi, O. Makansi, and T. Brox, "Fighting class imbalance with contrastive learning," in *Medical*

Image Computing and Computer Assisted Intervention – MICCAI 2021, pp. 466–476, Springer International Publishing, 2021.

32. X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, "Range loss for deep face recognition with long-tailed training data," in *2017 IEEE International Conference on Computer Vision (ICCV)*, IEEE, Oct. 2017.
33. C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, June 2016.
34. L. Gautheron, A. Habrard, E. Morvant, and M. Sebban, "Metric learning from imbalanced data with generalization guarantees," *Pattern Recognition Letters*, vol. 133, pp. 298–304, May 2020.

Table 2. Selected parameters for the randomized split experiments. For μ, the values 0.80, 0.85, 0.90 and 0.95 were tested. For p, the values 10, 20 and 50 were tested.

Cancer	μ (confidence threshold)	p (number of pseudosamples each iteration)
BRCA	0.90	20
OV	0.85	20
CESC	0.90	10
SKCM	0.90	10
LUAD	0.90	10

Table 3. Selected parameters for the double holdout experiments. For μ, the values 0.70, 0.75, 0.80, 0.85, 0.90 and 0.95 were tested. For p, the values 6, 10 and 20 were tested.

Cancer	μ (confidence threshold)	p (number of pseudosamples each iteration)
BRCA	0.85	6
OV	0.80	6
CESC	0.75	6
SKCM	0.75	20
LUAD	0.90	10

Table 4. Selected parameters for the Mulitple SL label sources experiments. For μ, the values 0.75, 0.80, 0.85, 0.90 and 0.95 were tested. For p, the values 4, 6, 10 and 20 were tested.

Training study	Test study	Cancer	p (number of pseudosamples each iteration)	μ (confidence threshold)
ISLE	dSL	BRCA	6	0.85
dSL	ISLE	BRCA	4	0.95
ISLE	dSL	LUAD	10	0.80
dSL	ISLE	LUAD	10	0.85
EXP2SL	dSL	LUAD	6	0.85
dSL	EXP2SL	LUAD	10	0.80

Table 5. Final distribution of classes in ST in the Randomized split experiments. The percentage of final train set that are reported are averaged over 10 runs.

Cancer	Share majority class (%)	Share minority class (%)
BRCA	55 ± 2	45 ± 2
OV	69 ± 3	31 ± 3
SKCM	70 ± 5	30 ± 5
CESC	68 ± 6	32 ± 6
LUAD	56 ± 2	44 ± 2

Table 6. Sizes of datasets for the Multiple SL label sources experiments

Training study	Test study	Cancer	training	pos	neg	test	pos	neg	unlabeled
ISLE	dSL	BRCA	1509	573	935	960	885	75	151882
dSL	ISLE	BRCA	893	854	39	1575	590	985	151882
ISLE	dSL	LUAD	4897	168	4729	711	372	339	150944
dSL	ISLE	LUAD	711	372	339	4897	168	4729	150944
EXP2SL	dSL	LUAD	2676	307	2369	711	372	339	150944
dSL	EXP2SL	LUAD	711	372	339	2676	307	2369	150944

Table 7. Sizes of individual runs for the randomized split experiment after splitting the data and balancing for each set.

