Computational and Structural Biotechnology Journal 19 (2021) 3255–3268

COMPUTATIONAL AND STRUCTURAL BIOTECHNOLOGY JOURNAL

journal homepage: www.elsevier.com/locate/csbj

Review

A comparative study of multiple instance learning methods for cancer detection using T-cell receptor sequences

Danyi Xiong [a,b], Ze Zhang [b], Tao Wang [b,*], Xinlei Wang [a,*]

[a] Department of Statistical Science, Southern Methodist University, 3225 Daniel Avenue, Dallas 75275, TX, USA
[b] Department of Population and Data Sciences, University of Texas Southwestern Medical Center, 5323 Harry Hines Boulevard, Dallas 75390, TX, USA

ARTICLE INFO

Article history:
Received 28 February 2021
Received in revised form 12 May 2021
Accepted 20 May 2021
Available online 24 May 2021

Keywords:
Binary classification
Primary instance
T-cell receptor
Witness rate
Weakly supervised learning

ABSTRACT

As a branch of machine learning, multiple instance learning (MIL) learns from a collection of labeled bags, each containing a set of instances. The learning process is weakly supervised due to ambiguous instance labels. Since its emergence, MIL has been applied to solve various problems including content-based image retrieval, object tracking/detection, and computer-aided diagnosis. In biomedical research, the use of MIL has been focused on medical image analysis and molecule activity prediction. We review and apply 16 methods to investigate the applicability of MIL to a novel biomedical application, cancer detection using T-cell receptor (TCR) sequences. This important application can be a viable approach for large-scale cancer screening, as TCRs can be easily profiled from a subject's peripheral blood. We consider two feasible data-generating mechanisms, and for the purpose of performance evaluation, we simulate data under each mechanism, where we vary potentially important factors to mimic realistic situations. We also apply the methods to sequencing data of ten cancer types from The Cancer Genome Atlas, as an early proof of concept for distinguishing tumor patients from healthy individuals via TCR sequencing of peripheral blood. We find that given an appropriate MIL method is used, satisfactory performance with Area Under the Receiver Operating Characteristic Curve above 80% can be achieved for five in the ten cancers. Based on our numerical results, we make suggestions about selection of a proper method and avoidance of any method with poor performance. We further point out directions of future research as well as identify a pressing need of new MIL methodologies for improved performance (for some cancer types) and more explainable outcomes.

Contents

* Corresponding author.
E-mail addresses: Tao.Wang@UTSouthwestern.edu (T. Wang), swang@smu.edu (X. Wang).

https://doi.org/10.1016/j.csbj.2021.05.038

1. Introduction

First introduced in [1], multiple instance learning (MIL) has been used to tackle a wide range of problems, in which the learning task is performed on a set of labeled "bags", each being a collection of "instances". Each individual instance is described by a set of covariates (or features). Instances in one bag contribute to the observed bag-level response (or label). Often, the instance label cannot be observed directly, and sometimes is even not defined clearly. The main objective of MIL is to predict bag labels based on the instance-level covariates by learning the relationship among bags and instances. In applications such as object detection, instance labels are also of interest.

The binary classification problem is most frequently encountered in MIL. For example, in drug activity prediction, Dietterich et al. [1] developed an MIL algorithm to classify whether a molecule (bag) of different conformations (instances) is biologically active; in content-based image retrieval [2], the authors employed MIL to determine whether a given image (bag) contains a particular object in at least one of non-overlapping regions of the image (instances); in document classification [3], an article (bag) is categorized based on passages contained (instances). MIL methods have also been developed for multi-label classification, with major applications in scene/image categorization [4–8]. In addition to classification, MIL is applicable for real-valued responses as well [9–11]. A recent application of multiple instance regression studied the relationship between tumor immune response and immunogenic neoantigens using a Bayesian hierarchical model [12]. Less common than classification and regression, unsupervised learning tasks, such as MI ranking and clustering, where no response is attached to any bag, have also been investigated by researchers [13–16].

Over the past two decades, numerous MIL methods have been developed by researchers to adapt to the diverse characteristics of multiple instance (MI) problems. Several papers have compared or categorized existing MIL methods and applications. Foulds and Frank [18] gave a detailed review of the standard MI assumption and alternative assumptions made on the data generation process with respect to the relationship between bags and instances; their work focused on clarification of relevant concepts involved in MIL rather than performance evaluation of different methods. Amores [19] provided a concise categorization of MIL methods for binary classification, depending on the means that a method takes to learn bag labels from instances in the bags. However, this work only considered two MI problem characteristics (i.e., witness rate and number of components in the distribution for positive instances) in the simulation design. It also excluded more recent MIL methods [20,21]. A more recent study conducted by Carbonneau et al. [22] formally identified four MI problem characteristics. Nevertheless, it lacked a clear distinction between bag composition and label ambiguity, hence may hinder one's understanding of how instances contribute to bag labels in a specific MI application. Finally, previous research has demonstrated the suitability of MIL methods in many applications from various fields such as biology and chemistry, computer vision, document classification, web mining, and activity recognition [17,21–25]. In this paper, we focus on a novel biomedical application, cancer detection using T-cell receptor (TCR) sequences, where the applicability of MIL methods is yet to be examined.

Accurate and timely cancer detection, especially for aggressive cancer types, is extremely important for patients to receive appropriate treatments for best possible prognosis [26]. Various experimental methods exist, which, however, are less ideal for detecting certain types of cancer [27–29]. Based on previous findings that the host immune responses to tumor cells are already activated during tumorigenesis process [30–32], one possible and more universal approach to discern tumors from normal tissue samples is to examine the TCR sequences, which are capable of reflecting the state of the host T cell immunity system, and may contain critical information regarding whether tumors have been progressing in the human body. This problem fits naturally into the MIL framework as there are a large number of T cells with different TCRs (instances) in each patient (bag). TCRs are proteins expressed on the surface of the T cells and used by the latter to target and initiate the destruction of the tumor cells. Structural characteristics of the TCRs, which can be obtained by well established sequencing techniques from a patient's blood, could be used to predict whether the patient has tumor(s) or not.

Apart from biomedical studies that only descriptively characterized TCRs in tumor and normal tissues, such as Jin et al. [33], few researchers have sought to predict tumor or normal status based on TCRs of T cells. Beshnova et al. [34] developed a deep learning-based method for predicting tumor associated TCRs. While showing some promise, this method is not an MIL approach, hence it ignores the bag-instance relationship (i.e. patient-TCR relationship in their work) that naturally arises in the context of this application, rendering model interpretation difficult. Ostmeyer et al. [35] developed an MIL model for distinguishing tumor infiltrating T cells from T cells of adjacent normal tissues. However, this work suffers from small sample sizes (only 28 and 32 patients for breast cancer and colorectal cancer, respectively). The employed MIL model has a simple design based on the standard MIL assumption and does not utilize global bag-level information. Furthermore, there was no comparison with the state-of-the-art MIL methods. Thus, whether their conclusion is generalizable remains open.

This study provides an up-to-date review of MIL methods that are applicable to our application. We examine the performance of the methods in cancer detection via comprehensive simulation and real data examples. The remainder of this paper is organized as follows. In Section 2, we describe data and problem characteristics that are relevant to our MIL application, and identify key

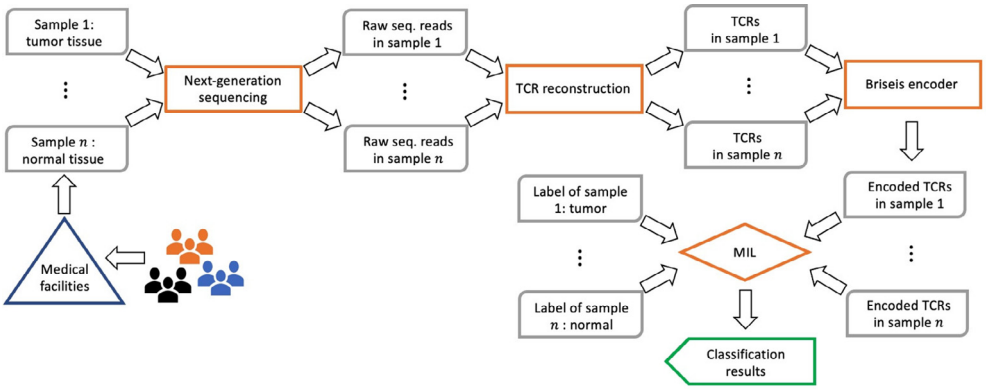


Fig. 1. The pipeline of data processing in our MIL application of cancer detection using TCR sequences. Each encoded TCR sequence is represented by a d-dimensional numeric feature vector learned by the auto-encoder.

concepts. In Section 3, we describe a list of MIL methods for our application and comment briefly on their implementation. In Section 4, we carry out a simulation study comparing the performance of the selected MIL methods on synthetic datasets, which simulate various scenarios that may occur in real data. In Section 5, we conduct analysis on real datasets obtained from The Cancer Genome Atlas (TCGA). We discuss our major findings and future work in Section 6.

2. Cancer detection using TCR sequences

2.1. Data generation

Fig. 1 depicts how human biospecimens are processed to generate the input data for MIL algorithms. Tissue samples collected from patients by medical facilities are analyzed using next-generation sequencing techniques and genomic data for each sample are obtained. TCR sequences in each sample are then detected from its raw sequencing reads via TCR reconstruction software such as TRUST [36] and MiTCR [37]. Under the MIL framework, each sample is considered as a bag consisting of TCR sequences (instances), which are essentially text strings. We embed each TCR sequence into a numeric vector using our previously published Tessa model [38], which is equipped with a deep learning auto-encoder that converts complex information (strings of amino acids in this case) to numeric values. In short, each amino acid of a TCR sequences is encoded by the five Atchley factors [39] that can fully capture their physicochemical properties. A stacked auto-encoder is then applied to the "Atchely matrices" of TCRs to represent the Atchley-factor-encoded TCR sequences by d-dimensional numeric vectors through a decomposition-reconstruction process. Our previous work has systematically established the validity of this approach [38]. By representing each TCR sequence using a numeric vector, we make it convenient for MIL methods to utilize these features, for instance, to calculate distances among instances and/or bags.

