Fall 12-18-2021

Bayesian Multiple Instance Learning with Application to Cancer Detection Using TCR Repertoire Sequencing Data

DANYI XIONG
Southern Methodist University, dxiong@smu.edu

BAYESIAN MULTIPLE INSTANCE LEARNING WITH APPLICATION TO CANCER DETECTION USING TCR REPERTOIRE SEQUENCING DATA

Approved by:

Dr. Xinlei Wang
Professor in Department of Statistical Science, SMU

Dr. Daniel F. Heitjan
Professor in Department of Statistical Science, SMU & Population and Data Sciences, UTSW

Dr. Lynne Stokes
Professor in Department of Statistical Science, SMU

Dr. Sandi L. Pruitt
Associate Professor in Department of Population and Data Sciences, UTSW

Dr. Tao Wang
Assistant Professor in Department of Population and Data Sciences, UTSW

BAYESIAN MULTIPLE INSTANCE LEARNING WITH APPLICATION TO CANCER DETECTION USING TCR REPERTOIRE SEQUENCING DATA

A Dissertation Presented to the Graduate Faculty of the

Dedman College

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Doctor of Philosophy

with a

Major in Biostatistics

by

Danyi Xiong

B.S., Mathematics and Statistics, University of Illinois at Urbana-Champaign
M.A., Statistics, University of California at Berkeley

December 18, 2021

ACKNOWLEDGMENTS

My deepest gratitude to my advisor Dr. Xinlei Wang, who is a great mentor, and I could not have been in better hands. I thank my lucky stars for her. Dr. Wang worked with me closely and guided me every step of the way. She had more faith in me than I did. Without her patient training and expert guidance, this dissertation would not have been possible. As my advisor, Dr. Wang not only trained me to be a biostatistician, but also continued to be a source of wisdom and guidance. It is with her dedication and friendly supervision, I have grown intellectually and purposefully in this journey.

I am also very grateful for the rest of my committee members. Our Biostatistics program director, Dr. Daniel Heitjan, taught me all the essentials about Biostatistics and provided me with unwavering support and encouragement in the past few years. Dr. Lynne Stokes taught me a tremendous amount about writing and got me ready to write my thesis in the third year. And my very generous professors at UTSW, Dr. Sandi Pruitt and Dr. Tao Wang, offered me valuable opportunities for joining their work and collaborating with many talented researchers. I also want to thank all faculty members in the Department of Statistical Science at SMU for their teaching and support during my Ph.D. study.

Enormous thanks to Ze Zhang and Dr. Seongoh Park, who made this dissertation a better piece of work through their insightful suggestions and tremendous support along the way.

Finally, my heartfelt thanks to my friends and family, who have shaped me in innumerable ways. My close friends and cheerleaders, Can Xu, Jingjing Qu, Yu Hu, Shuang Jiang, Xiaohan Xu, Heng Cui, Zi Yin, Fan Wu, Jia Yu, Yuecheng Zhou, Wentian Huang, and Wenmian Hua, have kept me sane and grounded throughout this journey. I appre-

ciate their company and support. I also want to thank my friends at UTSW, especially Yingfei Chen, whose help cannot be overestimated. Thank you to my parents, Zhiyong Xiong and Youjun Zou, who let me find my way and support me with unparalleled love. And last but not least, thank you to my nine-year-old bunny, Booboo, who has been with me since my undergrad and has brought a lot of joy to my life.

Xiong, Danyi
B.S., Mathematics and Statistics, University of Illinois at Urbana-Champaign
M.A., Statistics, University of California at Berkeley

Bayesian Multiple Instance Learning with Application to Cancer Detection Using TCR Repertoire Sequencing Data

Advisor: Dr. Xinlei Wang
Doctor of Philosophy degree conferred December 18, 2021
Dissertation completed November 30, 2021

As a branch of machine learning, multiple instance learning (MIL) learns from a collection of labeled bags, each containing a set of instances. Each instance is described by a feature vector. 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.

The first part of this dissertation focuses on a comparative study of MIL methods for a novel biomedical application. To date, the majority of the off-the-shelf MIL methods are developed in the computer science domain and so algorithm-driven. We review and apply a large collection of existing methods to investigate the applicability of MIL to 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. Based on our numerical results from extensive simulation and analysis of sequencing data from The Cancer Genome Atlas for ten types of cancer, we make suggestions about selection of a proper method and avoidance of any method with poor performance. We further identify a pressing need of new model-based MIL methodologies for accurate modeling of increasingly complex structures of real world data and more explainable outcomes.

The second part of this dissertation proposes a novel Bayesian MIL method for binary classification based on hierarchical probit regression (MICProB), which contributes a significant portion to the suite of statistical methodologies for MIL. MICProB is composed of two nested probit regression models, where the inner model is estimated for predicting primary instances, which are considered as the "important" ones that determine the bag label, and the outer model is for predicting bag labels based on the features of primary instances estimated by the inner model. The posterior distribution of MICProB can be conveniently approximated using a Gibbs sampler, and the prediction for new bags can be performed in a fully integrated Bayesian way. We evaluate the performance of MICProB against various benchmark methods and demonstrate its competitiveness in simulation and real data examples. In addition to its capability of identifying primary instances, as compared to existing optimization-based approaches, MICProB also enjoys great advantages in providing a transparent model structure, straightforward statistical inference of quantities related to model parameters, and favorable interpretability of covariate effects on the bag-level response.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I dedicate this dissertation to my family.

CHAPTER 1

A COMPARATIVE STUDY OF MULTIPLE INSTANCE LEARNING METHODS FOR CANCER DETECTION USING T-CELL RECEPTOR SEQUENCES

1.1. Introduction

First introduced in Dietterich et al. [19], 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. [19] developed an MIL algorithm to classify whether a molecule (bag) of different conformations (instances) is biologically active; in content-based image retrieval [41], 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 [4], 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

[11, 49, 70, 76, 80]. In addition to classification, MIL is applicable for real-valued responses as well [2, 60, 67]. A recent application of multiple instance regression studied the relationship between tumor immune response and immunogenic neoantigens using Bayesian hierarchical models [48]. 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 [8, 30, 51, 71, 73].

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 [20] 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 [3] 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 [5, 17]. A more recent study conducted by Carbonneau et al. [13] 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 [5, 13, 24, 61, 62]. 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.

In accordance with the Resolution on Cancer Prevention and Control (WHA58.22) at the 58th World Health Assembly, accurate and timely cancer detection, especially for aggressive cancer types, is extremely important for patients to receive appropriate treatments for best possible prognosis. Various experimental methods exist, which, however, are less ideal for detecting certain types of cancer [12, 18, 59]. Based on previous findings that the host immune responses to tumor cells are already activated during tumorigenesis process [26, 47, 53], 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. [34], few researchers have sought to predict tumor or normal status based on TCRs of T cells. Beshnova et al. [9] 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. [46] 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 1.2, we describe data and problem characteristics that are relevant to our MIL application, and identify key concepts. In Section 1.3, we describe a list of MIL methods for our application and comment briefly on their implementation. In Section 1.4, 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 1.5, we conduct analysis on real datasets obtained from The Cancer Genome Atlas (TCGA). We discuss our major findings and future work in Section 1.6.

1.2. Cancer Detection Using TCR Sequences

1.2.1. Data generation

Figure 1.1 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 [38] and MiTCR [10]. 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 [75], 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 [6] 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 [75]. 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.

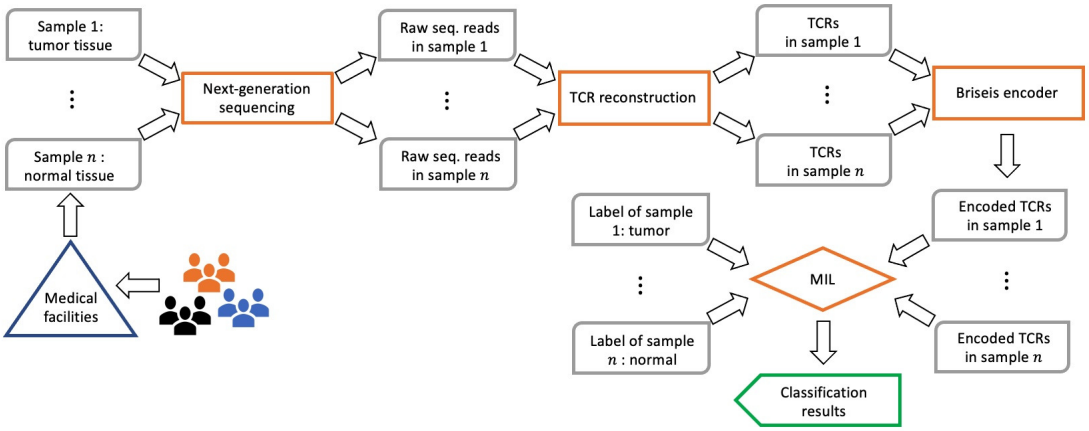


Figure 1.1: The pipeline of data processing in our MIL application of cancer detection using TCR sequences. Each encoded TCR sequence is represented by a d-dimensional numeric feature vector learned by the auto-encoder.

1.2.2. 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 [35]. 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 [13]. 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 [19]. 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. [13], 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 Ray and Page [55]. 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. [48] further 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 [21]. 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 gener-

ated 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.

1.3. 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 1.2.2. According to Amores [3], 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 (Figure 1.3).

Table 1.1 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.

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

Table 1.1: The selected MIL methods for cancer detection using TCR sequences. Those that can perform instance classification are highlighted in italics.

1.3.1. 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 predictions in a bag), and the maximum or minimum of the instance predictions. For each of the six IS methods included, we describe how instance classification is performed below.

EMDD [74]: Expectation-Maximization Diverse Density is a generalization of the Diverse Density (DD) algorithm [41], 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 [4]: 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 [54] and SI-kNN [13]: 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 kNN classifiers are optimized on the reduced (single-instance) problem.

MILBoost [7]: 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 [22].

mi-Net [66]: Named by Wang et al. [66], 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).

1.3.2. 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 kNN and SVM, to learn the bag-to-bag relationship. The following BS methods are considered for our application.

CkNN [63]: CkNN (Citation-kNN) is a variant of SI-kNN 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 [23]: 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 [72]: The proposed approach employs Earth Mover's Distance (EMD) [56] 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. [72], 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 [77]: Motivated by an observation made in Zhou and Xu [78] 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 [17]: 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.

1.3.3. 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 [3]: 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 [79]: 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 [16]: 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 [81] 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 [66]: 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 Wang et al. [66], one adding deep supervision [37] and the other considering residual connections [29], which can improve the performance sometimes.

ADeep [32]: 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 [52], where all instances are assumed independent. Unlike traditional pooling operators such as max and mean that are pre-defined 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.

1.3.4. Implementation

The MATLAB “MILSurvey" toolbox is made available online by Carbonneau et al. [13]. We use this software package to implement the MIL methods covered in Sections 1.3.1–1.3.3 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. [66] 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 1.2.2. We also refer readers to the GitHub link https://github.com/danyixiong/MIL_Comparative_Study for more detail on implementation.

1.4. 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 1.5), TCRs are converted into numeric vectors by the Briseis encoder [75]. 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.

1.4.1. 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 ap-

plication, 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 auto-immune diseases, infections, etc. [45, 57]. 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 $\mathrm{WR} = 0.5$.

Figure 1.2 shows bag classification performance of different MIL methods in terms of mean AUROC under various simulation scenarios. Overall, all BS methods except for

CkNN 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.

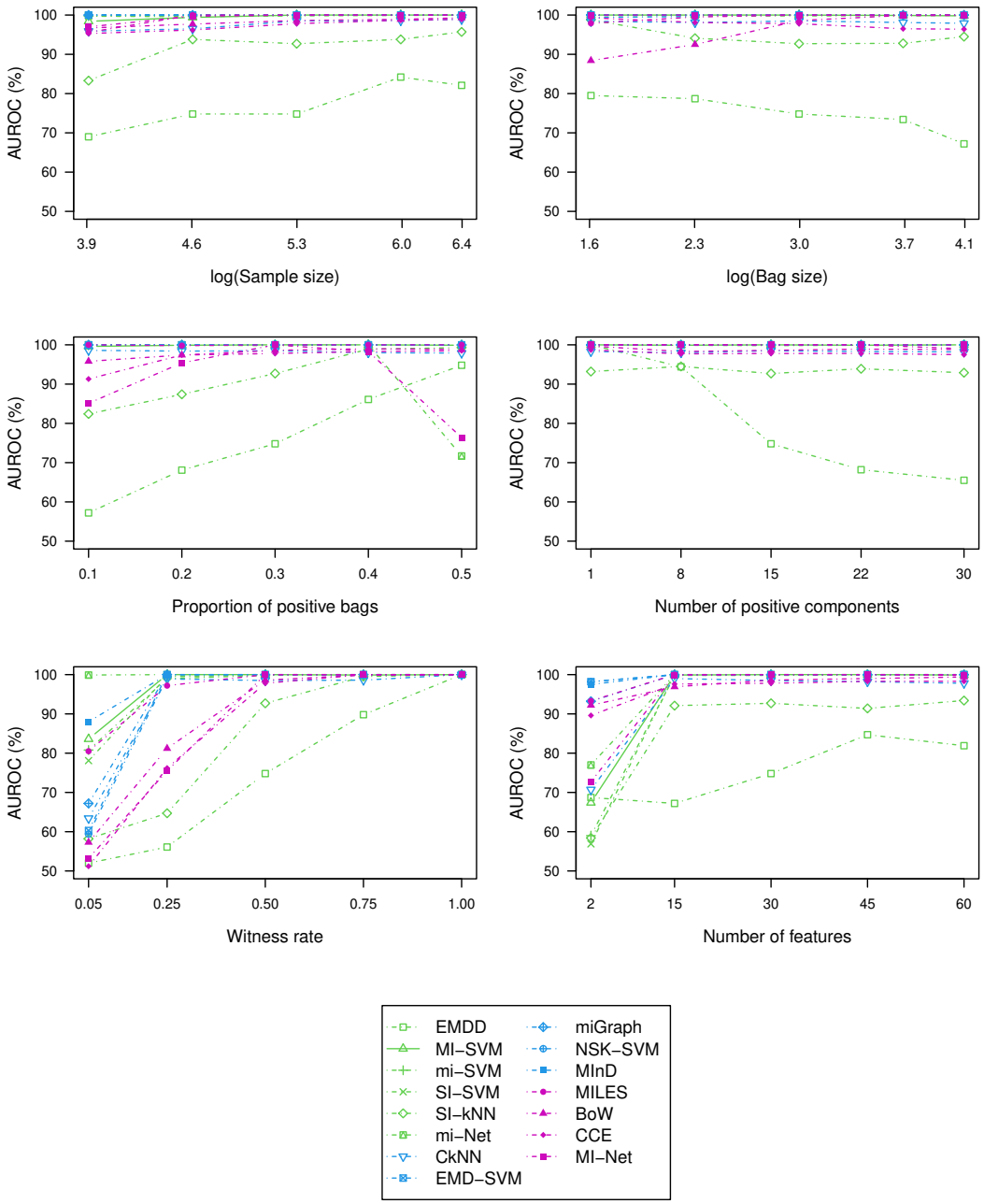


Figure 1.2: Mean AUROC (%) of bag classification using different MIL methods, evaluated on simulation scenarios each with 100 replicates generated under model I. IS/BS/ES methods are distinguished by green, blue, and magenta lines.