Iteration	Cancer	training	pos	neg	validation	pos	neg	test	pos	neg
1	BRCA	1294	647	647	324	162	162	402	201	201
2	BRCA	1294	647	647	324	162	162	402	201	201
3	BRCA	1294	647	647	324	162	162	402	201	201
4	BRCA	1294	647	647	324	162	162	402	201	201
5	BRCA	1294	647	647	324	162	162	402	201	201
6	BRCA	1294	647	647	324	162	162	402	201	201
7	BRCA	1294	647	647	324	162	162	402	201	201
8	BRCA	1294	647	647	324	162	162	402	201	201
9	BRCA	1294	647	647	324	162	162	402	201	201
10	BRCA	1294	647	647	324	162	162	402	201	201
1	OV	324	162	162	80	40	40	102	51	51
2	OV	324	162	162	80	40	40	102	51	51
3	OV	324	162	162	80	40	40	102	51	51
4	OV	324	162	162	80	40	40	102	51	51
5	OV	324	162	162	80	40	40	102	51	51
6	OV	324	162	162	80	40	40	102	51	51
7	OV	324	162	162	80	40	40	102	51	51
8	OV	324	162	162	80	40	40	102	51	51
9	OV	324	162	162	80	40	40	102	51	51
10	OV	324	162	162	80	40	40	102	51	51
1	CESC	184	92	92	46	23	23	58	29	29
2	CESC	184	92	92	46	23	23	58	29	29
3	CESC	184	92	92	46	23	23	58	29	29
4	CESC	184	92	92	46	23	23	58	29	29
5	CESC	184	92	92	46	23	23	58	29	29
6	CESC	184	92	92	46	23	23	58	29	29
7	CESC	184	92	92	46	23	23	58	29	29
8	CESC	184	92	92	46	23	23	58	29	29
9	CESC	184	92	92	46	23	23	58	29	29
10	CESC	184	92	92	46	23	23	58	29	29
1	SKCM	138	69	69	34	17	17	42	21	21
2	SKCM	138	69	69	34	17	17	42	21	21
3	SKCM	138	69	69	34	17	17	42	21	21
4	SKCM	138	69	69	34	17	17	42	21	21
5	SKCM	138	69	69	34	17	17	42	21	21
6	SKCM	138	69	69	34	17	17	42	21	21
7	SKCM	138	69	69	34	17	17	42	21	21
8	SKCM	138	69	69	34	17	17	42	21	21
9	SKCM	138	69	69	34	17	17	42	21	21
10	SKCM	138	69	69	34	17	17	42	21	21
1	LUAD	760	380	380	190	95	95	238	119	119
2	LUAD	760	380	380	190	95	95	238	119	119
3	LUAD	760	380	380	190	95	95	238	119	119
4	LUAD	760	380	380	190	95	95	238	119	119
5	LUAD	760	380	380	190	95	95	238	119	119
6	LUAD	760	380	380	190	95	95	238	119	119
7	LUAD	760	380	380	190	95	95	238	119	119
8	LUAD	760	380	380	190	95	95	238	119	119
9	LUAD	760	380	380	190	95	95	238	119	119
10	LUAD	760	380	380	190	95	95	238	119	119

Table 8. Sizes of individual runs for the double holdout experiments

Iteration	Cancer	training	pos	neg	validation	pos	neg	test	pos	neg
1	BRCA	520	260	260	130	65	65	214	107	107
2	BRCA	532	266	266	132	66	66	210	105	105
3	BRCA	460	230	230	114	57	57	216	108	108
4	BRCA	524	262	262	130	65	65	208	104	104
5	BRCA	558	279	279	140	70	70	202	101	101
6	BRCA	494	247	247	124	62	62	202	101	101
7	BRCA	460	230	230	116	58	58	256	128	128
8	BRCA	520	260	260	130	65	65	224	112	112
9	BRCA	534	267	267	134	67	67	202	101	101
10	BRCA	538	269	269	134	67	67	206	103	103
1	OV	142	71	71	36	18	18	42	21	21
2	OV	142	71	71	36	18	18	50	25	25
3	OV	136	68	68	34	17	17	40	20	20
4	OV	144	72	72	36	18	18	44	22	22
5	OV	148	74	74	36	18	18	44	22	22
6	OV	144	72	72	36	18	18	50	25	25
7	OV	136	68	68	34	17	17	46	23	23
8	OV	152	76	76	38	19	19	44	22	22
9	OV	140	70	70	36	18	18	46	23	23
10	OV	140	70	70	34	17	17	44	22	22
1	CESC	90	45	45	22	11	11	28	14	14
2	CESC	90	45	45	22	11	11	30	15	15
3	CESC	90	45	45	22	11	11	28	14	14
4	CESC	92	46	46	22	11	11	30	15	15
5	CESC	90	45	45	22	11	11	28	14	14
6	CESC	88	44	44	22	11	11	30	15	15
7	CESC	92	46	46	22	11	11	30	15	15
8	CESC	88	44	44	22	11	11	28	14	14
9	CESC	92	46	46	22	11	11	28	14	14
10	CESC	90	45	45	22	11	11	24	12	12
1	SKCM	120	60	60	30	15	15	22	11	11
2	SKCM	120	60	60	30	15	15	22	11	11
3	SKCM	120	60	60	30	15	15	22	11	11
4	SKCM	120	60	60	30	15	15	22	11	11
5	SKCM	120	60	60	30	15	15	22	11	11
6	SKCM	120	60	60	30	15	15	22	11	11
7	SKCM	120	60	60	30	15	15	22	11	11
8	SKCM	120	60	60	30	15	15	22	11	11
9	SKCM	120	60	60	30	15	15	22	11	11
10	SKCM	120	60	60	30	15	15	22	11	11
1	LUAD	322	161	161	80	40	40	106	53	53
2	LUAD	334	167	167	84	42	42	104	52	52
3	LUAD	322	161	161	80	40	40	100	50	50
4	LUAD	334	167	167	84	42	42	106	53	53
5	LUAD	334	167	167	84	42	42	104	52	52
6	LUAD	380	190	190	94	47	47	120	60	60
7	LUAD	340	170	170	84	42	42	106	53	53
8	LUAD	348	174	174	86	43	43	108	54	54
9	LUAD	322	161	161	80	40	40	100	50	50
10	LUAD	330	165	165	82	41	41	106	53	53