2.2. Problem characteristics and related concepts

MIL differs from standard supervised learning in that a single class label is assigned to a bag of instances rather than every individual instance. MIL is weakly supervised as instance-level labels may be vaguely defined and not observed, and the relationship between a bag and its instances is unclear. Consider a bag with m instances, collectively described by a feature matrix $X = \{x_1, \ldots, x_m\}$. Each instance j is represented by a feature vector x_j in a d-dimensional feature space (i.e., $x_j \in \mathbb{R}^d$). For binary classification, as in our cancer detection application, we may build a function $F(X) \in [0, 1]$ and determine the bag to be positive (or negative) if $F(X)$ is above (or below) some cutoff value. The classification function $F(X)$ is learned from a training set with the sample size n (i.e., n bags), denoted by $\mathrm{T} = \{(X_i, y_i)_{i=1}^n\}$, where $y_i \in \{0, 1\}$ is the label of bag i ($y_i = 1$ if the bag is positive and $y_i = 0$ otherwise).

Although instance labels may not be directly observable, instance classification can be of interest in MI applications as well. For example, in image-based object detection, one or more segments of an image correspond to a dog if a dog is in the image; in this sense, an instance (segment) is positive if it contributes to a positive bag (i.e., an image containing a dog). In drug activity prediction, a molecule (bag) has to have the "right" conformation (instance) to possess the binding potency. In our application, the primary objective is to classify bags. That is, we aim to identify whether a screening subject has tumor or not. However, instance classification can be useful if tumor-specific TCRs can be identified and leveraged for engineering TCR-T therapies [40]. Analogous to the bag-level classifier $F(X)$, we use the generic notation $f(x)$ to denote the instance-level classifier, which assigns a label to the instance with the feature vector x.

For our application and potentially other applications of MIL, we consider two possible formulations (or data generation mechanisms) with respect to how the label of a bag is determined by its instances. The first relies on instance classification and the concept of witness rate (WR), defined as the proportion of positive instances in positive bags [22]. It is assumed that only the number or proportion of the positive instances is responsible for labeling a bag as positive (e.g., a positive bag has at least t positive instances, or a positive bag has at least t% of instances being positive), with more positive instances typically indicating a higher confidence of a positive bag. The standard assumption, which is the most commonly used in the MI literature, states that a positive bag has at least one positive instance and a negative bag only has negative instances [1]. This classical assumption fits in the WR framework, in which WR equals 0 for a negative bag and ranges from 0 (exclusively) to 1 for a positive bag. As pointed out by Carbonneau et al. [22], positive bags with low WRs may result in poor performance of many MIL methods as positive and negative bags become similar. In general, the concept of WR applies to scenarios where instance classification is meaningful. In our application of cancer detection, WR may naturally correspond to the proportion of tumor-specific TCR clones out of all TCR clones in a tissue sample. Tumor-specific TCRs, once present in a bag (sample), make the bag positive (tumor sample), whereas the negative TCRs are generated from host immune responses that are not triggered by cancer, but

by other physiological processes such as auto-immune diseases, infection, etc.

An alternative formulation is based on the concept of primary instances, first introduced by Park et al. [12]. In this vein, a bag label, whether it is positive or negative, is determined by a (small) number of instances in this bag (called primary instances), while all other instances are irrelevant. That is, the bag-level classification function $F(X)$ can be written into $F(X^*)$, where X^* is a subset of $\{x_1, \ldots, x_m\}$, representing the primary instances of the bag. Thus, the bag classification remains unchanged if non-primary instances are removed from the bags, though one has no clue about their presence in advance. It is important to note that whether an instance is primary or not is an indicator rather than an instance label. Park et al. [12] makes a simplifying assumption that one bag has only one primary instance based on the finding that only a very small portion of instances are responsible for bag-level responses in their application. In general situations, multiple primary instances should be allowed. Analogous to WR, the proportion of primary instances (PPI) is defined as the number of primary instances divided by the bag size (i.e., the number of instances in the bag), describing the proportion of "responsible" instances of a bag. Another classical assumption in the MI literature, known as the collective assumption, states that each instance contributes equally and independently so that the label of a bag is determined by all its instances collectively [41]. Obviously, this assumption fits in the PPI framework (with PPI $= 100\%$) rather than the WR framework. We note that the collective assumption is not suitable for our application, because not all TCRs are equally important in the biological context. In distinguishing tumor from normal samples, the distinction between primary and non-primary instances seems to be reasonable: on one hand, there could be abundant T cells with irrelevant TCRs generated by the host immune system that are by-standers naive to any antigenic events; on the other hand, there could be "important" TCRs that serve as signatures of immune responses against tumor (positive bags), or other diseases that trigger immune responses in non-tumourous individuals (negative bags). These TCRs are the primary instances of the bags. Ideally, under the PPI framework, the first step of MIL is to identify the primary instances in each bag, as these instances are the only ones that are responsible for the bag label. However, this formulation is fairly new, and existing MIL methods for classification do not possess the capability of formal identification of primary instances. Thus, the performance of the methods needs to be validated for MI data generated under the PPI framework, which, we believe, is a reasonable assumption for many applications in the real world. This would help answer an important question whether new MIL methodologies need to be specifically developed under the PPI framework to better accommodate such data.

As discussed above, both WR and PPI formulations are feasible in our application. For the purpose of performance evaluation of existing MIL methods, we will consider both data generation mechanisms in our simulation and compare the results with those obtained in real data analysis. Other factors, such as sample size, bag size, number of features, and proportion of positive bags, can potentially affect the performance of MIL methods. We will consider such factors collectively to guide the simulation design and the subsequent selection of appropriate MIL methods for our new application in cancer detection.

3. Review of selected MIL methods

As we focus on binary classification in our application, we do not consider MIL methods that address multi-class classification, regression, ranking, or clustering. Eligible MIL methods should be able to accommodate the problem characteristics and related concepts as discussed in Section 2.2. According to Amores [19], MIL methods can be grouped into one of three categories: instance-space (IS), bag-space (BS), and embedded-space (ES) methods. This categorization is based on how a method extracts and exploits information from the MI data. For IS methods, the learning process occurs at the instance level, where $f(x)$ is trained to separate the instances in positive bags from those in negative ones; instance-level scores produced by $f(x)$ are then combined to create a bag-level classifier $F(X)$ according to a reasonable MI assumption. Thus, IS methods consider the characteristics of individual instances and ignore more global characteristics of the entire bag. By contrast, both BS and ES methods treat each bag as a whole entity, and train $F(X)$ utilizing the global, bag-level information. Specifically, BS methods attempt to measure the distance or similarity between each pair of bags and predict the bag labels directly using distance- or kernel-based classifiers such as k-Nearest Neighbors (kNN) and Support Vector Machine (SVM), while ES methods employ a mapping function to embed multiple instances of a bag into a single "meta" instance defined on a new feature space, and then make direct bag-level prediction using standard classifiers.

Both BS and ES methods seek to represent each bag using a single instance defined on a new feature space. As we shall see in subsequent sections, there are various ways to map the original feature space to a new feature space by using either a distance or kernel function. Since IS methods are applied to MI data in its original representation, the dimensionality is the same as the number of features. For BS and ES methods, the dimensionality depends on the new feature space created from the original feature space.

Under the WR framework, IS methods can naturally deal with both bag and instance classification while BS and ES methods usually do not directly classify instances. Under the PPI framework, IS methods are less appropriate as the non-primary instances introduce irrelevant information to the bag. However, the selected BS and ES methods might still be appropriate for the task of bag classification, as the information of primary instances of each bag could be utilized by the flexible embedding/summary behaviors of these methods. Therefore, we anticipate that IS methods perform poorly under the PPI mechanism, which is later confirmed by our simulation (Fig. 3).

Table 1 displays the 17 methods selected for our application, including seven IS, five BS and five ES methods. Methods that can perform instance classification are highlighted in italics.

3.1. Instance-space methods

Beginning with propagating bag labels to the corresponding instances, IS methods ignore bag structures and build classifiers at the instance level. Bag labels are then obtained by aggregating instance prediction based on a suitable MI assumption, such as the standard assumption, the collective assumption (e.g., sum or average of individual instance predications in a bag), and the maximum or minimum of the instance predications. For each of the six

Table 1
The selected MIL methods for cancer detection using TCR sequences. Those that can perform instance classification are highlighted in italics.

	MIL Methods						
IS	*EMDD*	*MI-SVM*	*mi-SVM*	*SI-SVM*	*SI-kNN*	*MILBoost*	*mi-Net*
BS	*Ck*NN	NSK-SVM	EMD-SVM	miGraph	MInD		
ES	*MILES*	BoW	CCE	MI-Net	*ADeep*		

IS methods included, we describe how instance classification is performed below.

EMDD [42]: Expectation–Maximization Diverse Density is a generalization of the Diverse Density (DD) algorithm [2], which aims to identify a point with the maximum DD in the feature space that is close to as many different positive bags as possible, while staying as far from the negative bags as possible. EMDD searches for the maximum DD point via the Expectation–Maximization (EM) algorithm and instance classification is made based on the distance from the maximum DD point.

mi-SVM and MI-SVM [3]: Both methods are extended from SVM, known as a maximum-margin classifier, to fit in the MI setting. For binary classification, SVM finds a hyperplane that yields the largest separation (or margin) between the two classes. mi-SVM assigns negative labels to all instances in negative bags but treats labels of instances in positive bags as unknown. Then a soft-margin criterion, defined at the instance level, is maximized jointly over the hyperplanes and unobserved instance labels in positive bags such that all instances in every negative bag are located on one side of the hyperplane and at least one instance in every positive bag is located on the other side of the hyperplane. In each iteration, an SVM classifier is built and instance labels are re-assigned. The SVM is then retrained to refine the decision boundary using the newly assigned labels until the imputed labels do not change further. Instead of maximizing the instance-level margin, MI-SVM represents each bag by one representative instance of the bag and maximizes the bag-level margin; that is, the margin of a positive bag is defined by the margin of the "most positive" instance, while the margin of a negative bag is defined by the "least negative" instance. An SVM classifier is built when the representative instance remains unchanged in each bag. The authors suggested that, if one aims to make an accurate instance classification, mi-SVM is preferable; otherwise, MI-SVM is more appropriate.

SI-SVM [43] and SI-*k*NN [22]: These methods train vanilla (single-instance) supervised classifiers on MI data by completely discarding the bag-membership information of instances. In their implementation, each instance inherits the bag label and the SVM and *k*NN classifiers are optimized on the reduced (single-instance) problem.