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-kNN. 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-kNN 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-kNN, 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 CkNN. 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 CkNN 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 Figure A.1.

As discussed in Section 1.3, MILES (an ES method) and all IS methods can be used to classify instances. Figure A.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-kNN performs the best.

1.4.2. Simulation under model II

In addition to the factors shared with model I (n, m, d, p_+), we consider varying $\overline{\text{PPI}}$ (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 $p_{ij} \equiv \Pr(\text{ffi}_{\text{ij}} = 1) = \Phi\left(b_0 + \sum_{r=1}^{d} x_{ijr} b_r\right)$, where $\Phi(\cdot)$ is the standard normal cumulative distribution function (CDF), and b_0 and b_r for $r = 1, ..., d$ are regression coefficients in the probit regression model for p_{ij}. Further, we simulate the bag label Y_i from $Ber(\pi_i)$, with $\pi_i \equiv \Pr(\text{Y}_\text{i} = 1) = \Phi\left(\beta_0 + \sum_{j=1}^{m} \delta_{ij} \sum_{r=1}^{d} x_{ijr}\beta_r\right)$, where β_0 and β_r for $r = 1, ..., d$ are regression coefficients associated with the probit model for π_i. In case where $\delta_{ij} = 0$ for all $j = 1, ..., m$, we simply generate Y_i from $Ber(\Phi(\beta_0))$. We adjust the intercepts b_0 and β_0 to vary values of $\overline{\text{PPI}}$ and proportion of positive bags p_+, respectively. We set $l = -10$, $u = 10$, $b_j = 2$, $\beta_j = -1$ $\forall j$, $\overline{\text{PPI}} = 0.05, 0.25, 0.5, 0.75, 1$ 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 $\overline{\text{PPI}} = 0.5$.

The performance of the methods under various simulation scenarios is shown in Figure 1.3. 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 CkNN have poor performance in all the scenarios, with AUROC close to 0.5, which is only slightly better than random guessing; BoW and CCE also perform poorly 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 CkNN, 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 $\overline{\mathrm{PPI}}$ increases, most methods show higher AUROC by capturing the increased amount of useful information. Greater improvement is observed when $\overline{\mathrm{PPI}}$ 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

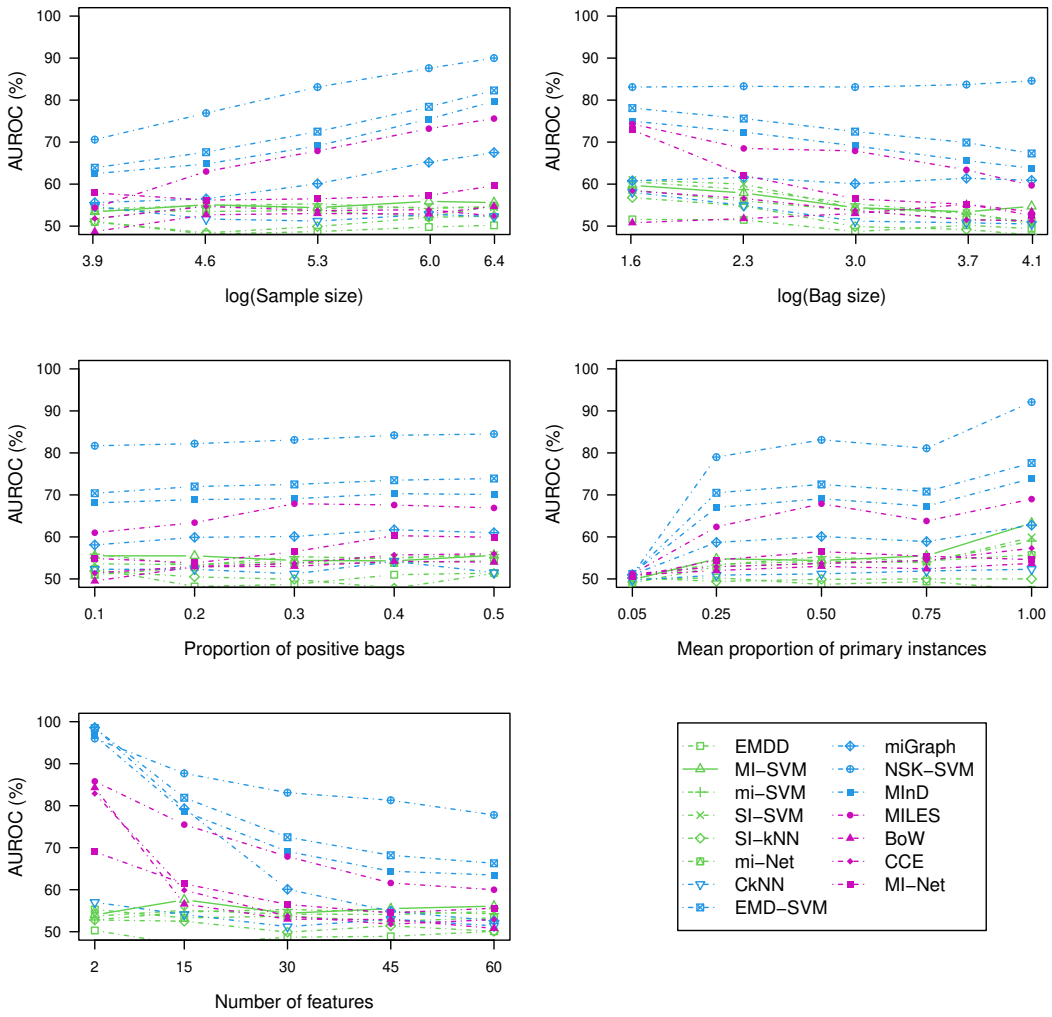


Figure 1.3: Mean AUROC (%) of bag classification using different MIL methods, evaluated on simulation scenarios each with 100 replicates generated under model II. IS/BS/ES methods are distinguished by green, blue, and magenta lines.

framework as d goes up but not under the PPI framework.

In terms of AUPRC, as showed in Figure A.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.

1.4.3. Computation time

We provide the runtime information of each method under the basic setting of each model in Table 1.2. Fourteen methods are run 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.

IS methods	MILBoost	SI-kNN	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	CkNN	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)			

Table 1.2: 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.

1.5. Real Data Examples

1.5.1. 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. Figure A.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) [36, 39, 43]. 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.

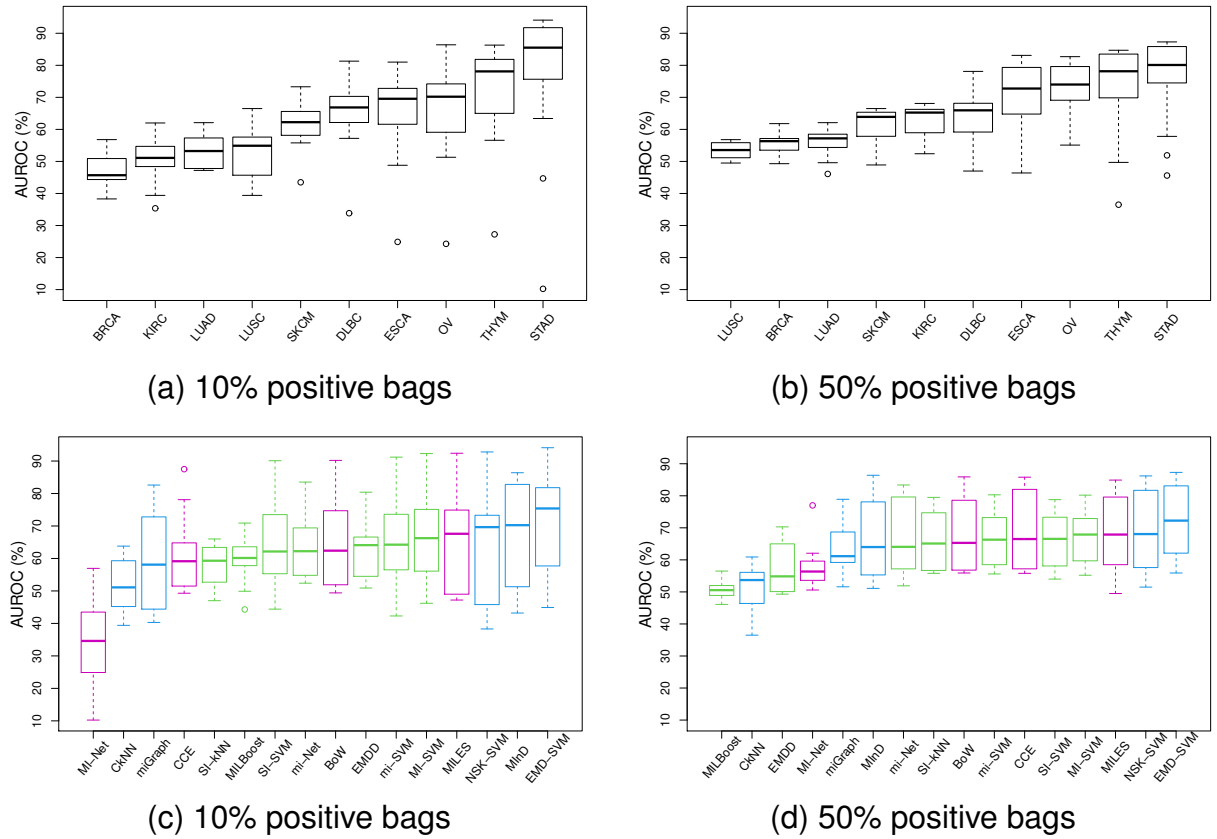
TCR sequences were reconstructed by MiTCR from the TCGA RNA-sequencing data. MiTCR is a commonly used software for reconstructingTCR sequences from next generation sequencing data [10]. MiTCR also records the number (abundance) of each unique TCR in each sample (bag). We exclude TCRs whose abundance 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 $\sim$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 (Figure A.4), the sample sizes are 90, 216, and 332, respectively; for each of the remaining cancers, the total sample size is 404. Table 1.3 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.

	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

Table 1.3: TCGA data: descriptive statistics including the sample size and bag size (i.e., the number of instances per bag) for selected cancer types.

1.5.2. 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 and validation, a nested cross-validation (CV) procedure [13, 14] 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.



(a) 10% positive bags

(b) 50% positive bags

(c) 10% positive bags

(d) 50% positive bags

Figure 1.4: 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).

Figure 1.4(a) 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. Figure

1.4(c) 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 CkNN 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 Figure A.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 CkNN 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, Figure 1.4(b) 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. Figure 1.4(d) 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 Figure A.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 Figure 1.4(d) 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 [64], 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 [33, 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 [27, 65]. 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 can-

cer types including but not limiting to the ones mentioned above.

1.6. 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 1.4, 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, CkNN, 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-kNN, 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, CkNN, miGraph, CCE and perhaps all the IS methods as well.

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*, **CkNN**, BoW, CCE
TCGA	Imbalanced case	**EMD-SVM**, **MInD**, **NSK-SVM**, MILES	MI-Net, **CkNN**, **miGraph**, CCE
	Balanced case	**EMD-SVM**, **NSK-SVM**, MILES	*MILBoost*, **CkNN**, *EMDD*, MI-Net

Table 1.4: 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).

For instance classification (if relevant), based on simulation using the WR model, we suggest to use SI-kNN (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 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 [1] 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.

CHAPTER 2

BAYESIAN MULTIPLE INSTANCE CLASSIFICATION BASED ON HIERARCHICAL PROBIT REGRESSION

2.1. Introduction

In contrast to conventional machine learning where the observed response is associated with only one feature (or covariate) vector, multiple instance learning (MIL) assumes that input data are organized as a collection of bags, each containing one or more instances and each instance described by a set of features [19]. In supervised problems including multiple instance classification (MIC) and multiple instance regression (MIR), a response variable, or often referred to as a label in literature, is observed at the bag level, but not individually at the instance level. The primary objective of MIL is to predict the bag label based on all its instances by learning the underlying relationship between bags and instances. Besides the voluminous data introduced by multiple instances per bag, the instance labels are not observed directly or even not clearly defined, bringing additional challenge to the learning process. Nevertheless, MIL gains great popularity as it provides an approach to solve many real-life tasks that naturally consist of multiple instance (MI) data. For example, in drug activity prediction, a molecule of different conformations (instances) is treated as a bag [19]. In weakly supervised object detection, an image is viewed as a bag with multiple non-overlapping regions (instances) [13].

In the past decades, development of MIL methods for a variety of MI problems has been quite active, especially in the field of computer vision. Several works have re-

viewed and/or compared existing MIL methods and applications [3, 13, 20]. In Amores [3], MIL methods for binary classification are categorized into three different paradigms, namely, instance-space (IS), bag-space (BS), and embedded-space (ES), depending on the means that a method takes to learn bag labels from instances in the bags. For IS methods, the learning process occurs at the instance level, where an instance-level classifier is trained to predict scores for instances. Bag labels are then obtained by aggregating instance prediction based on a suitable MI assumption. In this sense, IS methods focus on the characteristics of individual instances and overlook global characteristics of the entire bag. By contrast, both BS and ES methods treat each bag as a whole entity, and train a bag-level classifier 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 in a new feature space, and then make direct bag-level prediction using standard classifiers. In the first application of MIL, Dietterich et al. [19] proposed an IS method named Axis-parallel Rectangles to predict drug activity. Later, more IS methods are developed based on the standard MI assumption that a positive bag has at least one positive instance and a negative bag only has negative instances [41, 74]. To solve applications in computer vision and text categorization which have more complex data structures, BS and ES methods are developed subsequently and become quite popular [4, 16, 17, 32, 66, 72]. While MIC problems receive a lot of attention from computer scientists, statisticians are less aware of this type of problems and statistical methodologies in this field are under-developed, except for Chen et al. [15] that proposed a MI logistic regression model (MILR) with an optional Lasso penalty term and developed an R package. Since MILR associates the bag probability to the predicted instance probabilities of being positive via the standard MI assumption, it belongs to the IS category.