MILBoost [44]: This method classifies each instance individually by a linear combination of decision dumps (i.e., 1-level decision trees) whose performance may only be slightly better than random guessing. The weak classifiers are then combined to minimize the bag-level loss function (e.g., the negative log likelihood), using gradient boosting [45].

mi-Net [46]: Named by Wang et al. [46], mi-Net represents multiple instance neural networks (MINNs) that first predict the probability of positiveness for each instance and then employ an MIL pooling layer to aggregate instance-level probabilities to produce bag-level probabilities. Suppose an MINN is composed of L layers. At the beginning, each instance is fed into several fully-connected (FC) layers with an activation function. After instance-level probabilities are predicted from the last FC layer (i.e., the $(L-1)$ th layer of the MINN), the bag-level probability is obtained from the last layer for each bag using an MIL pooling function (such as maximum pooling, mean pooling, and log-sum-exp pooling).

3.2. Bag-space methods

Unlike IS methods that ignore the bag structure during the learning process, BS methods learn the distance or similarity between each pair of bags. In short, BS methods use a suitable distance or kernel function to embed the bags using their member instances, and then employ a standard supervised learning method, such as *k*NN and SVM, to learn the bag-to-bag relationship. The following BS methods are considered for our application.

C*k*NN [47]: C*k*NN (Citation-*k*NN) is a variant of SI-*k*NN adapted to MI data, which uses the minimal Hausdorff distance to calculate the distance between a pair of bags so that the resulting distance is robust to extreme instance values. The authors also introduced so-called "reference" and "citer", where references are the nearest neighbors of a given bag and citers are bags that consider the given bag as their nearest neighbor. By using references and citers collectively, a bag is labeled as positive if there are more positive bags than negative bags among its references and citers. For example, suppose a bag has $R = R_+ + R_-$ references and $C = C_+ + C_-$ citers, where the subscript indicates the bag label. The target bag is thus identified to be positive if $R_+ + C_+ > R_- + C_-$. If there is a tie, the bag is assigned to the negative class to mitigate the tendency to produce false positives that occur more frequently than false negatives in MI applications.

NSK-SVM [48]: NSK-SVM is an extended version of kernel methods, in which a normalized set kernel (NSK) is proposed for MI data. Specifically, the set kernel is defined on bags and derived from a chosen instance-level kernel. Matching kernel, polynomial kernel, and radial basis function kernel are common choices. To reduce the effect of varying bag sizes, normalization is critical and is achieved by averaging the pairwise distances between all instances contained in two bags. Subsequently, an SVM using the normalized set kernel is built to predict bag labels.

EMD-SVM [49]: The proposed approach employs Earth Mover's Distance (EMD) [50] to measure the similarity between any two bags (say i and i'). EMD can be defined as a weighted sum of the ground distances between all pairs of instances (j,j'), where instance j (j') is from bag i (i'), respectively. In Zhang et al. [49], the ground distance measure is chosen to be the Euclidean distance and the weights are obtained by solving a linear programming problem. For bag classification, an SVM is used after transforming the calculated distances to a Gaussian kernel function.

miGraph [51]: Motivated by an observation made in Zhou and Xu [52] that instances are rarely independently and identically distributed (i.i.d.) in a bag, the authors propose miGraph for bag classification that can make use of the relations among instances by treating instances as inter-correlated components of the bag. The miGraph method represents each bag by a graph, where its nodes are the instances. An edge exists between a pair of instances if their Gaussian distance is smaller than some threshold (e.g., the average distance in the bag). Since instances are potentially dependent, their weights contributing to the bag classification are adjusted by cliques identified in the graph. After representing all bags by their corresponding graphs, an SVM with a graph kernel (constructed by using instance weights) is used to perform the classification based on between-bag similarity. This method can also handle i.i.d. instances by using an identity edge matrix (i.e., no edge between any two instances).

MInD [20]: Multiple Instance learning with bag Dissimilarities (MInD) uses bag dissimilarities as features, obtained by representing each bag by a vector of its dissimilarities to the other bags in the training set. An SVM is then trained for bag classification. The authors recommend using the *meanmin* function as the bag dissimilarity measure given its superior performance in numerical experiments. Specifically, the dissimilarity from bag i to bag i' is defined as $D_{i,i'} = \frac{1}{m_i}\sum_j \min_{j'} d(x_{ij}, x_{i'j'})$, an average over the minimum squared Euclidean distances from each instance in bag i (with m_i instances in total) to instances in bag i'. As a result, the dissimilarity matrix is asymmetric (i.e., $D_{i,i'} \neq D_{i',i}$), which is more generalized compared with a symmetric representation.

3.3. Embedded-space methods

As in BS methods, ES methods extract information contained in MI data at the bag level and transform an MI problem to a standard supervised learning problem by summarizing a bag using a single feature vector. However, ES methods focus on instance embedding. We discuss three methods, each using a different strategy to embed instances to a new feature space.

BoW [19]: Bag-of-Words (BoW) provides a general framework to represent the bag-instance relationship. Under MIL, the training instances are used to build a word dictionary (or vocabulary). A bag can thus be represented by a histogram over the dictionary, which forms a new feature space. An SVM is then used to make bag classification using the new features.

CCE [53]: Constructive Clustering based Ensemble (CCE) first assigns all instances in a training set into C clusters using the k-means clustering method, and then represents each bag by a binary feature vector of length C: if the bag has at least one instance belonging to cluster c, the corresponding cth feature component is 1, and 0 otherwise. With new bag-level features created, an SVM can be built for bag classification. Since there is no restrictions on the choice of C, it is advised to train several classifiers based on different clustering results and combine their predictions via a majority vote. In this sense, CCE also takes advantage of ensemble learning. When a new bag is given for classification, CCE re-represents it through querying the clustering results, and then feeds the generated feature vectors to the ensemble classifier to predict the bag label. Note that in CCE, k-means, SVM, and majority voting can be replaced by any other algorithms for clustering, classification and ensemble, respectively.

MILES [54]: Multiple-Instance Learning via Embedded instance Selection (MILES) assumes a subset of instances is responsible for bag labels. In the embedding step, each bag is mapped into a new feature space, represented by a vector of similarity scores between the current bag and the set of instances from all the bags. The dimensionality of the new feature space is thereby equal to the total number of instances, which can potentially be large, resulting in high-dimensional features, including those redundant or irrelevant. Therefore, an SVM with LASSO penalty [55] is applied to select important features as well as construct classifiers simultaneously. In addition, MILES can also be used for instance classification by calculating the contribution of instances to the bag classification based on a given threshold. Unlike other MIL methods we have discussed, the design of MILES is compatible with the PPI framework.

MI-Net [46]: Unlike mi-Net that focuses on calculating instance-level probabilities, MI-Net is the first MINN method in the ES category, which strives to learn bag representation from instance features and generates bag classification directly. Suppose an MINN has L layers. In MI-Net, after several FC layers, the MIL pooling process aggregates instances in one bag into a single feature vector as a bag representation, which occurs in the $(L-1)$ th layer. The last FC layer (i.e., the Lth layer) takes the bag representation as input and outputs bag-level probabilities with a sigmoid activation function. Besides the above basic version, there are two variants of MI-Net proposed in [46], one adding deep supervision [56] and the other considering residual connections [57], which can improve the performance sometimes.

ADeep [58]: Besides mi-Net and MI-Net, Attention-based Deep MIL (ADeep) is an MINN method. It modifies the ES approach to achieve better interpretability by using a novel MIL pooling method that relies on a special case of the attention mechanism [59], where all instances are assumed independent. Unlike traditional pooling operators such as max and mean that are predefined and non-trainable, a weighted average of instances is proposed, where the weights are determined by a two-layer neural network and sum to 1 so that they are not affected by the size of a bag. Naturally, instances that are likely to be positive receive higher weights in a bag, rendering more interpretable results. In this sense, ADeep links the ES approach to the IS approach by providing instance weights as a proxy to instance probabilities.

3.4. Implementation

The MATLAB "MILSurvey" toolbox is made available online by Carbonneau et al. [22]. We use this software package to implement the MIL methods covered in Sections 3.1–3.3 except for the three MINN methods (mi-Net, MI-Net, and ADeep) on simulated MI data generated under either WR or PPI framework. We use Python code available from Wang et al. [46] to implement mi-Net and MI-Net (the basic version). Due to lack of instructions on the code usage and data input format, we were not able to implement ADeep. For each of the methods implemented, the default setting is used in our evaluation. For example, for SVM-based methods, we use the default kernel function. In cross validation, default ranges of values for tuning parameters are used. For MINNs, default choices of activation function, number of layers, number of neurons, and MIL pooling method are implemented. Each selected IS method predicts bag labels from the predicted instance labels based on the standard MI assumption mentioned in Section 2.2. We also refer readers to the GitHub link https://github.com/danyixiong/MIL_Comparative_Study for more detail on implementation.

4. Simulation

We evaluate the performance of 16 MIL methods under various simulated scenarios, which attempt to mimic realistic situations in our cancer screening application using TCR sequences by varying key factors that can potentially affect the performance. Consisting of amino acid sequences, TCRs are essentially text strings that need to be converted to numeric values before applying MIL methods. In the analysis of TCGA data (Section 5), TCRs are converted into numeric vectors by the Briseis encoder [38]. In our simulation, rather than generating TCR sequences, we directly generate numeric values for instances to simplify the process. We adopt two data generation models based on different assumptions about the instance-to-bag relationship. Model I adopts the standard assumption under the WR framework; that is, a positive bag has at least one positive instance and a negative bag only has negative instances. Model II adopts the PPI mechanism, assuming that only the primary instances are responsible for the bag labels. Thus, WR/PPI plays a key role in bag composition under model I/II. In addition, for both models, we examine the impact of sample size n, bag size m, number of features d, and proportion of positive bags p_+ on the performance of the methods. For simplicity, we assume different bags in one dataset have a constant number of instances and constant WR/PPI.

We randomly generate 100 replication datasets under each scenario. For each replicate, we train the methods on the training set (70%) and evaluate their performance on the test set (30%). We evaluate the performance using the Area Under the Receiver Operating Characteristic Curve (AUROC). Since the IS method MILBoost performs poorly under both models, we exclude it when displaying results for better visibility.

4.1. Simulation under model I

Based on model I, each instance has a label. We separately generate positive and negative instances from two different Gaussian mixture distributions. In our real data application, non-cancer-specific TCRs (negative instances) are usually more diverse than cancer-specific TCRs (positive instances) due to the existence of

diverse antigens from bacteria, virus, and antigens caused by autoimmune diseases, infections, etc. [60,61]. Therefore, compared to positive instances, negative instances are simulated from a distribution with a wider dynamic range. Besides the factors n, m, d, p_+ and WR mentioned above, we consider varying the number of components in the positive instance distribution (N_+) as well.

For each positive bag, we generate $\lceil m \times \mathrm{WR} \rceil$ positive instances from a Gaussian mixture with N_+ components and $m - \lceil m \times \mathrm{WR} \rceil$ negative instances from a Gaussian mixture with 30 components. For each negative bag, all m instances are negative and hence generated from the same Gaussian mixture with 30 components. The feature dimensionality is d and the mixing probability is uniform for each component in either Gaussian mixture. We then simulate mean vectors and covariance matrices for the mixture distributions. For each component of the Gaussian mixture for positive instances, a d-dimensional mean vector is randomly generated from a uniform distribution $U[-5, 5]$. For each component of the Gaussian mixture for negative instances, a d-dimensional mean vector is randomly generated from a uniform distribution $U[-10, 10]$. The covariance matrices of each component for positive and negative instances are identity matrices with the scale parameter being 2.5 and 5, respectively. Thus, the features are independently generated. We vary n = 50, 100, 200, 400, 600; m = 5, 10, 20, 40, 60; d = 2, 15, 30, 45, 60; p_+ = 0.1, 0.2, 0.3, 0.4, 0.5; N_+ = 1, 8, 15, 22, 30; and WR = 0.05, 0.25, 0.5, 0.75, 1 and assess their influence on performing multiple instance classification. To reduce the workload of simulation, not all combinations of the 6 parameters are evaluated. Instead, we vary one of them at a time while fixing all others at the basic setting, where n = 200. m = 20, d = 30, p_+ = 0.3, N_+ = 15, and WR = 0.5.

Fig. 2 shows bag classification performance of different MIL methods in terms of mean AUROC under various simulation scenarios. Overall, all BS methods except for C*k*NN perform fairly well in most scenarios, (closely) followed by the three ES methods. Among all IS methods, mi-Net and three SVM-based methods (MI-SVM, mi-SVM, and SI-SVM) outperform the others. Their green lines are virtually invisible because they overlap with those of the top performing methods and so are covered by the blue or magenta lines. We note that IS methods appear to be more sensitive to the change of the factors under the WR framework, as opposed to BS and ES methods.

Next, we discuss how each factor affects the performance of MIL methods excluding MILBoost. First, the performance tends to improve with an increased sample size (n), especially for EMDD and SI-*k*NN. Meanwhile, mi-Net and three SVM-based IS methods and all BS and ES methods perform adequately well even when n is as small as 50, and so as n increases, their improvement is not as obvious. Secondly, as the bag size (m) increases, BoW (an ES method) has improved performance, while EMDD has decreased performance. The performance of the other methods is not much affected by increased m. Thirdly, as the proportion of positive bags p_+ increases towards 50%, the performance of EMDD and SI-*k*NN substantially improves and the performance of mi-Net and MI-Net shows non-monotonic patterns. As the number of components in the positive instance distribution N_+ increases, the performance of EMDD worsens. Other methods, especially the BS methods, perform adequately well across these scenarios. We now discuss the influence of WR on the performance of the methods, where WR = 0.05 represents the scenario with only one positive instance in a positive bag. As the WR increases, these methods perform better until AUROC gets close to 100% and there is not much room left for further improvement. When WR = 0.05, mi-Net has 100% AUROC and the BS method MInD has nearly 90% AUROC. The BS methods except MInD exhibit the most dramatic improvement when the WR is changed from 0.05 to 0.25. The two IS methods, EMDD and SI-*k*NN, increases at a slower pace than the other methods. Lastly, we find that the number of features d is another factor which can substantially affect the performance of many MIL methods, including the six IS methods, MI-Net and C*k*NN. The most dramatic improvement for these methods except EMDD occurs when d increases from 2 to 15, and with 30 features or more, their AUROC values are close to 100%. Meanwhile, all BS and ES methods except C*k*NN and MI-Net have good performance (AUROC above 90%) even when d is 2.

Focusing our evaluation of the MIL methods on their prediction capability for the minority class (the positive bags in this case), we observe that their performance evaluated by AUPRC (area under the precision-recall curve) maintains virtually the same ranking as evaluated by using AUROC. An MIL method with higher AUROC has higher AUPRC in general, as showed in Supplementary Fig. 1.

As discussed in Section 3, MILES (an ES method) and all IS methods can be used to classify instances. Supplementary Fig. 3 shows instance classification performance of six methods in terms of mean AUROC under various simulation scenarios. Besides MILBoost, we exclude results from mi-Net, whose code for performing instance classification is not available. We find that IS methods show better performance in instance classification than in bag classification. Furthermore, though MILES can also perform instance classification, its performance is worse than these IS methods except when the number of features d is 2. As an ES method, MILES performs better in bag classification. Overall, for instance classification, regardless of the number of features, SI-*k*NN performs the best.

4.2. Simulation under model II

In addition to the factors shared with model I (n, m, d, p_+), we consider varying $\overline{\mathrm{PPI}}$ (i.e., mean proportion of primary instances) for model II. For instance j in bag i, let x_{ijk} denote its kth covariate and $\delta_{ij} \in \{0, 1\}$ be a binary variable with $\delta_{ij} = 1$ indicating this instance is primary and 0 otherwise. Each x_{ijk} is independently generated from a uniform distribution $U[l, u]$ with $l < u$. We simulate δ_{ij} from a Bernoulli distribution $Ber(p_{ij})$, with $p_{ij} \equiv \Pr(\delta_{ij} = 1) = \Phi\left(b_0 + \sum_{r=1}^{d} x_{ijr} b_r\right)$, where $\Phi(\cdot)$ is the standard normal cumulative distribution function (CDF), and b_0 and b_r for $r = 1, \ldots, d$ are regression coefficients in the probit regression model for p_{ij}. Further, we simulate the bag label Y_i from $Ber(\pi_i)$, with $\pi_i \equiv \Pr(\mathrm{Y_i} = 1) = \Phi\left(\beta_0 + \sum_{j=1}^{m} \delta_{ij} \sum_{r=1}^{d} x_{ijr} \beta_r\right)$, where β_0 and β_r for $r = 1, \ldots, d$ are regression coefficients associated with the probit model for π_i. In case where $\delta_{ij} = 0$ for all $j = 1, \ldots, m$, we simply generate Y_i from $Ber(\Phi(\beta_0))$. We adjust the intercepts b_0 and β_0 to vary values of $\overline{\mathrm{PPI}}$ and proportion of positive bags p_+, respectively. We set $l = -10, u = 10, b_j = 2, \beta_j = -1$ $\forall j, \overline{\mathrm{PPI}} = 0.05, 0.25, 0.5, 0.75, 1$ and use the same settings as in model I for n, m, d, and p_+. Again, we employ the vary-one-at-a-time strategy to reduce the work load in this simulation, where the basic setting has $n = 200, m = 20, d = 30, p_+ = 0.3$, and $\overline{\mathrm{PPI}} = 0.5$.

The performance of the methods under various simulation scenarios is shown in Fig. 3. First, the relative performance of the methods is quite consistent across different scenarios. NSK-SVM and EMD-SVM are the top two performers and NSK-SVM outperforms the latter in nearly all the scenarios. MInD wins the third place, followed by MILES and then miGraph. Among the remaining nine methods, the IS methods and C*k*NN have poor performance in all the scenarios, with AUROC close to 50%, which is only slightly better than random guessing; BoW and CCE also perform poorly