In Chapter 1 a recent MIL application on cancer detection using T-cell receptor (TCR) sequences, Xiong et al. [69] formally discussed two types of data generation mechanisms and evaluated the performance of 16 existing MIC methods under each mechanism. The first relies on witness rate (WR), defined as the proportion of positive instances in positive bags [13]. Under this so-called WR framework, only the number or proportion of the positive instances is responsible for labeling a bag as positive, which implies that more positive instances typically indicating a higher confidence of a positive bag. The second formulation is based on primary instances, a concept first introduced by Ray and Page [55] in MIR, yet rarely mentioned in the MIC literature. It is assumed that 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. Analogous to WR, Xiong et al. [69] defined the proportion of primary instances (PPI) as the number of primary instances divided by the total number of instances in a bag, and referred to this formulation as the PPI framework. Until this work, the PPI framework has not been investigated in the MIC literature. Based on numerical results from both simulation and analyses of sequencing data for multiple cancer types, the authors recommended EMD-SVM [72] and NSK-SVM [23] for their overall better performance over other compared methods. Possibly due to the presence of abundant by-standing TCRs (corresponding to non-primary instances) naive to any antigenic process, the authors also pointed out that results from real data were much more consistent with those from simulated data under the PPI framework, hence calling for new MIC methodology to be developed based on the PPI framework.

In fact, other applications can also be formulated under the PPI framework. For example, Wang et al. [67] predicted aerosol optical depth from satellite measurements where they treated instances as noisy versions of the primary instance. Recently, Park et al. [48] developed a Bayesian multiple instance regression model (BMIR) to study the relationship between tumor immune response and immunogenic neoantigens, assuming that each bag contains a single primary instance. Although BMIR enables interpretability of covariate effects due to its regression formulation, the assumption of a single primary

instance per bag is rather restrictive. Furthermore, after fitting the model with training data, BMIR requires an auxiliary random forest model to identify the primary instances, rendering the prediction for new bags less straightforward.

Motivated by Xiong et al. [69], we develop a Bayesian MIC method based on a two-tier probit regression model (MICProB) under the PPI framework. This novel method does not belong to any of the three paradigms (IS, BS, and ES) defined in Amores [3], and adds to the suite of methodologies to address MIC problems where primary instances need to be identified. MICProB provides a fully integrated Bayesian solution that not only performs training and prediction simultaneously, but also allows for statistical inference and offers great explainability. By contrast, most existing MIC methods are algorithm-driven based on the WR framework, and so cannot offer insights about the mechanism behind data. The two statistical methods, MILR and BMIR, are both model-based, and so are similar to MICProB in terms of explainability; however, MILR is based on the standard MI assumption under the WR framework. While BMIR is based on the PPI framework, it requires continuous outcomes and predicts the labels for new bags after the training process is done. Such a sequential approach ignores estimation uncertainty from the training step.

The remainder of this paper is organized as follows. In Section 2.2, we describe the proposed Bayesian model, computation, inference, and posterior-based prediction. In Section 2.3, we conduct simulation studies to assess the performance of our method and compare it with 15 benchmark methods, under various design configurations. In Section 2.4, we illustrate the usage of proposed method with a real application of cancer detection using TCR sequences and also demonstrate its high explainability in the application of modeling immunogenic neoantigens. We summarize our findings and discuss potential extensions of our method in Section 2.5.

2.2. Methods

Let B_i denote bag i containing m_i instances, and y_i denote the observed binary bag label (or outcome) for $i = 1, ..., n$, where n is the total number of observed bags (or sample size). Suppose there are d features (or covariates) that characterize each instance j. We use $X_i = (x_{ij})_{j=1}^{m_i}$ to denote the $m_i \times (d+1)$ feature matrix of B_i by stacking x_{ij}'s row-wisely, where $x_{ij} = (1, x_{ij1}, ..., x_{ijd})$ is a row vector of length $d+1$. In many practical situations, not all the instances are necessarily relevant, and there might be instances inside one bag that do not convey any information about its label. Furthermore, its own feature vector x_{ij} may help predict whether an instance is relevant or not. For each bag i, we refer to those relevant instances as its primary instances, collectively denoted by $\widetilde{\mathsf{B}}_i$. Let δ_{ij} be a latent indicator variable, with $\delta_{ij} = 1$ indicating that instance j is a primary instance of B_i and 0 otherwise.

2.2.1. Model and prior specification

By assuming primary instances of all bags are known, we first consider a probit regression setup to model the relationship between the feature vectors x_{ij}'s of B_i and the outcome y_i. Namely,

$$\begin{aligned} y_i &= \mathsf{sign}(Z_i), \\ Z_i &= \sum_{j=1}^{m_i} \delta_{ij} x_{ij} \beta / C_i + \epsilon_i, \end{aligned}$$

where $\epsilon_i \overset{ind}{\sim} N(0,1)$ for $i = 1, ..., n$, and $\beta = (\beta_r)_{r=0}^{d}$ is a column vector of intercept and regression coefficients describing the covariate effects on the observed outcome variable. Further, C_i is a normalizing factor that may account for the different number of primary instances in different bags. For example, $C_i = 1$ corresponds to the sum contribution, while $C_i = |\widetilde{\mathsf{B}}_i|$ corresponds to the average contribution of $\widetilde{\mathsf{B}}_i$. In case where $\widetilde{\mathsf{B}}_i = \emptyset$, we

let $Z_i = \beta_0 + \epsilon_i$. Without loss of generality, we focus on the sum contribution model. Thus, Pr$(y_i = 1|X_i, \beta, \delta_{i1}, ..., \delta_{im_i}) = \Phi\left(\sum_{j=1}^{m_i} \delta_{ij} x_{ij} \beta\right)$ for $C_i = 1$, where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.

Next, we model the latent primary indicator of instance j in bag i (i.e., δ_{ij}) through another probit regression model:

$$\begin{aligned} \delta_{ij} =& \text{sign}(U_{ij}), \\ U_{ij} =& x_{ij} b + e_{ij}, \end{aligned}$$

where $e_{ij} \overset{ind}{\sim} N(0,1)$ for $j = 1, ..., m_i$ and $i = 1, ..., n$. Here, $b = (b_r)_{r=0}^{d}$ is the column vector of intercept and coefficients describing the covariate effects on the instance status (primary vs. non-primary). Similarly, we have Pr$(\delta_{ij} = 1|x_{ij}, b) = \Phi(x_{ij} b)$. In each probit model, Z_i's or U_{ij}'s are latent variables which are introduced to make the subsequent Markov Chain Monte Carlo (MCMC) algorithm for posterior sampling become more convenient, due to the data augmentation technique proposed for binary response data in Albert and Chib [1]. The variances of ϵ_i and e_{ij} are both fixed at 1 for model identifiability.

For the above two-tier probit regression model, we employ (conditional) conjugate priors, which are commonly used in the Bayesian regression literature to achieve convenient posterior sampling. The prior for the regression coefficients β is specified as $\beta|\mu_\beta, \Sigma_\beta \sim$ MVN$(\mu_\beta, \Sigma_\beta)$. It is routine to set $\mu_\beta = (0, 0, ..., 0)$. Similarly, we assign $b|\mu_b, \Sigma_b \sim$ MVN(μ_b, Σ_b) and set $\mu_b = (0, 0, ..., 0)$. For Σ_β and Σ_b, we adopt the hyper-parameter values suggested by Polson et al. [50] and employ a diagonal matrix with $(16, 4, ..., 4)$ on the diagonal entries. Figure 2.1(a) describes the Bayesian hierarchical model structure.

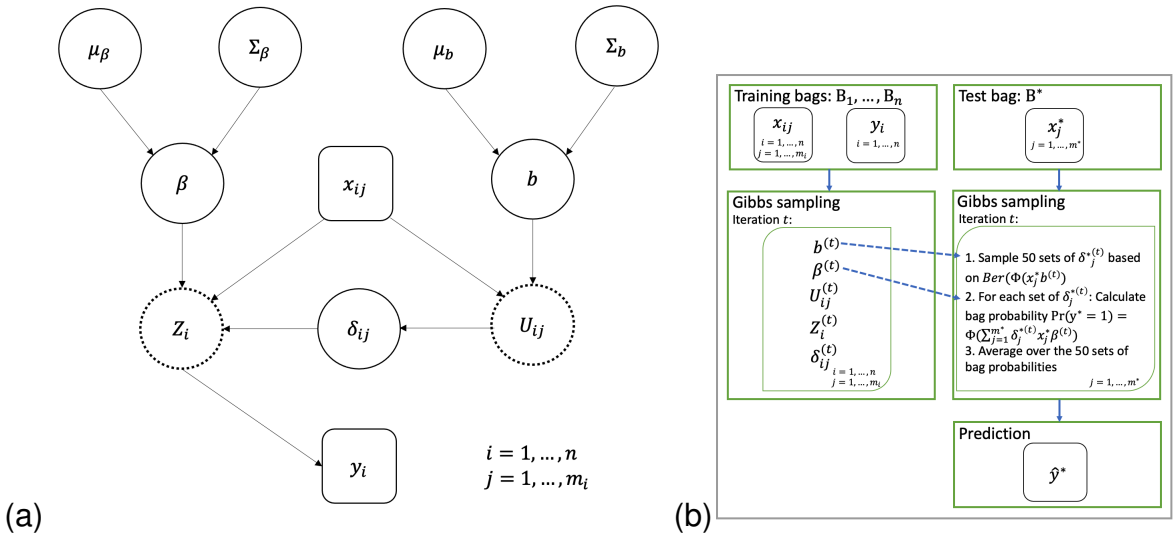


Figure 2.1: (a) Bayesian hierarchical model structure of MICProB. Observed data, including instances x_{ij}'s and bag labels y_i's, are showed in square boxes. Latent variables, including Z_i's for response variables and U_{ij}'s for indicators of primary instances, are showed in dashed circles. Hyper-parameters include μ_β, Σ_β, μ_b, and Σ_b. (b) Workflow of MICProB. Left panels explain the model fitting process on training bags and right panels describe prediction steps for a new bag.

2.2.2. Posterior computation

Let $y = (y_i)_{i=1}^n$ be a column vector of length n and $X = (X_i)_{i=1}^n$ be the collection of covariate matrices from all bags used for model fitting (i.e., the training cohort). Let Δ and U be a $(\sum_{i=1}^n m_i) \times 1$ column vector of binary indicators δ_{ij}'s and their corresponding latent variables U_{ij}'s, for $i = 1, ..., n$ and $j = 1, ..., m_i$, respectively. Similarly, let $Z = (Z_i)_{i=1}^n$ denote a $n \times 1$ column vector of latent variables Z_i's associated with the bag labels y_i's, for $i = 1, ..., n$. Let $\Theta = (\beta, b, \Delta, Z, U)$ denote the collection of all model parameters and latent variables involved. With hyper-parameters μ_β, Σ_β, μ_b and Σ_b specified, the full probability

model is given by

$$
\begin{aligned}
p(y, \Theta|X) &= p(y|Z) \times p(Z|X, \Delta, \beta) \times p(\Delta|U) \times p(U|X, b) \\
&\quad \times p(\beta|\mu_\beta, \Sigma_\beta) \times p(b|\mu_b, \Sigma_b) \\
&= \prod_{i=1}^{n} \left\{ p(y_i|Z_i) \cdot p(Z_i|x_{\delta_i}, \beta) \cdot \left[\prod_{j=1}^{m_i} p(\delta_{ij}|U_{ij}) \cdot p(U_{ij}|x_{ij}, b) \right] \right\} \\
&\quad \times p(\beta|\mu_\beta, \Sigma_\beta) \times p(b|\mu_b, \Sigma_b).
\end{aligned}
$$

We use MCMC to draw random samples from the joint posterior distribution $p(\Theta|X, y) \propto p(y, \Theta|X)$. One advantage of the proposed modeling is that the conditional posterior distribution of each parameter (or latent variable) given all others, becomes tractable as a known family of distributions, as detailed below.

- $\beta|\cdots \sim$ MVN(m_β, V_β), where

$$
\begin{aligned}
m_\beta &= (\Sigma_\beta^{-1} + X_\delta^T X_\delta)^{-1}(\Sigma_\beta^{-1}\mu_\beta + X_\delta^T Z), \\
V_\beta &= (\Sigma_\beta^{-1} + X_\delta^T X_\delta)^{-1}.
\end{aligned}
$$

  Here,

$$
X_\delta = \begin{pmatrix} x_{\delta_1} \\ x_{\delta_2} \\ \vdots \\ x_{\delta_n} \end{pmatrix} = \begin{pmatrix} 1 & \sum_{j=1}^{m_1} \delta_{1j} x_{1j1} & \cdots & \sum_{j=1}^{m_1} \delta_{1j} x_{1jd} \\ 1 & \sum_{j=1}^{m_2} \delta_{2j} x_{2j1} & \cdots & \sum_{j=1}^{m_2} \delta_{2j} x_{2jd} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \sum_{j=1}^{m_n} \delta_{nj} x_{nj1} & \cdots & \sum_{j=1}^{m_n} \delta_{nj} x_{njd} \end{pmatrix}
$$

  is a $n \times (d+1)$ covariate matrix, where the i-th row is formed by the primary instances of bag i, for $i = 1, ..., n$.

- $b|\cdots \sim$ MVN(m_b, V_b), where

$$m_b = (\Sigma_b^{-1} + X^T X)^{-1}(\Sigma_b^{-1}\mu_b + X^T U),$$
$$V_b = (\Sigma_b^{-1} + X^T X)^{-1}.$$

Let $I(\cdot)$ be an indicator function that equals 1 if the condition inside the parentheses is satisfied (0 otherwise). Latent variables Z_i's and U_{ij}'s can be sampled from truncated normal distributions, respectively:

- $Z_i|\cdots \sim \begin{cases} \mathsf{N}(g(x_{\delta_i}, \beta), 1) \cdot I(Z_i > 0) & \text{if } y_i = 1 \\ \mathsf{N}(g(x_{\delta_i}, \beta), 1) \cdot I(Z_i \leq 0) & \text{if } y_i = 0 \end{cases}$, where x_{δ_i} corresponds to the i-th row of X_δ and $g(x_{\delta_i}, \beta) = \beta_0 + \sum_{j=1}^{m_i} \delta_{ij} \sum_{r=1}^{d} x_{ijr}\beta_r$, for $i = 1, ..., n$.
- $U_{ij}|\cdots \sim \begin{cases} \mathsf{N}(h(x_{ij}, b), 1) \cdot I(U_{ij} > 0) & \text{if } \delta_{ij} = 1 \\ \mathsf{N}(h(x_{ij}, b), 1) \cdot I(U_{ij} \leq 0) & \text{if } \delta_{ij} = 0 \end{cases}$, where $h(x_{ij}, b) = b_0 + \sum_{r=1}^{d} x_{ijr} b_r$, for $i = 1, ..., n$ and $j = 1, ..., m_i$.