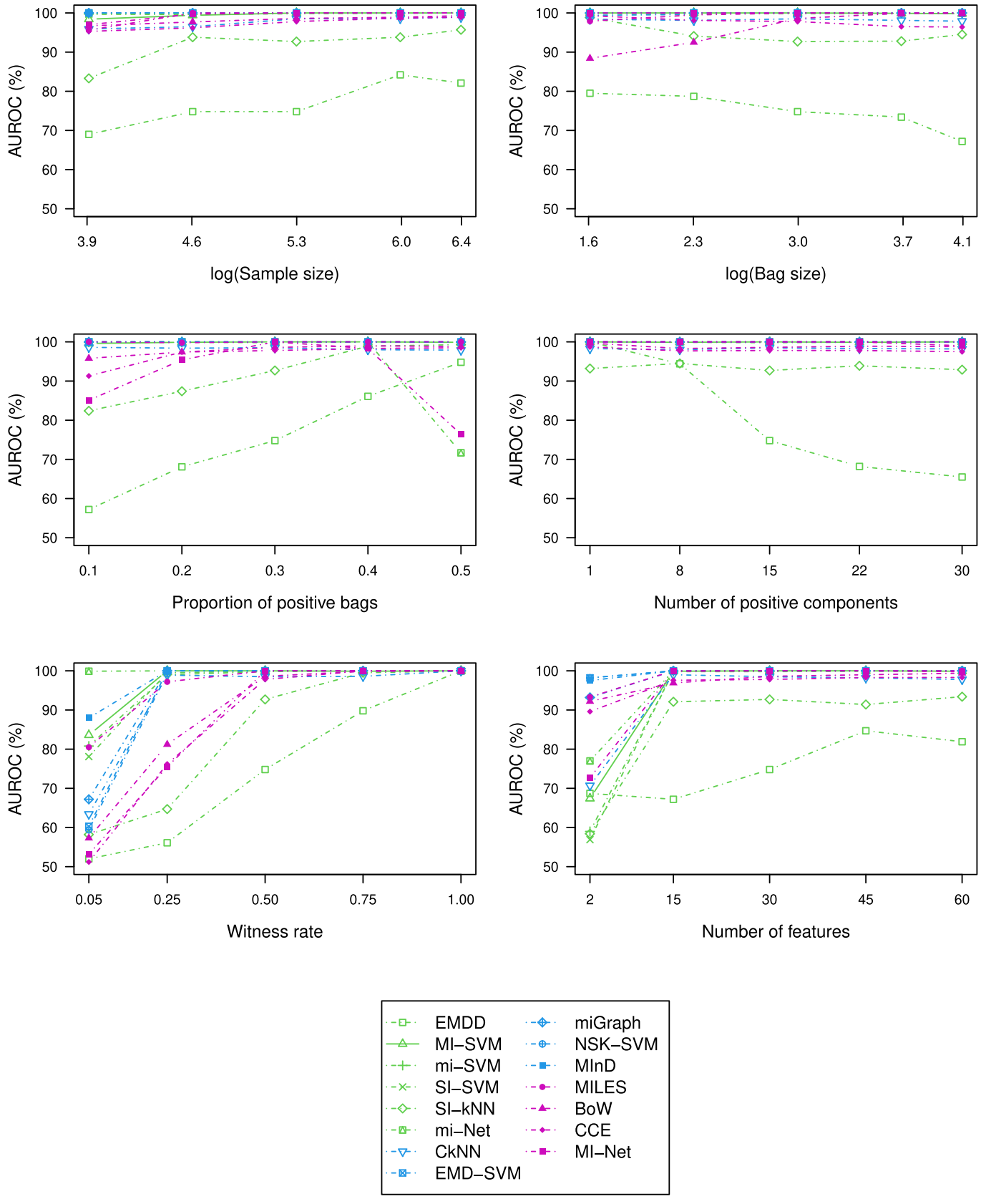


Fig. 2. Mean AUROC (%) of bag classification using different MIL methods, evaluated on simulation scenarios each with 100 replicates generated under model I. IS/BS/ES methods are distinguished by green, blue, and magenta lines.

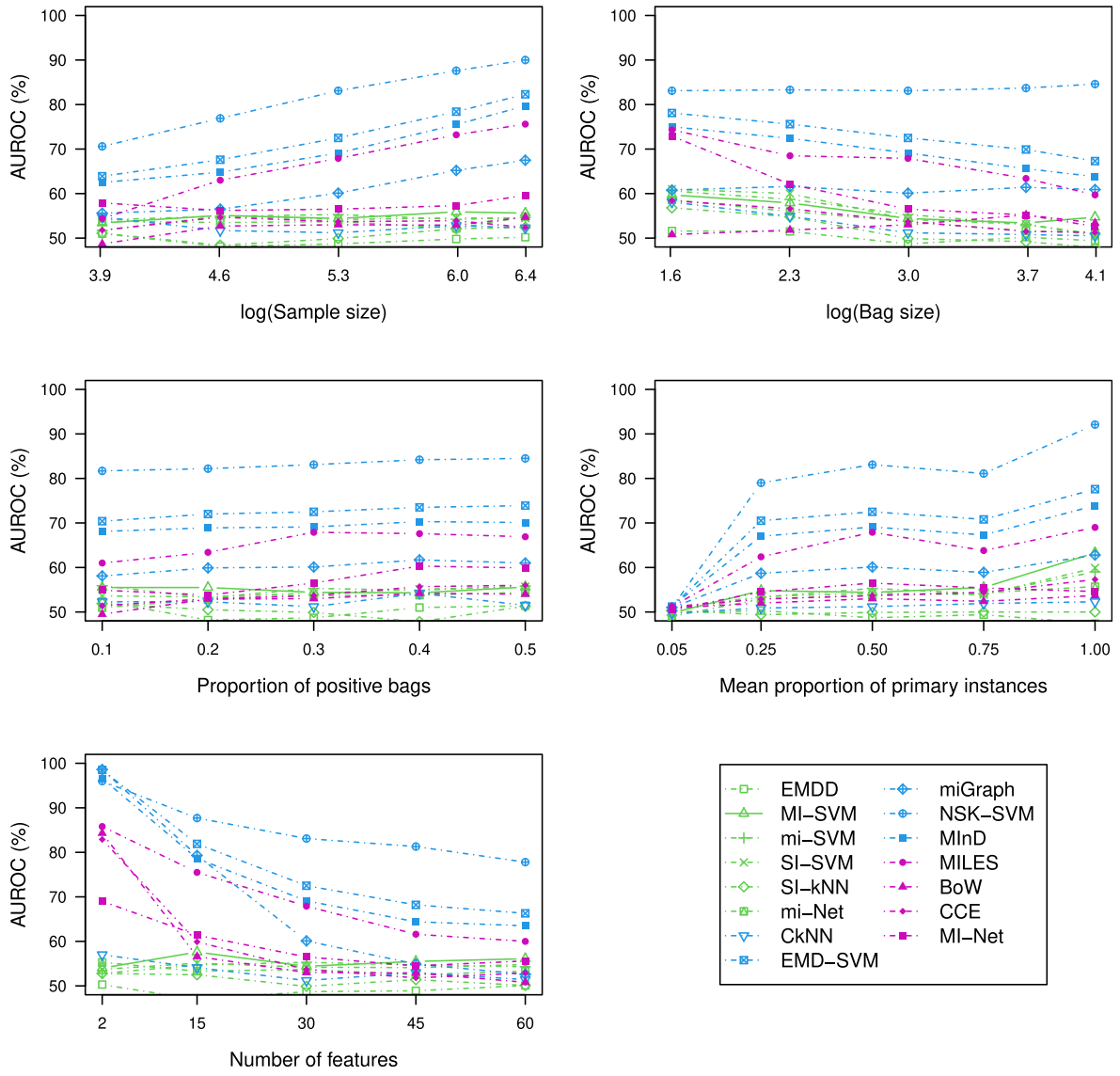


Fig. 3. Mean AUROC (%) of bag classification using different MIL methods, evaluated on simulation scenarios each with 100 replicates generated under model II. IS/BS/ES methods are distinguished by green, blue, and magenta lines.

except for the scenario with $d = 2$. Second, the performance varies with a wider range among BS and ES methods, as opposed to IS methods. Overall, the performance of all methods under model II is (much) worse than that under model I, which is as expected, since existing methods are not equipped with the capacity to handle data generated under the PPI framework.

Excluding all the IS methods and C*k*NN, which have steadily poor performance, we discuss the impact of each factor on the performance of the remaining MIL methods. First, increasing sample size n tends to improve the performance. Secondly, as the bag size m increases, the performance of EMD-SVM, MInD, MI-Net, and MILES decreases, while the other methods are not sensitive to the change. In particular, NSK-SVM maintains good performance with AUROC above 80% regardless of the bag size. Thirdly, the proportion of positive bags p_+ appears not to have much impact on the performance. Further, when $\overline{\text{PPI}}$ increases, most methods show higher AUROC by capturing the increased amount of useful information. Greater improvement is observed when $\overline{\text{PPI}}$ changes from 0.05 to 0.25. Lastly, all the methods have worse performance as the number of features d increases. Steeper drops in AUROC occur when d increases from 2 to 15. Recall that under model I, the performance of the methods shows an increasing pattern overall. As d goes up, the signal in the simulated data becomes stronger in general, no matter which model is used for data generation. As the PPI framework is relatively new, none of the methods were specifically designed for it; instead, many were designed under the WR framework. Thus, these methods are able to capture the stronger signal under the WR framework as d goes up but not under the PPI framework.

In terms of AUPRC, as showed in Supplementary Fig. 2, observations about the relative performance of the MIL methods and the impact of each factor on the performance are similar to those from AUROC with one exception: the performance on correct prediction for positive bags has improved as the proportion of positive bags increases.

4.3. Computation time

We provide the runtime information of each method under the basic setting of each model in Table 2. Fourteen methods are run

Table 2
Average computation time (with standard error) in seconds for each MIL method based on 20 datasets under the basic setting of each model. Clock time is counted from loading data to producing classification results.

IS methods	MILBoost	SI-*k*NN	SI-SVM	EMDD	mi-SVM	MI-SVM	mi-Net
Model I (WR)	9 (0.05)	13 (0.06)	17 (0.05)	24 (0.22)	22 (1.14)	33 (2.75)	13 (0.07)
Model II (PPI)	10 (0.12)	9 (0.03)	18 (4.21)	18 (1.00)	14 (0.24)	13 (0.27)	13 (0.09)
BS methods	MInD	C*k*NN	EMD-SVM	miGraph	NSK-SVM		
Model I (WR)	9 (0.03)	21 (0.05)	3 (0.50)	77 (0.38)	80 (0.08)		
Model II (PPI)	13 (0.17)	9 (0.04)	3 (0.61)	20 (0.31)	21 (0.20)		
ES methods	BoW	CCE	MILES	MI-Net			
Model I (WR)	60 (21.84)	14 (0.36)	42 (0.21)	14 (0.07)			
Model II (PPI)	20 (3.78)	14 (0.14)	20 (0.22)	13 (0.04)			

on MATLAB 2019b GUI from a computing cluster while MI-Net and mi-Net are run in a Python environment. The average computation time and its standard error are provided based on 20 datasets simulated under the basic setting of each model. Overall, applying MIL methods to data generated under model I (the WR framework) takes longer time than to data generated under model II (the PPI framework). Furthermore, NSK-SVM, miGraph, BoW, and MILES are more time consuming than the other methods, regardless of the model used for data generation.

5. Real data examples

5.1. TCGA data

As a landmark cancer genomics program, TCGA characterized over 20,000 primary and metastatic cancer samples on over thirty cancer types with matched adjacent normal tissues. Supplementary Fig. 4 shows the number of tumor versus normal tissue samples for each of the cancer types. In the TCGA data, the number of positive bags (tumor samples) is much greater than that of negative bags (normal tissue samples). This is because TCGA is mainly focused on studying cancer patients. We analyze the RNA sequences of samples from ten cancer types in the TCGA database, including skin cutaneous melanoma (SKCM), kidney renal clear cell carcinoma (KIRC), lung adenocarcinoma (LUAD), lung squamous cell carcinoma (LUSC), lymphoid neoplasm diffuse large B-cell lymphoma (DLBC), breast invasive carcinoma (BRCA), stomach adenocarcinoma (STAD), ovarian serous cystadenocarcinoma (OV), thymoma (THYM), and esophageal carcinoma (ESCA) [62–64]. These cancer types are selected as they have reasonably large sample sizes (i.e., the number of normal + tumor tissue samples) and bag sizes (i.e., the number of TCRs in one sample).

In real applications of cancer screening, there are supposed to be many more samples without cancer than those with cancer. To adjust for oversampling (more positive bags than negative bags) in TCGA data, we randomly sample positive bags so that the resulting dataset only includes a subset of positive bags for each cancer type. Furthermore, we combine all normal tissue samples available from the 30+ cancer types in the TCGA data to increase the number of negative bags to 405. Mixing negative bags across datasets for different cancer types is reasonable because the characteristics of normal tissue samples should be similar across patients.

TCR sequences were reconstructed by MiTCR from the TCGA RNA-sequencing data. MiTCR is a commonly used software for reconstructingTCR sequences from next generation sequencing data [37]. MiTCR also records the number (abundance) of each unique TCR in each sample (bag). We exclude TCRs whose abundance is 1, because they are most likely the ones that have not been exposed to any antigens. We randomly sample 50% of the 405 negative bags (i.e., 202 normal tissues samples) to reduce the computation time and for each of the selected cancer types, we further downsample positive bags so that the corresponding data contain ∼10% positive bags. As a result, we have an equal number of positive bags (23) and the total sample size is the same (225) for all selected cancer types. As pointed by one reviewer, in the literature it is often preferred to apply MIL methods to balanced data. Thus, we also include analysis on sampled TCGA datasets with 50% positive and 50% negative bags: for DLBC, THYM, and ESCA, due to a small number of positives (Supplementary Fig. 4), the sample sizes are 90, 216, and 332, respectively; for each of the remaining cancers, the total sample size is 404. Table 3 shows descriptive statistics including the sample size and the number of instances for selected cancer types after data pre-processing. We further embed each TCR sequence into a 30-dimensional numeric vector using the Briseis encoder, as mentioned in Section [subsec:TCR-sequencing-data]. In addition, we include log-abundance as an additional feature for each TCR sequence.

5.2. Analysis results

We apply the 16 MIL methods to classify tumor and normal tissue samples for the ten cancer types from TCGA. For model training

Table 3
TCGA data: descriptive statistics including the sample size and bag size (i.e., the number of instances per bag) for selected cancer types.

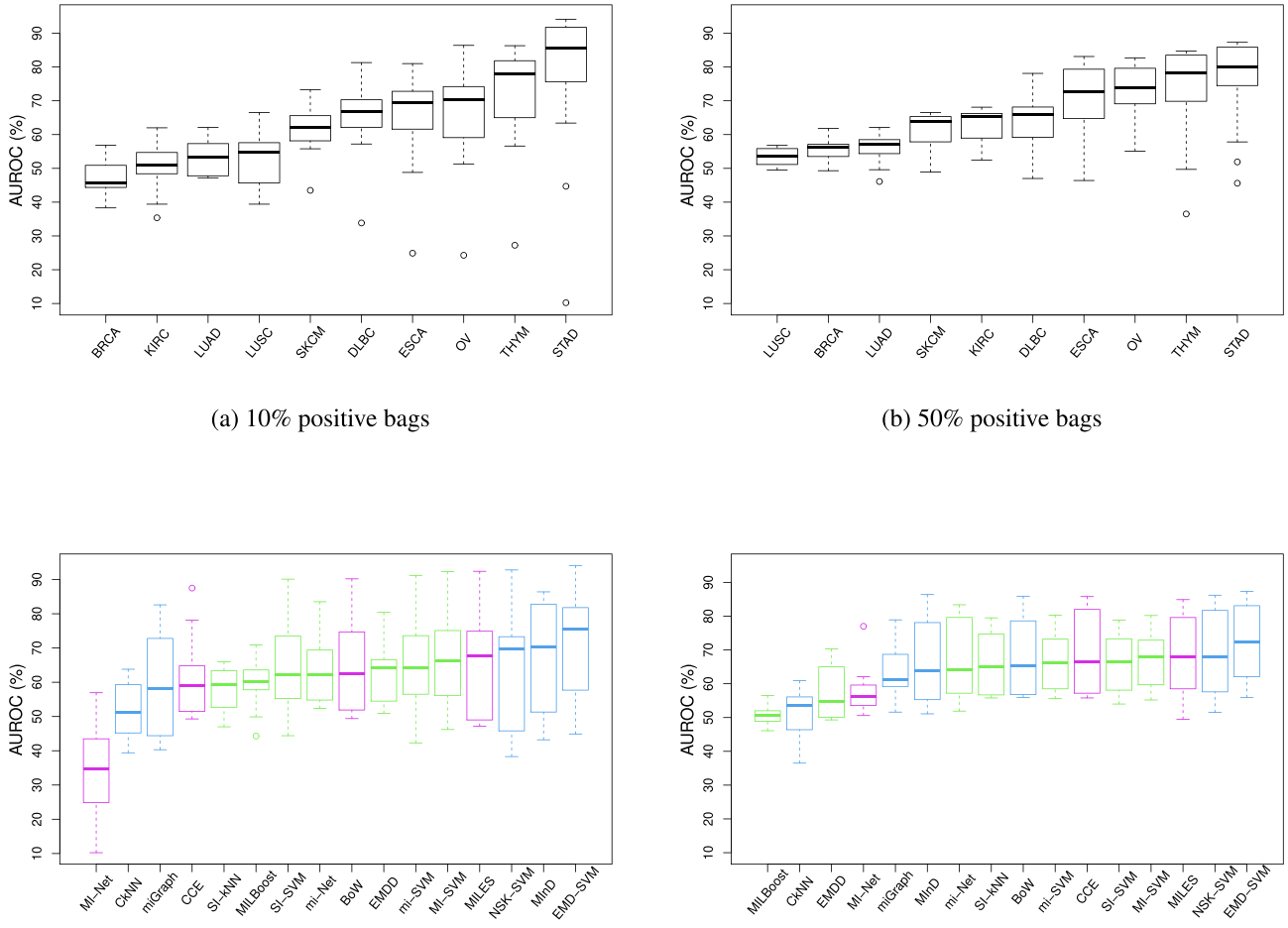
	Sample size			Bag size		
Cancer type	Total		Mean (SD)	Total	Mean (SD)	Total
Proportion of positive bags	10%	50%	10%		50%	
DLBC	225	90	6.4 (13.8)	1446	11.8 (21.8)	1063
THYM	225	216	6.3 (14.2)	1421	15.2 (24.5)	3277
ESCA	225	332	8.8 (24.5)	1979	10.2 (20.6)	3380
BRCA	225	404	5.3 (12.9)	1200	5.6 (12.6)	2255
KIRC	225	404	6.0 (17.0)	1347	6.2 (10.3)	2518
LUAD	225	404	4.5 (14.5)	1018	4.9 (12.9)	1974
LUSC	225	404	4.9 (10.7)	1093	4.0 (6.3)	1622
OV	225	404	5.6 (11.1)	1263	9.1 (17.3)	3670
SKCM	225	404	5.8 (12.0)	1306	6.2 (13.2)	2515
STAD	225	404	9.9 (23.9)	2221	18.3 (31.9)	7401

and validation, a nested cross-validation (CV) procedure [65,22] is deployed, in which the model is tuned (if the hyperparameters are optimized over a range of values) in the inner layer CV and the performance of fitted model is evaluated in the outer layer CV. In implementation, both inner and outer layers have ten folds. The average performance in terms of AUROC of each method is calculated from nested cross-validation.

Fig. 4a shows boxplots of mean AUROC by cancer type for different methods using the imbalanced TCGA data, arranged in an increasing order of the median AUROC of each boxplot. Evidently, the performance of the methods depends on cancer type. For example, all methods perform poorly for BRCA, KIRC, LUAD and LUSC, all with the 75th percentile of mean AUROC below 60% and the maximum below 70%. On the other hand, for STAD, most MIL methods perform well and achieve AUROC at least 80%, with the median around 85%. The median is about 80% for THYM, and around 70% for OV, ESCA, and DLBC; for these four cancers, the best method can achieve AUROC at least 80%, indicating adequate performance given an appropriate MIL method is selected. Fig. 4c shows boxplots of mean AUROC for different cancer types by method, arranged in an increasing order of the median of each boxplot. Overall, the three BS methods, EMD-SVM, MInD, and NSK-SVM, are top performers, followed by the ES method MILES. It is interesting to observe that these four methods also form the top tier in our simulation under model II. By contrast, the other two BS methods C*k*NN and miGraph do not perform well and fall into the bottom group along with the ES method MI-Net that performs much worse than the others. Here, the poor performance of MI-Net is perhaps due to the fact that it is based on deep learning, which typically requires balanced data with a very large sample size to perform well.

We further plot how individual methods perform by cancer type for imbalanced TCGA data in Supplementary Fig. 5, where we only include the five cancers with the maximum AUROC greater than 75% and exclude the other five for which none of the methods works adequately. EMD-SVM works very well and achieves the best or close to the best performance for all the five cancers. MInD and NSK-SVM both achieve the best or close to the best performance in three out of the five cancers and their performance is always above average. MI-Net performs the worst for all the five cancers and C*k*NN is often the second worst while miGraph is above average for three cancers but is dragged down by poor performance in the other two cancers. The IS and ES methods (except for MI-Net) somewhat stand in the middle between the two groups of BS methods, with MILES having better performance than the other mediocre methods.

For balanced TCGA data, Fig. 4b shows that again, the methods perform better on some cancer types than the others, and their median AUROC values for the top five follow the same order STAD>THYM>OV>ESCA>DLBC as in the imbalanced case. Fig. 4d



(a) 10% positive bags

(b) 50% positive bags

(c) 10% positive bags

(d) 50% positive bags

Fig. 4. TCGA data: panels (a) and (b) show boxplots of mean AUROC (%) by cancer type for different MIL methods using data with 10% and 50% positive bags, respectively; panels (c) and (d) show boxplots of mean AUROC (%) by MIL method for different cancer types using data with 10% and 50% positive bags, respectively. Categorization of MIL methods are distinguished by color (green: IS methods; blue: BS methods; magenta: ES methods).

shows that the AUROC varies in a narrower range, indicating the differences between the methods become less when compared to the imbalanced case. Also, some MIL methods are more sensitive to the balancing of classes. When p_+ increases from 10% to 50%, MILBoost and EMDD move down to the bottom group from the middle and MInD moves down to the middle from the top. On the other hand, the performance of MI-Net is improved as the sample size becomes larger (due to more positives) and the data becomes balanced. Nevertheless, EMD-SVM, NSK-SVM and MILES are still top performers. We further plot how individual methods perform by cancer type for the balanced case in Supplementary Fig. 6, where we only include the top five cancers with the maximum AUROC greater than 75%. In all the five cancers, EMD-SVM has the best performance, often followed by NSK-SVM and then MILES, while MILBoost has the worst performance. We also find that MInD works quite well in these cancer types, hence the its decreased performance as shown in Fig. 4d is due to its poor performance on the other five cancers.