Lastly, the binary indicator for primary instance follows a Bernoulli distribution conditioning on other parameters:

- $\delta_{ij}|\cdots \sim$ Ber $\left(\frac{A\Phi(h(x_{ij},b))}{A\Phi(h(x_{ij},b))+B[1-\Phi(h(x_{ij},b))]}\right)$, where

$$A = \exp\left\{-\frac{1}{2}\left(Z_i - \beta_0 - \sum_{j' \neq j}^{m_i} \delta_{ij'} \sum_{r=1}^{d} x_{ij'r}\beta_r - \sum_{r=1}^{d} x_{ijr}\beta_r\right)^2\right\},$$
$$B = \exp\left\{-\frac{1}{2}\left(Z_i - \beta_0 - \sum_{j' \neq j}^{m_i} \delta_{ij'} \sum_{r=1}^{d} x_{ijr'}\beta_r\right)^2\right\},$$

and $h(x_{ij}, b) = b_0 + \sum_{r=1}^{d} x_{ijr} b_r$, for $i = 1, ..., n$ and $j = 1, ..., m_i$.

The above analytical forms allow us to utilize a Gibbs sampler to easily draw samples from $p(\Theta|X, y)$ after proper convergence of the MCMC algorithm.

2.2.3. Posterior inference

Suppose we run the Gibbs sampler for T iterations after the burn-in period. Point estimates of quantities of interest are made based on posterior means (or medians/modes). For example, the covariate effects on the response variable are estimated by $\hat{\beta} = \frac{1}{T}\sum_{t=1}^{T}\beta^{(t)}$, where $\beta^{(t)}$ is the draw of β at iteration t. Estimation of uncertainty is quantified using Bayesian credible intervals (highest posterior density intervals or equal-tailed intervals). This enables us to readily conduct statistical inference about such quantities or their functions and interpret relevant results.

To identify primary instances of a bag in the training cohort, we calculate the posterior inclusion probability:

$$\widehat{\pi}_{ij} = \frac{1}{T}\sum_{t=1}^{T}\delta_{ij}^{(t)},\ j = 1, ..., m_i,$$

where $\delta_{ij}^{(t)}$ is defined similarly as $\beta^{(t)}$. The instance j in bag i is primary if $\widehat{\pi}_{ij} > \theta$, where θ is a cutoff determined by controlling the Bayesian false discovery rate (FDR) [44] on all instances from the entire dataset. For a given θ, the estimated FDR is

$$\widehat{\mathrm{FDR}}(\theta) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m_i}(1-\widehat{\pi}_{ij})\cdot I(\widehat{\pi}_{ij} > \theta)}{\sum_{i=1}^{n}\sum_{j=1}^{m_i} I(\widehat{\pi}_{ij} > \theta)}.$$

In case when the denominator is zero, we define the FDR by 0. We choose the value of θ so that $\widehat{\mathrm{FDR}}(\theta) \leq \kappa$, where $\kappa \in (0, 1)$ is a pre-specified FDR we aim to control.

The above FDR control method can be used to identify positive bags in the training cohort as well, where the probability that bag i is positive, $\pi_i \equiv \mathrm{Pr}\,(y_i = 1 \mid X_i)$, can be

estimated by

$$\widehat{\pi}_i = \frac{1}{T}\sum_{t=1}^{T} \Phi\left(\sum_{j=1}^{m_i} \delta_{ij}^{(t)} x_{ij}\beta^{(t)}\right).$$

2.2.4. Prediction for new bags

We evaluate the performance of the proposed MICProB on the test bags using posterior-based prediction. Given a new bag B* with a collection of m^* instances described by the feature matrix $X^* = (x_j^*)_{j=1}^{m^*}$ with $x_j^* \equiv (1, x_{j1}^*, ..., x_{jd}^*)$, we predict the label of B* (i.e., y^*) based on the probability $\pi^* \equiv \Pr(y^* = 1 \mid X^*, \beta, \Delta^*) = \Phi\left(\sum_{j=1}^{m^*} \delta_j^* x_j^* \beta\right)$, where $\Delta^* = (\delta_j^*)_{j=1}^{m^*}$. This can be computed from the joint distribution of (Δ^*, Θ) given observed data: $p(\Delta^*, \Theta | X^*, X, y) = p(\Delta^* | \Theta, X^*) p(\Theta | X, y)$. To sample from $p(\Delta^*, \Theta | X^*, X, y)$, we sequentially draw (i) Θ from the joint posterior distribution $p(\Theta | X, y)$ and (ii) Δ^* from $p(\Delta^* | \Theta, X^*)$, that is $\delta_j^* \sim \text{Ber}\left(\Phi\left(x_j^* b\right)\right)$ for $j = 1, \ldots, m^*$, where b is obtained in step (i).

Thus, our model takes an integrated approach to produce posterior samples and predict the labels for new bags within the same Gibbs sampling iteration, as described in Figure 2.1(b). Recall that BMIR [48], the only Bayesian method based on the primary instances, has to rely on an auxiliary frequentist model to make prediction for new bags. To reduce the uncertainty in prediction, we sample R replicates of δ^* based on $\beta^{(t)}$ and $b^{(t)}$ within iteration t. In this illustration, $R = 50$. For each replicate $\delta^{*(r,t)}$, we calculate the probability of being positive for each test bag. An averaged probability for a new bag is then computed across all replicates and iterations as

$$\widehat{\pi}^* = \frac{1}{TR}\sum_{t=1}^{T}\sum_{r=1}^{R} \Phi\left(\sum_{j=1}^{m_i} \delta_j^{*(r,t)} x_j^* \beta^{(t)}\right).$$

Similarly, we can compute $\widehat{\pi}_j^* = \frac{1}{TR}\sum_{t=1}^{T}\sum_{r=1}^{R} \delta_j^{*(r,t)}$ to estimate the probability that instance j in B* is a primary instance. The FDR control method can be used to identify both

primary instances and positive bags as in Section 2.2.3.

2.3. Simulation

We conduct simulation to illustrate the performance of the proposed MICProB and compare it with 15 existing MIC methods. We consider various simulated scenarios under the PPI framework by varying key factors that may affect the performance, including sample size n, bag size m (i.e., the number of instances in a bag), number of features d, and mean proportion of primary instances denoted as $\overline{\text{PPI}}$ (i.e., the total number of primary instances divided by the total number of instances across all bags). We also conduct sensitivity analysis using data generated from the WR framework. For simplicity, we assume different bags in one dataset have a constant number of instances. For each generated dataset, we initialize the parameters of MICProB with random values and run 100,000 iterations of the Gibbs sampler and discard the first half as burn-ins. Standard diagnostic techniques [25] are used to detect the convergence of our MCMC algorithm.

2.3.1. Benchmark methods

Prior to MICProB, many algorithm-based solutions to MIC have been proposed, which, as mentioned in the introduction, can be categorized as instance-space (IS), bag-space (BS), or embedded-space (ES) methods [3], based on how they extract and exploit information from the MI data. For the purpose of performance evaluation in various simulated settings, we compare MICProB against 15 benchmark methods, including seven IS methods: EMDD, MI-SVM, mi-SVM, MILR, SI-SVM, SI-kNN, MILBoost [4, 7, 13, 15, 54, 74]; five BS methods: CkNN, NSK-SVM, EMD-SVM, miGraph, MInD [17, 23, 63, 72, 77]; and three ES methods: MILES, BoW, CCE [3, 16, 79]. We also refer readers to Xiong et al. [69] for more detail on each selected benchmark method.

We use the MATLAB "MILSurvey" toolbox, made available by Carbonneau et al. [13], to implement all benchmark methods except for MILR which is implemented via the R package `milr` [15]. 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. For model tuning, default ranges of values for hyper-parameters are used. Each selected IS method predicts bag labels from the predicted instance labels based on the standard MI assumption [19]. For MILR, it iterates 500 steps for the EM algorithm. Since feature selection is beyond the scope of this paper, we do not impose the LASSO penalty term, which is specified by default in MILR.

2.3.2. Settings

Our proposed MICProB is the only method developed based on the PPI framework while all the benchmark methods considered are designed under the WR framework. As pointed out by Xiong et al. [69], under the PPI framework, IS methods are less proper as the non-primary instances introduce irrelevant information to the bag; BS and ES methods may still be suitable for bag classification, as the information of primary instances of each bag could be utilized by the flexible embedding/summary behaviors of these methods. Thus, it would be interesting to compare their performance with that of MICProB using data generated from the PPI framework. For instance j in bag i, each covariate x_{ijr} is independently generated from a standard normal distribution, and the primary status indicator δ_{ij} is generated from a Bernoulli distribution Ber(p_{ij}), with $p_{ij} = \Phi\left(b_0 + \sum_{r=1}^{d} x_{ijr} b_r\right)$, where b_0 and b_r for $r = 1, ..., d$ are regression coefficients in the probit regression model for δ_{ij}. Next, we simulate the bag label y_i from $\mathrm{Ber}(\pi_i)$, with $\pi_i = \Phi\left(\beta_0 + \sum_{j=1}^{m} \delta_{ij} \sum_{r=1}^{d} x_{ijr}\beta_r\right)$, where β_0 and β_r for $r = 1, ..., d$ are regression coefficients associated with the probit model for π_i. We adjust the intercepts b_0 to vary $\overline{\text{PPI}}$. We set $b_j = 1$ for $j = 1, ..., d$, $\beta_0 = 0.5$, $\beta_j = 1$ for $j = 1, ..., \lceil d/2 \rceil$ and $\beta_j = -0.5$ for $j = \lceil d/2 \rceil + 1, ..., d$. We vary $n \in \{150, 300, 450, 600\}$; $m \in \{5, 10, 20, 40\}$; $d \in \{2, 15, 30, 45\}$;

and $\overline{\text{PPI}} \in \{0.1, 0.4, 0.6, 0.9\}$ and assess their influence on performing multiple instance classification. We employ the vary-one-at-a-time strategy to reduce the work load in this simulation; that is, we vary one and only one factor each time while fixing the others at the basic setting in which $n = 300$, $m = 10$, $d = 30$, and $\overline{\text{PPI}} = 0.4$. We independently generate 50 replication datasets under each of the settings. The performance is measured by evaluating the area under the Receiver Operating Characteristic curve (AUROC) on 300 test bags in each replicate.

To further examine the robustness of MICProB, we generate data from the WR framework as well. Following Xiong et al. [69], we generate MI datasets by varying WR $\in \{0.05, 0.25, 0.5, 0.75, 1\}$, where WR $= 0.05$ represents the scenario where there is only one positive instance in each bag. For more details, we refer readers to Section 4.1 of Xiong et al. [69].

2.3.3. Results

Figure 2.2 compares the performance of MICProB with 15 benchmark methods for bag classification in various simulated scenarios under the PPI framework. Each line is the average AUROC (%) calculated from 50 replications. Across all scenarios, MICProB works best, followed by NSK-SVM, EMD-SVM, and MInD, all from the BS category. MILES from the ES category and MILR from the IS category are middle performers. All the remaining IS methods, miGraph and CkNN from the BS category, do not yield satisfactory performance in most scenarios, with AUROC below 70%, which is slightly better than random guessing. Notably, MILR is the best performing IS method among the selected ones, which shows some promise for statistical model-based approaches in tackling MI problems.

Next, we discuss the impact of each factor on the performance of MICProB and benchmark methods, excluding the bottom performers, which steadily have poor performance.

Firstly, increasing the sample size n tends to improve the performance. Secondly, as the bag size m increases, while the performance of MICProB shows a non-monotone pattern, the other methods are not sensitive to the change. In particular, the average AUROC of MICProB is the highest (above 80%) regardless of the bag size. Thirdly, as the number of features d increases, MICProB has improved performance, while all the benchmark methods show an opposite pattern. We note that the signal in data generated from the PPI framework becomes stronger in general as d goes up. Among all, only MICProB can capture the stronger signal because it is the only method designed for the PPI framework. Lastly, when $\overline{\text{PPI}}$ increases, most methods show higher AUROC by capturing the increased amount of useful information. Individual performance of each method evaluated on 50 replicates at different values of $\overline{\text{PPI}}$ is shown using box plots in Figure 2.3. We observe that MICProB performs significantly better, with median AUROC greater than 80%, than all the benchmark methods, when there are only 10% primary instances on average in each bag. As $\overline{\text{PPI}}$ increases, the performance of MICProB steadily improves and the spread (i.e., the width of the box) becomes narrower, while the performance for many other methods is more variable across different replications. More detail on individual performance of each method on 50 replicates by varying the sample size, bag size, and the number of features, are shown using box plots in Figures B.1-B.3. In general, MICProB produces consistently better performance with narrow spread.