Interestingly, we observe that the MIL methods seem to perform worse on cancer types regarded as immunogenic [66], namely the ones that have high levels of T cell infiltration. These include KIRC, LUSC, LUAD and SKCM. The biological mechanism of this observation is worth further experimental studies. But one possible explanation for this phenomenon is that the presence of tumors in patients of such cancer types have generated a much stronger overall activation of all T cells in the body, compared with non-immunogenic cancer types. This may have caused infiltration of both abundant tumor-specific and non-specific T cells in the tumor, which creates additional difficulty for MIL to distinguish tumor versus normal samples. Indeed, such bystander effects have been described before [67,68].

Among the five cancer types with relatively good performance given appropriate MIL methods are chosen, ESCA, OV, and STAD are among the ones with the lowest five-year survival rates; DLBC and THYM are among the most aggressive cancer types and lack physical symptoms [69,70]. Effective detection methods for asymptomatic cancer screening contribute substantially to reduce the mortality of such types of cancers. Screening using TCR sequences can be easily conducted under MIL. Such a procedure may also shed lights on more targeted experimental cancer screening methods for aggressive cancer types including but not limiting to the ones mentioned above.

6. Discussion

We explore a novel and important biomedical application of MIL and discuss its unique problem characteristics. In particular, we include a thorough discussion about two data-generation mechanisms, WR and PPI, the latter of which has not been investigated in the literature of MI classification. In our application of cancer screening using TCRs, both WR and PPI model frameworks are biologically plausible. We then provide a systematic review of 16 MIL methods that are applicable and can be readily implemented in our application. We conduct extensive simulation under the two frameworks, to benchmark these methods and to examine impacts of various key factors on their performance. We further apply the methods to TCGA sequencing data of ten cancer types.

Based on our simulation, we find that under either framework, for most MIL methods, the two most influential factors are the number of features and WR/PPI. Also, the methods appear to work better under the WR framework. This is not surprising – as mentioned before, the PPI framework is relatively new and none of the methods was originally designed for such MI data. In particular, the IS methods work poorly under the PPI framework because their bag-level predictions often rely on the standard MI assumption, which is incompatible with the PPI framework.

As for the relative performance of the different methods evaluated for bag classification, we summarize our numerical results from simulation and data examples in Table 4, to provide general guidelines in our application for the selection of an appropriate method. No matter whether data are synthesized or real, the top performers are mainly from the BS category: EMD-SVM, MInD, and NSK-SVM are the best three for our real data analysis, and they are also in the top tier under both WR and PPI simulation models; miGraph does well under the WR model; however, it falls in the bottom group for real data and usually does not perform well for simulated data from the PPI model. On the other hand, C*k*NN, as a BS method, is the worst for real data and in the bottom group under the PPI model, meanwhile it does not rank in the top group under the WR model. The two IS methods, MILBoost and SI-*k*NN, work poorly for all data; further, all the six IS methods are non-competitive as they are never in the top groups, regardless of the data or model types. Yet another interesting observation is that MILES works reasonably well under the PPI model but does not make it to the top under the WR model. This agrees with the fact that among all, MILES is more compatible with the PPI mechanism. Note that MILES also works quite well for real data. Collectively, our findings suggest that results from real data in this new application conforms more smoothly to results from the PPI framework. Overall, for bag classification in our application, we recommend EMD-SVM and NSK-SVM. Note that, in terms of computation time, EMD-SVM is much faster than NSK-SVM. We suggest to avoid MI-Net, C*k*NN, miGraph, CCE and perhaps all the IS methods as well.

For instance classification (if relevant), based on simulation using the WR model, we suggest to use SI-*k*NN (an IS method). In real data analysis, it is extremely difficult to obtain gold-standard knowledge regarding whether a TCR is tumor-specific or not. Such knowledge is not available in our study, hence the performance of the methods for instance classification cannot be evaluated using real data.

In this study, we use tumor resections and adjacent normal tissues to serve as a proof of concept for distinguishing cancer patients from healthy individuals via TCR sequencing of blood samples. Admittedly, TCRs of tumor resections are not exactly the same as TCRs from peripheral blood, which is one caveat of

Table 4
The best and worst MIL methods for bag classification based on our numerical evaluation using simulation and real data examples. Categorization of MIL methods are distinguished by color (green and italic: IS methods; blue and bold: BS methods; magenta: ES methods).

Evaluation		Best	Worst
Simulation	Model I (WR)	**NSK-SVM**, **miGraph**, **EMD-SVM**, **MInD**	*MILBoost*, *EMDD*, *SI-kNN*
	Model II (PPI)	**NSK-SVM**, **EMD-SVM**, **MInD**, MILES	*MILBoost*, *EMDD*, *SI-kNN*, *MI-SVM*, *mi-SVM*, *SI-SVM*, **C*k*NN**, BoW, CCE
TCGA	Imbalanced case	**EMD-SVM**, **MInD**, **NSK-SVM**, MILES	MI-Net, **C*k*NN**, **miGraph**, CCE
	Balanced case	**EMD-SVM**, **NSK-SVM**, MILES	*MILBoost*, **C*k*NN**, *EMDD*, MI-Net

the current study. However, we are not aware of any existing peripheral blood TCR-sequencing datasets with an adequately large number of patients comparable to TCGA, which is needed for proper training and testing of the MIL methods.

Feature selection is not considered in this study for two reasons. First, most MIL methods do not have built-in feature selection capacity. Second, for BS and ES methods, a new feature space is created from the original training instances, and the procedure for creating the new space can be rather diverse. It is thereby difficult to conduct head-to–head comparison of MIL methods with different feature embedding strategies. However, when developing new MIL methods, feature selection is definitely an important issue to consider.

With the flux of high-volume and high-dimensional data in the information era, we envision an increasing need for the development of MIL methods on burgeoning applications, especially when the PPI model is a fit and existing methods are not yet sufficient, as demonstrated in our application. One important direction is to develop model-based methods, dedicated to addressing MI problems where primary instances are required to be identified. One can extend the Bayesian hierarchical model of dichotomous response [71] to the MI setting, as a hierarchical Bayesian approach is well suited for modeling complicated data structures. Additionally, unlike most optimization-based methods, the Bayesian approach enjoys great advantage in providing statistical inference and interpretability.

Finally, we recognize the need to develop user-friendly and portable R and/or Python packages to implement existing MIL methods so that researchers in the statistical, biostatistical and bioinformatical fields can deploy the open-source software locally to explore their own datasets in other applications.

CRediT authorship contribution statement

Danyi Xiong: Methodology, Data curation, Software, Formal analysis, Writing - original draft, Writing - review & editing. **Ze Zhang:** Software, Data curation, Writing - review & editing. **Tao Wang:** Conceptualization, Data curation, Writing - review & editing, Project administration, Funding acquisition. **Xinlei Wang:** Conceptualization, Methodology, Writing - original draft, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by NIH grants R01CA258584 (PIs: T. Wang and X. Wang), R15GM131390 (PI: X. Wang), and P30CA142543 (PI: T. Wang), and Cancer Prevention and Research Institute of Texas (CPRIT) grant RP190208 (PI: T. Wang).

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.csbj.2021.05.038.