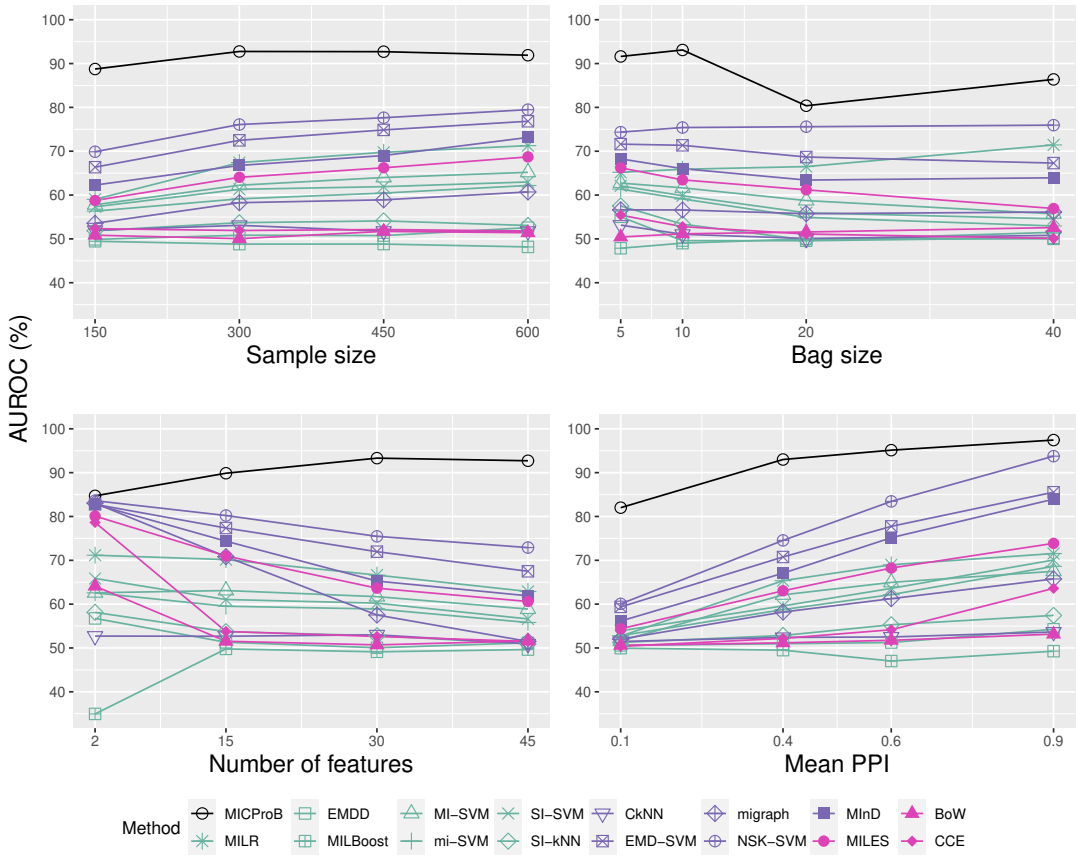


Figure 2.2: Simulation evaluation under the PPI framework: average AUROC (%) for bag classification using different MIL methods, evaluated on simulation scenarios each with 50 replicates. We vary the sample size, bag size, number of features, and mean PPI, and report the results in the four panels, respectively. Benchmark methods are distinguished by color (green: IS methods; purple: BS methods; magenta: ES methods).

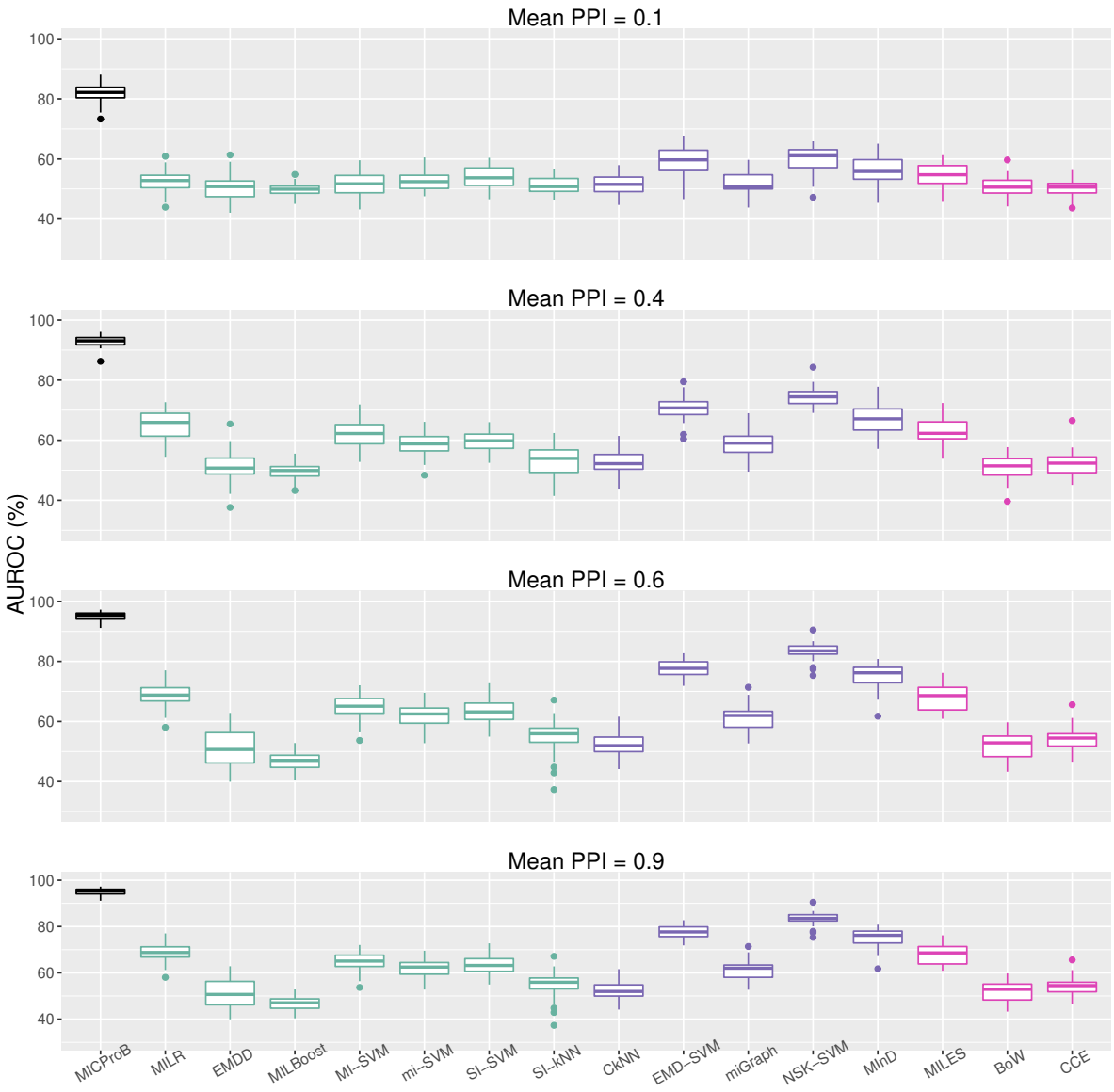


Figure 2.3: Simulation evaluation under the PPI framework: AUROC (%) for bag prediction using different MIL methods. We vary the mean PPI and report results for each of the methods in each setting using a box plot (based on 50 replicates). Benchmark methods are distinguished by color (green: IS methods; purple: BS methods; magenta: ES methods).

Figure 2.4 shows the performance of MICProB for identifying primary instances. Each boxplot is generated by AUROC values calculated from 50 replicates. In many scenarios, the proposed method works quite well, with AUROC greater than 95%. It only dips below 90% in only a few settings. Next, we discuss how each factor affects the performance

of MICProB. As in our observations made for bag classification, the performance for instance classification tends to improve with an increased sample size (n) or feature size (d) or mean proportion of primary instances ($\overline{\text{PPI}}$). Among these three factors, it seems that d has a larger impact than n or $\overline{\text{PPI}}$. Secondly, the performance decreases with an increased bag size (m), which could be due to noisy signal induced by more non-primary instances in larger bags.

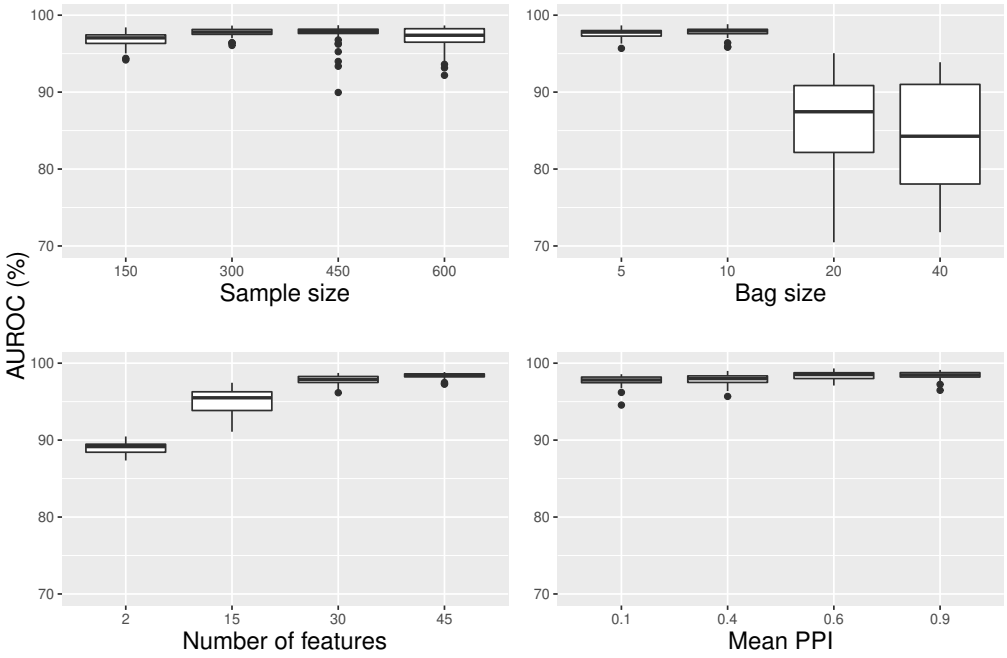


Figure 2.4: Simulation evaluation under the PPI framework: AUROC (%) for identifying primary instances using MICProB. We vary the sample size, bag size, number of features, and mean PPI, and report results for each of the methods in each setting using a box plot (based on 50 replicates) in the four panels, respectively. Note that all benchmark methods do not offer the functionality of identifying primary instances.

For robustness checking, Figure 2.5 compares the performance of MICProB with the 15 benchmark methods for bag classification using data generated from the WR framework. As we expect, MICProB is no longer the best method when WR ≤ 0.25, especially when WR $= 0.05$, where in each bag there is only one positive instance. This is because the presence of a large number of negative instances makes it difficult for MICProB to

identify any primary instances assumed under the PPI framework. Still, when WR $= 0.25$, it is much better than a few methods that are specially designed under the WR framework including MILBoost, EMDD, SI-kNN, CCE, and BoW. As WR further increases, MICProB has as good performance as other benchmark methods, with AUROC very close to 1. Individual performance of each method on 50 replicates at different values of WR is shown using box plots in Figure B.4, which clearly shows that as WR increases, the performance of MICProB steadily improves and the spread becomes narrower, demonstrating an amazingly high degree of robustness.

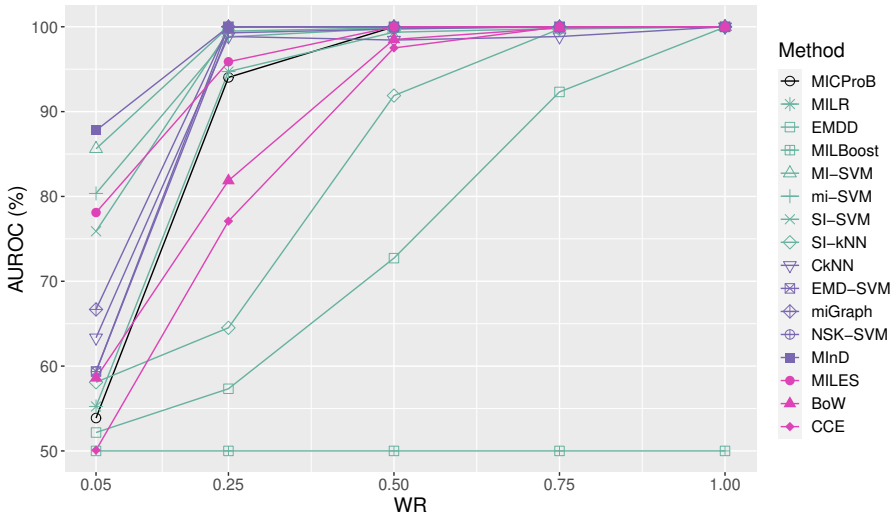


Figure 2.5: Simulation evaluation under the WR framework for robustness checking: average AUROC (%) for bag classification using different MIL methods, evaluated on simulation scenarios each with 50 replicates by varying WR. Benchmark methods are distinguished by color (green: IS methods; purple: BS methods; magenta: ES methods).

We provide the runtime information for MICProB and the 15 benchmark methods in Figure B.5. Due to sequential updates of the MCMC algorithm that cannot be easily parallelized, MICProB is not as computationally efficient as most competing methods. Nevertheless, for data generated from the PPI framework, MICProB outperforms all the benchmark methods including MILR, the only other regression-based approach, and enables statistical inference that optimization-based methods do not provide. For data from

the WR framework, it appears that MICProB is not overly sensitive and still has strong performance as long as WR is not too low, compared to the top performers.

2.4. Real Data Examples

We present two data examples: the first is on cancer detection using sequencing data from The Cancer Genome Atlas (TCGA), to evaluate the performance of our proposed MICProB against benchmark methods on detecting various types of cancer; the second is on modeling immunogenic neoantigens, to illustrate the explainability of MICProB.

2.4.1. Cancer detection using T-cell receptor sequences

Early diagnosis, especially for aggressive cancer types, is crucial for patients to receive appropriate treatments for best possible prognosis. Various tools have been developed to facilitate cancer screening, which, however, are less ideal for detecting certain types of cancer [12, 18, 59]. One possible new approach for cancer detection is to examine the TCR sequences in peripheral blood of patients, as TCRs are used by the T cells to target and initiate the destruction of tumor cells, and may contain critical information regarding tumor progression in the human body. This approach also has the advantage of being non-invasive as it requires blood samples. The problem can be formulated into the MIL framework by treating each patient as a bag with voluminous TCR data (instances).

Aiming to comprehensively explore genomic changes involved in human cancer, TCGA collected and analyzed tissue samples from patients of over thirty cancer types and obtained genomic data for each sample using next-generation sequencing techniques, such as RNA-sequencing, whole exome-sequencing, etc. We use MiTCR [10], a commonly used TCR reconstruction software to reconstruct TCRs from the RNA-sequencing data. 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.

Under the MIL framework, each sample is considered as a bag consisting of TCR sequences (instances) represented by text strings of amino acids. In order to make it convenient for MIL methods to utilize the physicochemical properties of TCRs, we embed each TCR sequence into a d-dimensional numeric vector using a deep learning auto-encoder, which has been systematically validated in our previous work [40, 75]. In this study, each instance is described by 31 features. The first 30 dimensions represent the embedded TCR and the last feature is the log-transformed abundance for each TCR sequence appeared in each bag.

We apply MICProB to tissue samples of 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) [36, 39, 43], to illustrate its utility on distinguishing cancer patients from healthy individuals via TCRs. 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 each sample).

In the TCGA data, the number of positive bags (tumor samples) is much greater than that of negative bags (normal tissue samples), as TCGA is mainly focused on studying cancer patients. To adjust for the imbalanced TCGA data (more positive bags than negative bags), we randomly sample from 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 more than 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.

We randomly sample about 50% of the 405 negative bags (i.e., 202 normal tissues samples) to reduce the computation time. For each of the selected cancer types, we create a balanced dataset with 50% positive and 50% negative bags, as advised in He and Garcia [28] that it is often preferred to apply machine learning methods to balanced data. As a result, for DLBC, THYM, and ESCA, due to a small number of positive bags, the sample sizes are 90, 216, and 332, respectively; for each of the remaining cancers, the total sample size is 404. Figure 2.6 shows the number of instances for selected cancer types after pre-processing, reflecting a more realistic situation in real data that the number of instances varies across bags. We also observe for each of the ten cancer types, the distribution of bag size is severely right-skewed, where most bags have a relatively small number of instances but a few can have many more instances. We standardize input variables so that they all have zero mean and unit standard deviation.

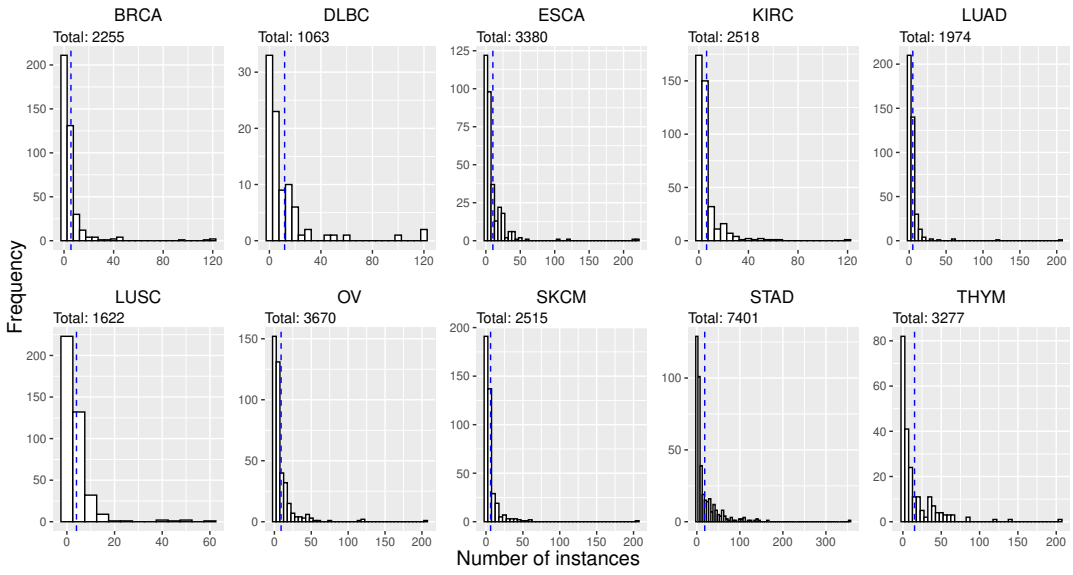


Figure 2.6: TCGA data: the number of instances for selected cancer types. Blue dashed line indicates sample mean.

For MICProB model training and validation, we employ a 10-fold cross-validation (CV) procedure. We run the derived Gibbs sampler for $100,000$ iterations and discard the first half as burn-ins. We illustrate convergence diagnostics for MICProB using five independent MCMC chains in Figures B.6 and B.7. Prediction for bags in the held-out fold is performed in an integrated manner, as shown in Figure 2.1 (b). Further, in this study, we take the average of the primary instances to measure their contribution to the bag. For benchmark methods, a nested cross-validation (CV) procedure [13, 14] is deployed, in which the model is tuned (i.e., the hyper-parameters are optimized over a range of values) in the inner layer CV and the performance of the fitted model is evaluated in the outer layer CV.In our implementation, both inner and outer layers have ten folds. We calculate the average performance from CV in terms of AUROC of each method.

Table 2.1 shows the performance of each method by cancer type. We only include seven cancers with average AUROC across all methods greater than 60% and exclude the other three (BRCA, LUAD, and LUSC) for which none of themethods works adequately. For each cancer type, the performance of MICProB is always (much) higher than the average. More importantly, MICProB works best in four (KIRC, SKCM, DLBC, and THYM) out of seven cancers. Notably, KIRC and SKCM are well known immunogenic cancer types with high levels of T-cell infiltration [64]. MICProB achieves higher performance than the benchmark methods in the presence of bystander effects [33, 68], that is, the strong T-cell activation in these cancers may have caused infiltration of both abundant tumor-specific and non-specific T cells in the tumor, creating additional difficulty for MIL to distinguish tumor versus normal samples. As a result, with average AUROC at 78.3% across all cancers, MICProB gives the best performance compared to benchmark methods, followed by EMD-SVM, with average AUROC at 78.1%, and NSK-SVM, with average AUROC at 76.3%. For THYM, the performance of MICProB (AUROC of (87.7 ± 2.2)%) is significantly better than the second best method MInD (AUROC of (84.7 ± 0.6)%). For STAD and ESCA, the performance of MICProB also stays in the upper range. Lastly, the performance of the methods depends on cancer type. For STAD, except EMDD, MIL-

Boost, and CkNN, all other methods can achieve AUROC at least 75%. For KIRC and SKCM, while MICProB performs the best, the average performance across all methods is below 70%.

		KIRC	SKCM	DLBC	ESCA	OV	THYM	STAD
	Overall	63.2	62.0	66.5	71.3	72.1	75.2	77.8
MICProB	**78.3**	**68.1** (2.5)	**69.6** (2.2)	**82.2** (5.1)	78.2 (3.7)	76.7 (3.0)	**87.7**(2.2)	85.6 (1.5)
MILR	74.7	66.1 (4.0)	66.0 (2.1)	64.4 (7.5)	80.5 (3.5)	78.7 (1.6)	81.0 (2.4)	86.0 (2.9)
EMDD	60.1	54.4 (4.1)	50.1 (3.6)	55.3 (4.3)	67.5 (4.2)	57.8 (2.3)	65.0 (5.7)	70.3 (6.5)
MILBoost	51.3	56.5 (2.4)	48.9 (1.6)	47.0 (4.6)	49.9 (1.3)	55.1 (4.1)	49.7 (3.0)	51.9 (0.6)
MI-SVM	72.3	66.2 (1.5)	64.6 (1.4)	69.6 (3.5)	72.9 (1.3)	72.6 (1.4)	80.2 (1.3)	79.9 (1.2)
mi-SVM	71.2	66.1 (1.6)	66.5 (1.7)	66.0 (1.8)	69.4 (2.1)	73.2 (1.3)	77.0 (1.4)	80.3 (1.3)
SI-SVM	71.3	66.3 (1.0)	66.5 (1.1)	66.6 (2.3)	72.2 (2.2)	73.3 (0.8)	75.1 (2.0)	78.8 (1.5)
SI-kNN	71.0	65.1 (1.1)	65.1 (1.5)	65.0 (2.7)	72.6 (0.7)	74.7 (1.0)	74.7 (1.8)	79.5 (0.9)
CkNN	50.2	52.4 (1.0)	54.7 (2.2)	55.2 (4.2)	46.4 (1.3)	60.9 (1.3)	36.5 (1.8)	45.6 (2.0)
EMD-SVM	78.1	66.4 (1.2)	65.3 (1.5)	78.1 (2.5)	**83.1** (0.5)	**82.7** (0.6)	84.0 (1.0)	**87.3** (0.5)
miGraph	67.8	61.6 (1.6)	60.4 (0.7)	65.9 (5.0)	60.7 (1.4)	68.7 (1.0)	78.9 (0.7)	78.7 (0.8)
NSK-SVM	76.3	67.5 (0.7)	65.4 (1.0)	68.6 (2.6)	80.8 (0.7)	81.7 (0.8)	83.8 (1.2)	86.2 (1.0)
MInD	73.2	58.3 (1.3)	58.5 (2.6)	78.1 (3.7)	76.9 (1.6)	69.5 (1.2)	84.7 (0.6)	86.4 (0.6)
MILES	75.0	**68.1** (0.9)	63.7 (1.7)	67.7 (3.1)	77.8 (1.1)	79.6 (0.5)	83.1 (0.8)	84.9 (0.7)
BoW	74.0	65.2 (0.9)	64.1 (1.8)	65.4 (4.6)	78.6 (0.6)	74.7 (1.2)	84.2 (1.5)	85.9 (0.3)
CCE	75.6	65.4 (1.1)	66.1 (0.6)	66.9 (2.5)	82.0 (0.7)	79.8 (0.6)	83.2 (1.0)	85.8 (0.5)

Table 2.1: TCGA data: average AUROC (%) with standard error given in parentheses for predicting bag labels for each method across seven cancers. The highest AUROC is highlighted in bold. The average performance across all methods is shown below each cancer type.

2.4.2. Modeling immunogenic neoantigens

We present another data example of modeling immunogenic neoantigens to demonstrate the high explainability of MICProB. Neoantigens are short peptides presented by the major histocompatibility complex (MHC) proteins on the surface of tumor cells, which serve as recognition markers for cytotoxic T cells via T-cell receptors. As one of the most fundamental and unsolved questions in tumor immunology, the relationship between tu-

mor immune responses and tumor neoantigens is the key to understanding the inefficiency of immunotherapy observed in many cancer patients. However, it is often a challenging task to quantify this relationship as the the properties of neoantigens that can elicit immune responses remain unclear. This biological problem is investigated in the MIR context by Park et al. [48], who modeled multiple instances (neoantigens) within each bag (patient specimen) with the continuous response (T-cell infiltration). Each instance is characterized by covariates of neoantigen qualities.

We use neoantigen data from several existing studies [42, 43, 58, 64]. To apply MICProB to the neoantigen data, we code samples with T-cell infiltration greater than or equal to 5 as one (high-level infiltration) and smaller than 5 as zero (low-level infiltration), resulting in 36% positive bags out of 728 bags. The distribution of numbers of neoantigens (instances) from different patients (bags) is shown in Figure B.8(a), with mean being 113, median being 30, and maximum being 664. Figure B.8(b) shows the distribution for each of the six covariates that describe neoantigens along the x-axis. These covariates include hydrophobicity (hydro), similarity to pathogenic epitopes (blast), rank of binding affinities to major histocompatibility complex molecules (perc_rank), an immunogenicity score previously established for class I neoantigens only (neoantigens could bind to both class I and class II human leukocyte antigen (HLA) molecules) (immune), transport efficiency (TAP), and the mutation type (mut_type, 1 for missense mutations, and 0 for insertions/deletions, stoploss mutations). We standardize all continuous variables to have a zero mean and unit standard deviation and leave the binary variable in its original scale.

We randomly split the dataset ten times and in each split, we use 75% data to train MICProB and the remaining 25% data to evaluate the performance. We run our sampler for 100,000 iterations and discard the first half as burn-ins. The average AUROC across ten random splits is 99%. The posterior mean estimates for the intercept, hydro, blast, perc_rank, immune, TAP, and mut_type are $-1.162, 0.087, 0.092, 6.473, -0.129, -0.111, 0.420$, respectively. Thus, according to the signs of these estimates, higher hydrophobicity,

higher similarity to pathogenic epitopes, and higher rank of binding affinities to major histocompatibility complex moleculesare, and missense mutations tend to increase the likelihood of high infiltration; meanwhile, the estimates of class I immunogenicity scores and TAP activity, are negative, and thus tend to decrease the likelihood of high infiltration. We also provide interval estimates of covariate effects from MICProB and estimates with standard errors from MILR using the default setting in Table B.2. We find that the two regression-based methods agree on the directions of effects of blast, perc_rank, TAP, and mut_type. They also agree that the intercept and the effect of perc_rank is statistically signifiant while the others are not. Lastly, we point out that other benchmark methods do not offer such interpretability via regression coefficients reflecting covariate effects.

2.5. Discussion

In MIL literature, methods for classification are exclusively based on the WR framework, under which the functionality of identifying primary instances is not offered. Further, these methods are mainly optimization-based and hence suffer from poor explainability. Under the PPI framework [69] that is much less explored in the MIC literature, we develop a novel Bayesian hierarchical model, MICProB, to learn from multiple instance data with a binary response and identify both primary instances and bag labels. Specifically, MICProB is composed of two nested probit regression models, where the inner model is estimated for predicting primary instances (i.e., predicting δ_{ij} from x_{ij}), and the outer model is estimated for predicting bag-level responses based on the primary instances. Thanks to its fully Bayesian formulation, prediction for new bags can be performed in an integrated manner via posterior predictive sampling. Furthermore, MICProB enables convenient statistical inference for quantities related to model parameters with posterior samples drawn. Regression coefficients that reflect covariate effects on the bag-level response are explicitly estimated, hence offering high explainability to the model, as demonstrated in the application to the neoantigen data.

Due to its special design for the PPI data generation mechanism, we recognize that MICProB does not belong to any of the existing categories of MIC methods of IS, BS, or ES paradigm. MICProB is not an IS method as the prediction step does not occur at the instance level (i.e., predicting whether an instance is positive or negative). Further, MICProB is not a BS method as there is no distance computed between each pairs of bags. Lastly, it might be tempting to view MICProB as an ES method. But as opposed to mainstream ES methods that are vocabulary-based, which use instances from training data to build a dictionary for feature embedding, MICProB does not have the embedding step that maps the original feature space to a new one. Thus, the proposed method does not fall under the ES category. Given above, MICProB provides a fresh perspective to the development of new MIL methods that are model-based and tailored for MI data with the concept of primary instances.

MICProB yields significantly better performance in various simulated scenarios than the 15 benchmark methods. Based on the assumption that the bag label is determined by primary instances in each bag, our model is capable of identifying these instances with high accuracy, even though they are not known in advance. Given that there is no such method that works universally well in real data application of cancer diagnosis, the proposed method performs the best in four out of seven cancer types. Across the seven cancer types, MICProB also has the highest performance on average, suggesting that the PPI framework is more likely to represent the underlying data generation mechanism for this particular application.

We make our code available at https://github.com/danyixiong/MICProB. A user of MICProB has the flexibility of choosing between the "sum" or "average" contribution of primary instances, according to the user's perception of the underlying data generation process or results from cross validation. MICProB can also be implemented using different priors, to incorporate various forms of prior knowledge. For example, in the real data application, we experiment with marginally non-informative prior distributions for covari-

ance matrices Σ_β and Σ_b, as suggested by Huang and Wand [31], to reflect our vague information on the regression coefficients. The resulting MCMC algorithm can still be conveniently implemented via Gibbs sampling with data augmentation technique (see Section B.2.2 for technical detail). However, this prior elicitation is not preferred as its resulting performance (Table B.1) is worse than that of simpler prior specifications for the original MICProB. Finally, it is also possible to consider a logistic model for the binary response variable. Under this formulation, however, the Bayesian inference would become harder due to the analytically inconvenient form of the model's likelihood function. A potential direction would be to employ a data augmentation strategy using Pólya-Gamma latent variables [50]. With these flexible options of design at different parts of the model, MICProB thereby provides a generic framework for developing future model-based statistical methods that are dedicated to addressing MI problems where primary instances need to be identified.