References

[1] Dietterich TG, Lathrop RH, Lozano-Pérez T. Solving the multiple instance problem with axis-parallel rectangles. Artif Intell 1997;89:31–71.
[2] Maron O, Lozano-Pérez T. A framework for multiple-instance learning. In: Advances in neural information processing systems 1998;570–576..
[3] Andrews S, Tsochantaridis I, Hofmann T. Support vector machines for multiple-instance learning. In: Advances in neural information processing systems 2003;577–584..
[4] Zhang Z-L, Zhang M-L. Multi-instance multi-label learning with application to scene classification. In: Advances in neural information processing systems 2007;1609–1616..
[5] Zha Z-J, Hua X-S, Mei T, Wang J, Qi G-J, Wang Z. Joint multi-label multi-instance learning for image classification. In: 2008 ieee conference on computer vision and pattern recognition, IEEE. 2008;1–8..
[6] Zhou Z-H, Zhang M-L, Huang S-J, Li Y-F. Multi-instance multi-label learning. Artif Intell 2012;176:2291–320.
[7] Briggs F, Lakshminarayanan B, Neal L, Fern XZ, Raich R, Hadley SJ, Hadley AS, Betts MG. Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach. J Acoust Soc Am 2012;131:4640–50.
[8] Pathak D, Shelhamer E, Long J, Darrell T. Fully convolutional multi-class multiple instance learning. arXiv preprint arXiv:1412.7144; 2014..
[9] Amar RA, Dooly DR, Goldman SA, Zhang Q. Multiple-instance learning of real-valued data. In: ICML, Citeseer. 2001, p. 3–10..
[10] Wang Z, Radosavljevic V, Han B, Obradovic Z, Vucetic S. Aerosol optical depth prediction from satellite observations by multiple instance regression. In: Proceedings of the 2008 SIAM international conference on data mining, SIAM. p. 165–76.
[11] Teramoto R, Kashima H. Prediction of protein–ligand binding affinities using multiple instance learning. J Mol Graph Model 2010;29:492–7.
[12] Park S, Wang X, Lim J, Xiao G, Lu T, Wang T. Bayesian multiple instance regression for modeling immunogenic neoantigens. Stat Meth Med Res 2020;0962280220914321..
[13] Bergeron C, Zaretzki J, Breneman C, Bennett KP. Multiple instance ranking. In: Proceedings of the 25th international conference on machine learning. p. 48–55.
[14] Hu Y, Li M, Yu N. Multiple-instance ranking: Learning to rank images for image retrieval. In: 2008 IEEE conference on computer vision and pattern recognition, IEEE. p. 1–8.
[15] Zhang M-L, Zhou Z-H. Multi-instance clustering with applications to multi-instance prediction. Appl Intell 2009;31:47–68.
[16] Zhang D, Wang F, Si L, Li T. Maximum margin multiple instance clustering with applications to image and text clustering. IEEE Trans Neural Netw 2011;22:739–51.
[17] Quellec G, Cazuguel G, Cochener B, Lamard M. Multiple-instance learning for medical image and video analysis. IEEE Rev Biomed Eng 2017;10:213–34.
[18] Foulds J, Frank E. A review of multi-instance learning assumptions. Knowl Eng Rev 2010;25:1–25.
[19] Amores J. Multiple instance classification: Review, taxonomy and comparative study. Artif Intell 2013;201:81–105.
[20] Cheplygina V, Tax DM, Loog M. Multiple instance learning with bag dissimilarities. Pattern Recogn 2015;48:264–75.
[21] Astorino A, Fuduli A, Gaudioso M. A lagrangian relaxation approach for binary multiple instance classification. IEEE Trans Neural Netw Learn Syst 2019;30:2662–71.
[22] Carbonneau M-A, Cheplygina V, Granger E, Gagnon G. Multiple instance learning: A survey of problem characteristics and applications. Pattern Recogn 2018;77:329–53.
[23] Gaudioso M, Giallombardo G, Miglionico G, Vocaturo E. Classification in the multiple instance learning framework via spherical separation. Soft Comput 2020;24:5071–7.
[24] Vocaturo E, Zumpano E. Multiple instance learning approaches for melanoma and dysplastic nevi images classification. In: 2020 19th IEEE international conference on machine learning and applications (ICMLA). IEEE. 2020, p. 1396–401..
[25] Vocaturo E, Zumpano E, Giallombardo G, Miglionico G. Dc-smil: A multiple instance learning solution via spherical separation for automated detection of displastyc nevi. In: Proceedings of the 24th symposium on international database engineering & applications. p. 1–9.
[26] Organization WH, et al. Wha58. 22 cancer prevention and control, World Health Assembly [Internet] 2005;1–5..
[27] Clarke-Pearson DL. Screening for ovarian cancer. N Engl J Med 2009;361:170–7.
[28] Byers LA, Rudin CM. Small cell lung cancer: where do we go from here? Cancer 2015;121:664–72.
[29] Singhi AD, Koay EJ, Chari ST, Maitra A. Early detection of pancreatic cancer: opportunities and challenges. Gastroenterology 2019;156:2024–40.
[30] Pardoll D. Does the immune system see tumors as foreign or self?. Ann Rev Immunol 2003;21:807–39.
[31] Raulet DH, Guerra N. Oncogenic stress sensed by the immune system: role of natural killer cell receptors. Nat Rev Immunol 2009;9:568–80.
[32] Grivennikov SI, Greten FR, Karin M. Immunity, inflammation, and cancer. Cell 2010;140:883–99.
[33] Jin Y-B, Luo W, Zhang G-Y, Lin K-R, Cui J-H, Chen X-P, Pan Y-M, Mao X-F, Tang J, Wang Y-J. Tcr repertoire profiling of tumors, adjacent normal tissues, and peripheral blood predicts survival in nasopharyngeal carcinoma. Cancer Immunol Immunother 2018;67:1719–30.
[34] Beshnova D, Ye J, Onabolu O, Moon B, Zheng W, Fu Y-X, Brugarolas J, Lea J, Li B. De novo prediction of cancer-associated t cell receptors for noninvasive cancer detection. Sci Trans Med 2020;12.

[35] Ostmeyer J, Christley S, Toby IT, Cowell LG. Biophysicochemical motifs in t-cell receptor sequences distinguish repertoires from tumor-infiltrating lymphocyte and adjacent healthy tissue. Cancer Res 2019;79:1671–80.
[36] Li B, Li T, Wang B, Dou R, Zhang J, Liu JS, Liu XS. Ultrasensitive detection of tcr hypervariable-region sequences in solid-tissue rna–seq data. Nat Genet 2017;49:482–3.
[37] Bolotin DA, Shugay M, Mamedov IZ, Putintseva EV, Turchaninova MA, Zvyagin IV, Britanova OV, Chudakov DM. Mitcr: software for t-cell receptor sequencing data analysis. Nat Meth 2013;10:813.
[38] Zhang Z, Xiong D, Wang X, Liu H, Wang T. Mapping the functional landscape of t cell receptor repertoires by single-t cell transcriptomics. Nat Meth 2021;18:92–9.
[39] Atchley WR, Zhao J, Fernandes AD, Drüke T. Solving the protein sequence metric problem. Proc Nat Acad Sci 2005;102:6395–400.
[40] Kershaw MH, Westwood JA, Darcy PK. Gene-engineered t cells for cancer therapy. Nat Rev Cancer 2013;13:525–41.
[41] Frank E, Xu X. Applying propositional learning algorithms to multi-instance data. 2003..
[42] Zhang Q, Goldman SA. Em-dd: An improved multiple-instance learning technique. In: Advances in neural information processing systems. 2002. p. 1073–80..
[43] Ray S, Craven M. Supervised versus multiple instance learning: An empirical comparison, in. In: Proceedings of the 22nd international conference on machine learning, ACM. p. 697–704.
[44] Babenko B, Dollár P, Tu Z, Belongie S. Simultaneous learning and alignment: Multi-instance and multi-pose learning, in. Workshop on Faces in'Real-Life'Images: Detection, Alignment, and Recognition 2008.
[45] Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Stat 2001:1189–232.
[46] Wang X, Yan Y, Tang P, Bai X, Liu W. Revisiting multiple instance neural networks. Pattern Recogn 2018;74:15–24.
[47] Wang J, Zucker J-D. Solving multiple-instance problem: A lazy learning approach, 2000..
[48] Gärtner T, Flach PA, Kowalczyk A, Smola AJ. Multi-instance kernels. In: ICML, vol. 2; 2002. p. 7..
[49] Zhang J, Marszałek M, Lazebnik S, Schmid C. Local features and kernels for classification of texture and object categories: A comprehensive study. Int J Comput Vision 2007;73:213–38.
[50] Rubner Y, Tomasi C, Guibas LJ. The earth mover's distance as a metric for image retrieval. Int J Comput Vision 2000;40:99–121.
[51] Zhou Z-H, Sun Y-Y, Li Y-F. Multi-instance learning by treating instances as non-iid samples. In: Proceedings of the 26th annual international conference on machine learning, ACM. p. 1249–56.
[52] Zhou Z-H, Xu J-M. On the relation between multi-instance learning and semi-supervised learning, in. In: Proceedings of the 24th international conference on machine learning. p. 1167–74.
[53] Zhou Z-H, Zhang M-L. Solving multi-instance problems with classifier ensemble based on constructive clustering. Knowl Inf Syst 2007;11:155–70.
[54] Chen Y, Bi J, Wang JZ. Miles: Multiple-instance learning via embedded instance selection. IEEE Trans Pattern Anal Mach Intell 2006;28:1931–47.
[55] Zhu J, Rosset S, Tibshirani R, Hastie TJ. 1-norm support vector machines. In: Advances in neural information processing systems, 2004. p. 49–56..
[56] Lee C-Y, Xie S, Gallagher P, Zhang Z, Tu Z. Deeply-supervised nets. In: Artificial intelligence and statistics, PMLR, 2015. p. 562–70..
[57] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. p. 770–8.
[58] Ilse M, Tomczak J, Welling M. Attention-based deep multiple instance learning. In: International conference on machine learning, PMLR, 2018. p. 2127–36..
[59] Raffel C, Ellis DP. Feed-forward networks with attention can solve some long-term memory problems, arXiv preprint arXiv:1512.08756; 2015..
[60] Rudolph MG, Stanfield RL, Wilson IA. How tcrs bind mhcs, peptides, and coreceptors. Annu Rev Immunol 2006;24:419–66.
[61] Okamoto S, Mineno J, Ikeda H, Fujiwara H, Yasukawa M, Shiku H, Kato I. Improved expression and reactivity of transduced tumor-specific tcrs in human lymphocytes by specific silencing of endogenous tcr. Cancer Res 2009;69:9003–11.
[62] Network CGAR et al. Comprehensive molecular characterization of clear cell renal cell carcinoma. Nature 2013;499:43–9.
[63] Lambrechts D, Wauters E, Boeckx B, Aibar S, Nittner D, Burton O, Bassez A, Decaluwé H, Pircher A, Van den Eynde K, et al. Phenotype molding of stromal cells in the lung tumor microenvironment. Nat Med 2018;24:1277–89.
[64] Liu J, Lichtenberg T, Hoadley KA, Poisson LM, Lazar AJ, Cherniack AD, Kovatich AJ, Benz CC, Levine DA, Lee AV, et al. An integrated tcga pan-cancer clinical data resource to drive high-quality survival outcome analytics. Cell 2018;173:400–16.
[65] Cawley GC, Talbot NL. On over-fitting in model selection and subsequent selection bias in performance evaluation. J Mach Learn Res 2010;11:2079–107.
[66] Wang T, Lu R, Kapur P, Jaiswal BS, Hannan R, Zhang Z, Pedrosa I, Luke JJ, Zhang H, Goldstein LD, et al. An empirical approach leveraging tumorgrafts to dissect the tumor microenvironment in renal cell carcinoma identifies missing link to prognostic inflammatory factors. Cancer Discov 2018;8:1142–55.
[67] Whiteside SK, Snook JP, Williams MA, Weis JJ. Bystander t cells: a balancing act of friends and foes. Trends Immunol 2018;39:1021–35.
[68] Iwahori K, Kakarla S, Velasquez MP, Yu F, Yi Z, Gerken C, Song X-T, Gottschalk S. Engager t cells: a new class of antigen-specific t cells that redirect bystander t cells. Mol Ther 2015;23:171–8.
[69] Harris C, Croce B, Xie A. Thymoma. Ann Cardiothor Surgery 2015;4:576.
[70] Wang X, Xiong H, Liang D, Chen Z, Li X, Zhang K. The role of srgn in the survival and immune infiltrates of skin cutaneous melanoma (skcm) and skcm-metastasis patients. BMC Cancer 2020;20:1–8.
[71] Albert JH, Chib S. Bayesian analysis of binary and polychotomous response data. J Am Stat Assoc 1993;88:669–79.