In the era of big data, we envision an increasing need for MIL to handle increasingly complex structures of real-world data. Advances in the biomedical research domain, in particular, propel the development of novel MIL techniques, especially Bayesian methodologies that can naturally incorporate prior beliefs into observed data, because the complicated nature of biological and medical applications necessitates consideration of prior knowledge available from domain experts or past studies, to narrow the search space of the MIL model for the observed new data. By developing MICProB, we provide a successful example of how statistical learning tackles an MIL problem, and we believe there is a broad space for new Bayesian MIL methods with diverse capacities to emerge to capture various characteristics of real-world data.

APPENDIX A

APPENDIX of CHAPTER 1

This appendix provides additional results from our simulation and TCGA data analysis mentioned in Chapter 1.

A.1. Additional simulation results for bag classification based on AUPRC

We also compare the performance of bag classification for each method in terms of AUPRC. For model I (Figures A.1), in general, we observe that an MIL method with higher AUROC also has higher AUPRC. For model II (Figure A.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.

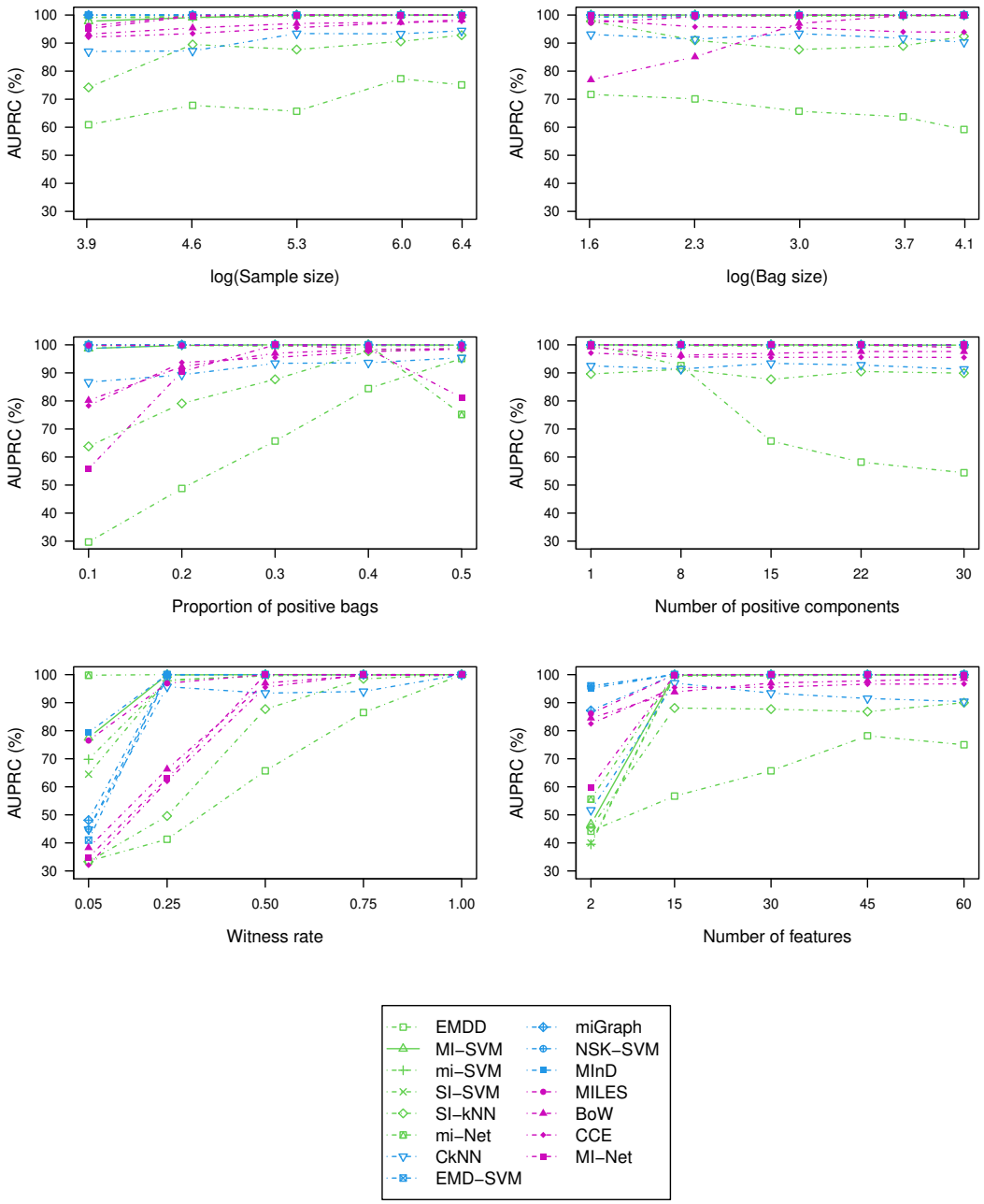


Figure A.1: Mean AUPRC (%) of bag classification using different MIL methods, evaluated on simulation scenarios each with 100 replicates generated under model I. IS/BS/ES methods are distinguished by green, blue, and magenta lines.

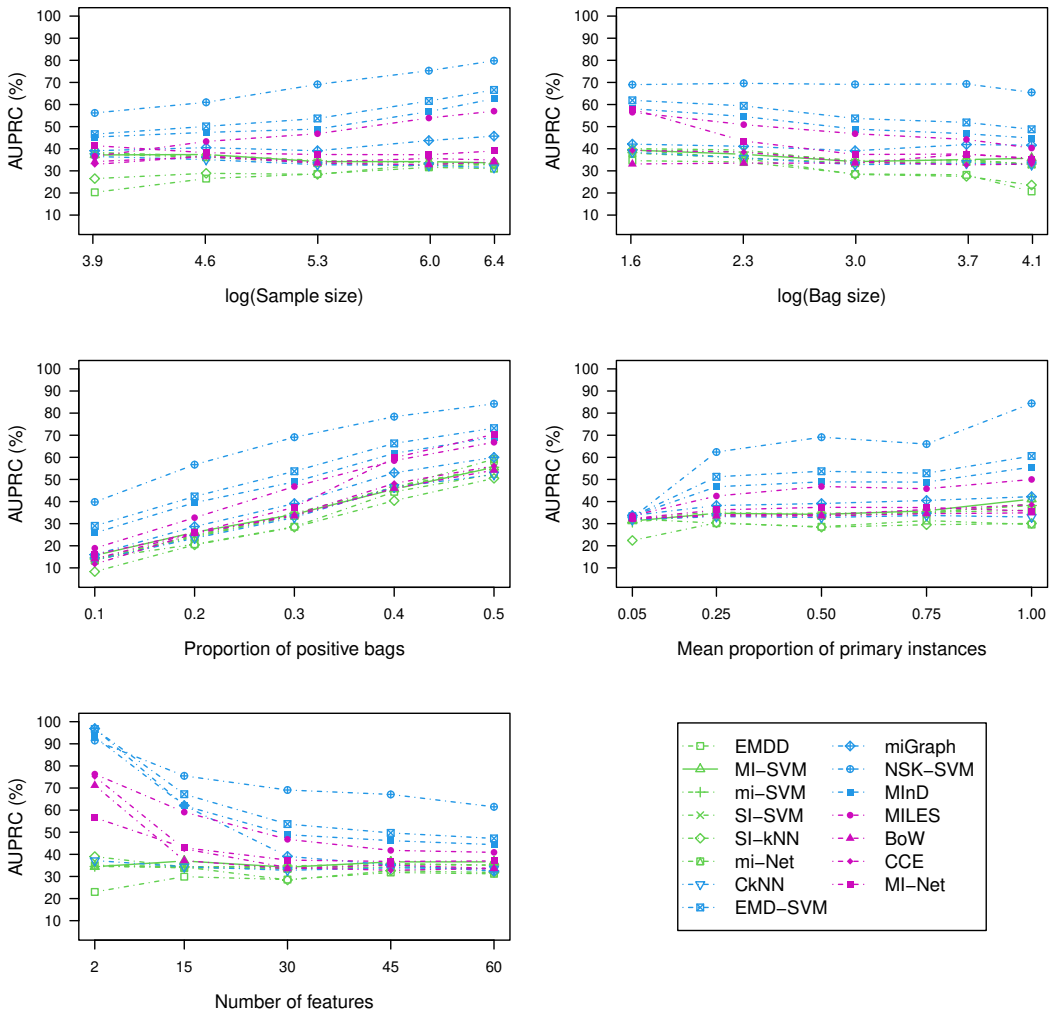


Figure A.2: Mean AUPRC (%) of bag classification using different MIL methods, evaluated on simulation scenarios each with 100 replicates generated under model II. IS/BS/ES methods are distinguished by green, blue, and magenta lines.

A.2. Simulation results for instance classification under model I

We show the performance of instance classification for six MIL methods, evaluated by AUROC in Figure A.3. It is observed that IS methods are generally more capable of classifying positive and negative instances than MILES (an ES method).

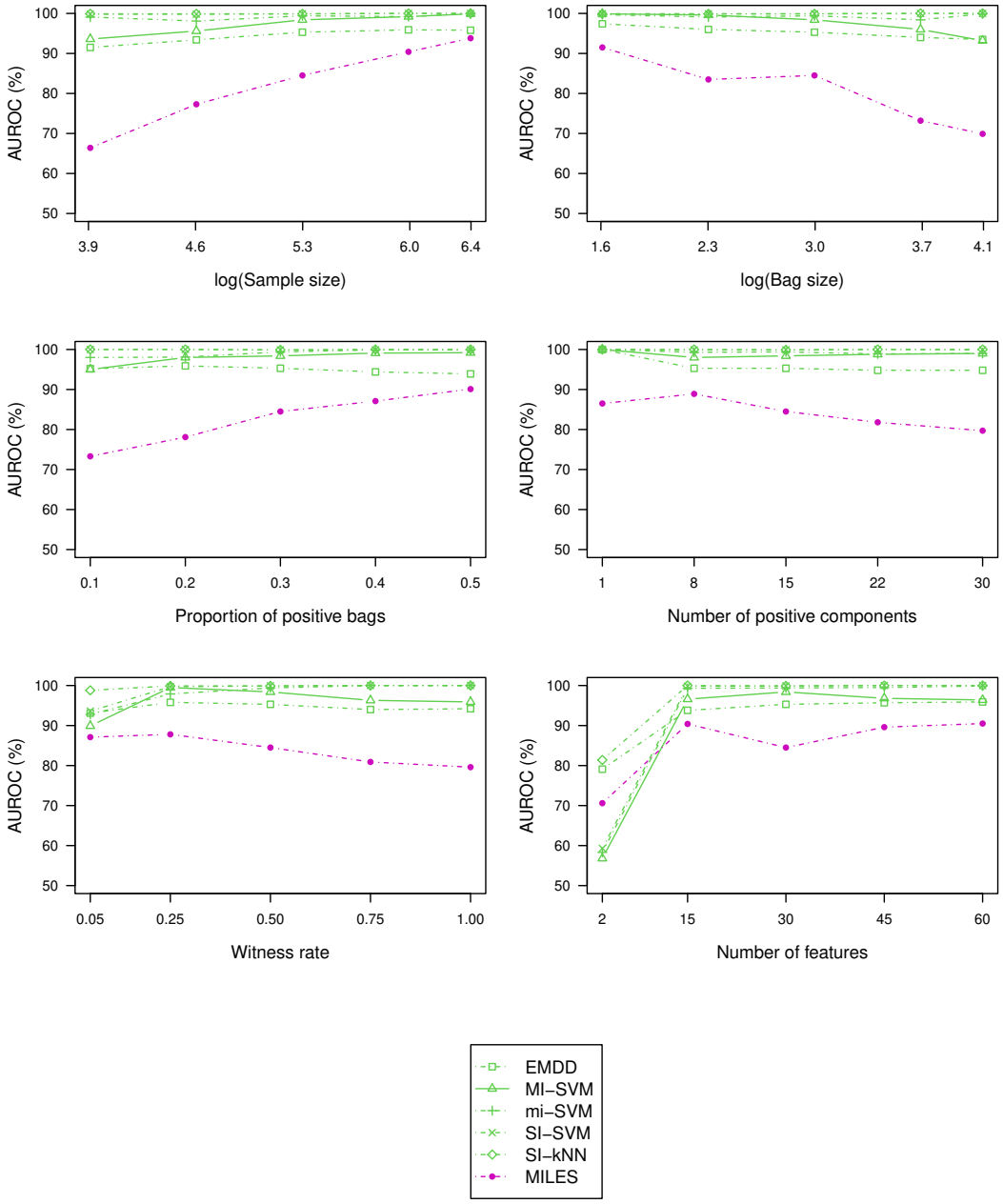


Figure A.3: Mean AUROC (%) of instance classification using different MIL methods, evaluated on simulation scenarios each with 100 replicates generated under model I. IS/ES methods are distinguished by green and magenta lines.

A.3. Additional results for TCGA data examples

Figure A.4 shows the number of tumor versus normal tissue samples for each of the cancer types. Since TCGA focusses on studying cancer patients, the number of positive bags (tumor samples) is much greater than that of negative bags (normal tissue samples). We further plot the performance of individual method on five cancer types in Figures A.5 and A.6.

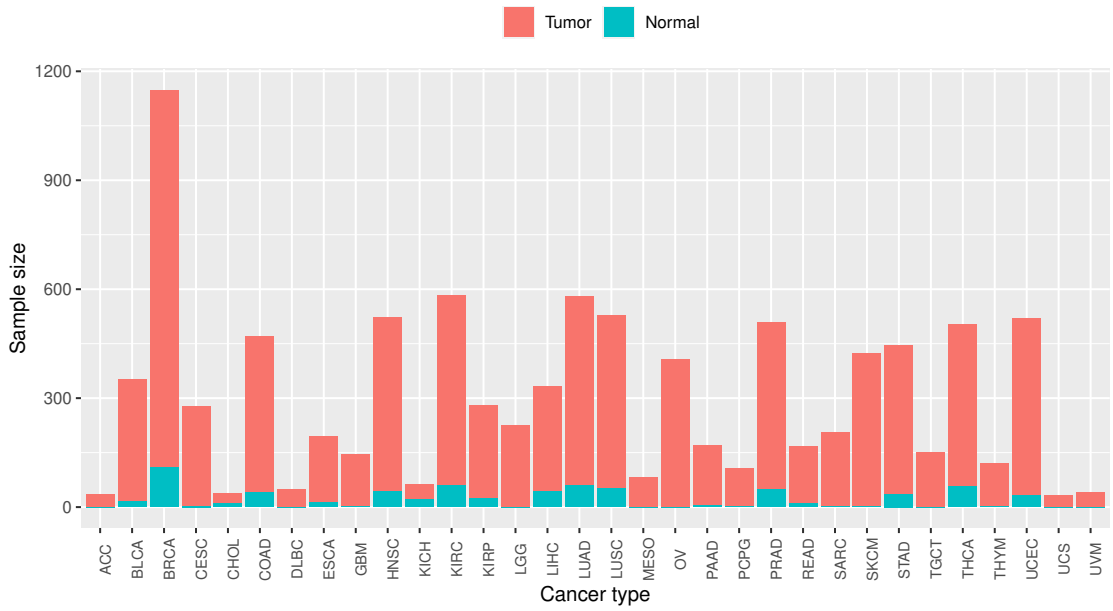


Figure A.4: TCGA data: numbers of tumor samples (positive bags) and normal tissue samples (negative bags) for over thirty cancer types.

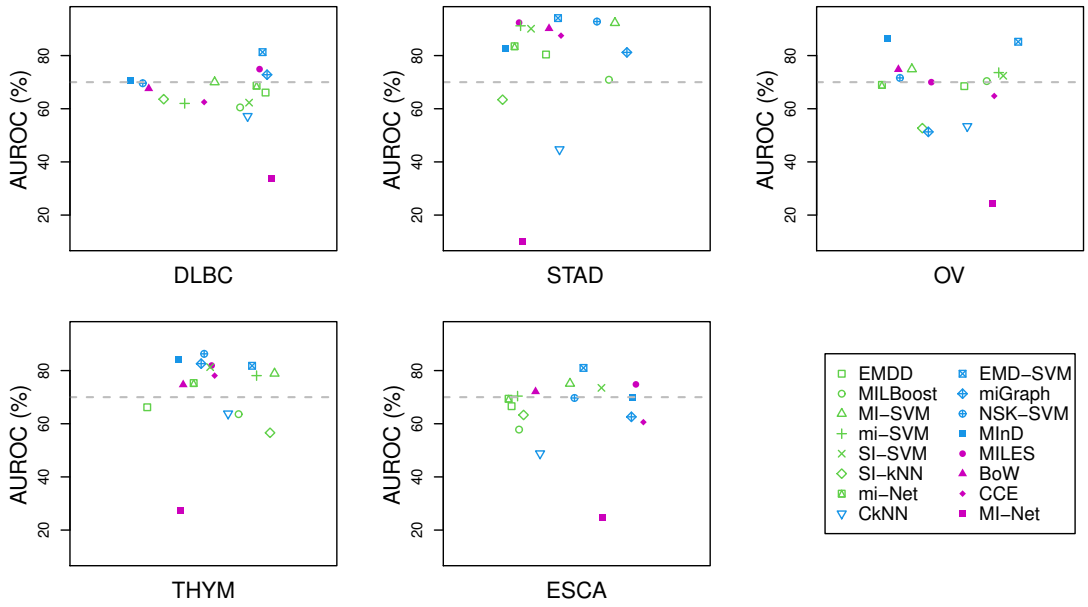


Figure A.5: TCGA data with 10% positive bags: mean AUROC (%) for DLBC, STAD, OV, THYM, and ESCA. The gray dashed line corresponds to 70% AUROC. MIL methods are distinguished by symbol shapes. Categorization of MIL methods is distinguished by color (green: IS methods; blue: BS methods; magenta: ES methods).

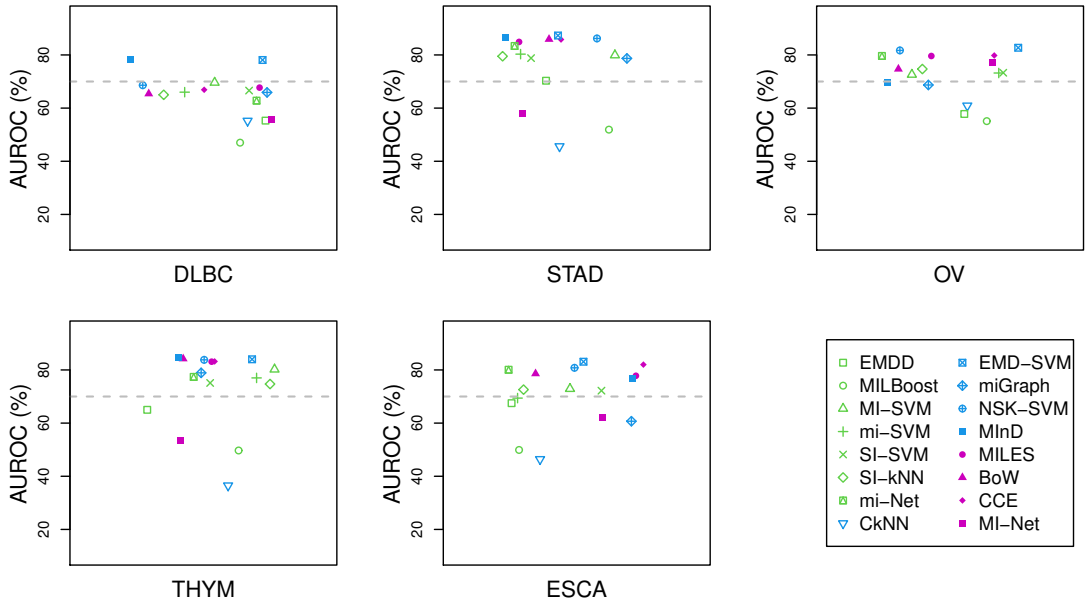


Figure A.6: TCGA data with 50% positive bags: mean AUROC (%) for DLBC, STAD, OV, THYM, and ESCA. The gray dashed line corresponds to 70% AUROC. MIL methods are distinguished by symbol shapes. Categorization of MIL methods is distinguished by color (green: IS methods; blue: BS methods; magenta: ES methods).

APPENDIX B

APPENDIX of CHAPTER 2

This appendix provides additional results for simulation and real data examples mentioned in Chapter 2.

B.1. Additional simulation results

B.1.1. Performance on bag classification

We show the predictive performance on bag classification of MICProB versus benchmark methods in Figures B.1-B.3 using data generated from the PPI framework under various simulation settings. Each box plot is generated based on 50 replicates of test data. Parameter settings for sample size (n), number of instances per bag (m), and number of features (d) are the same as mentioned in Section 3.1. In all the settings, MICProB consistently outperforms the other methods.

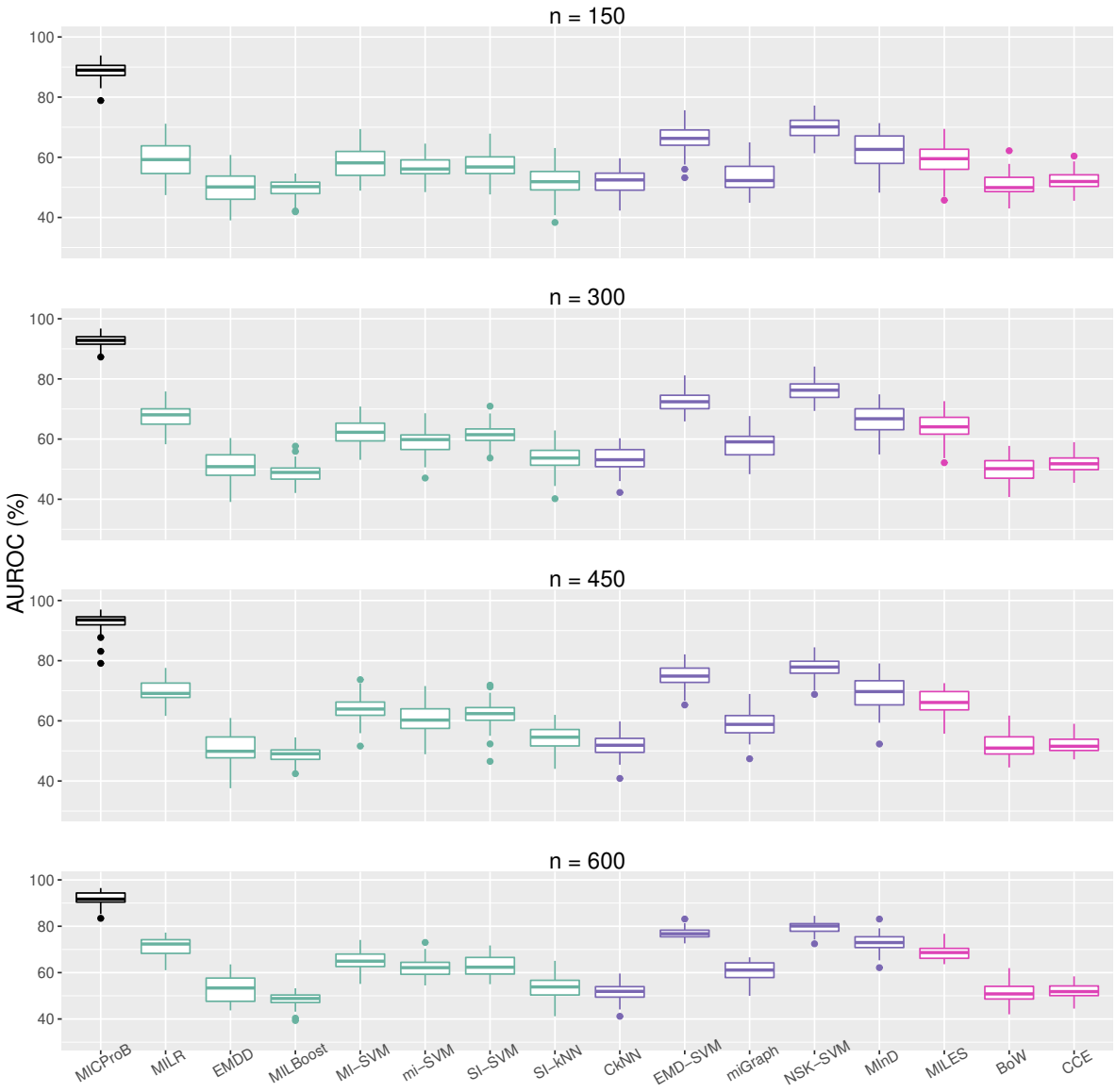


Figure B.1: Simulation evaluation under the PPI framework: AUROC (%) for bag prediction by varying the sample size (number of bags), evaluated on 50 replicates. Benchmark methods are distinguished by color (green: IS methods; purple: BS methods; magenta: ES methods).

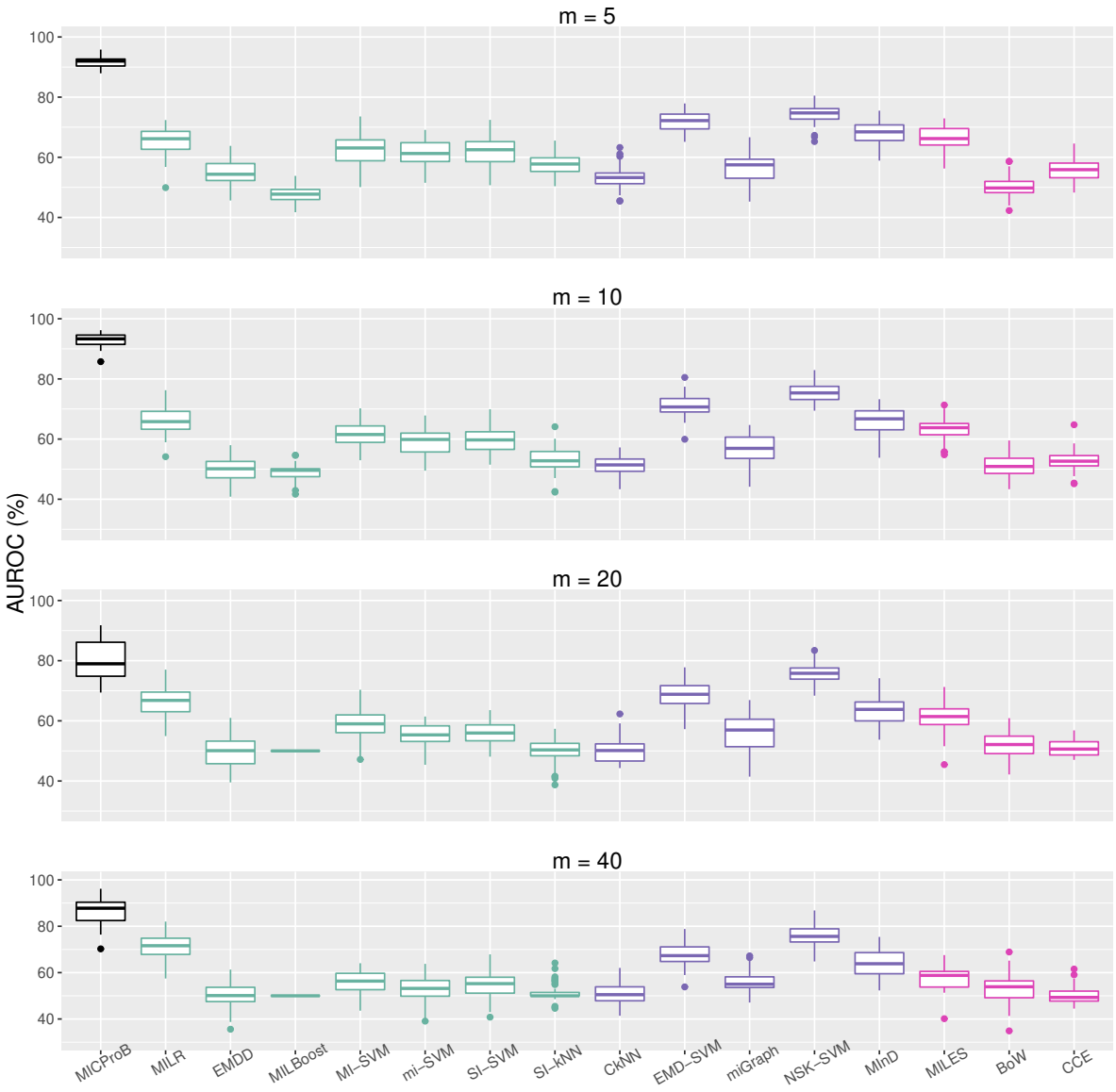


Figure B.2: Simulation evaluation under the PPI framework: AUROC (%) for bag prediction by varying the bag size (number of instances per bag), evaluated on 50 replicates. Benchmark methods are distinguished by color (green: IS methods; purple: BS methods; magenta: ES methods).

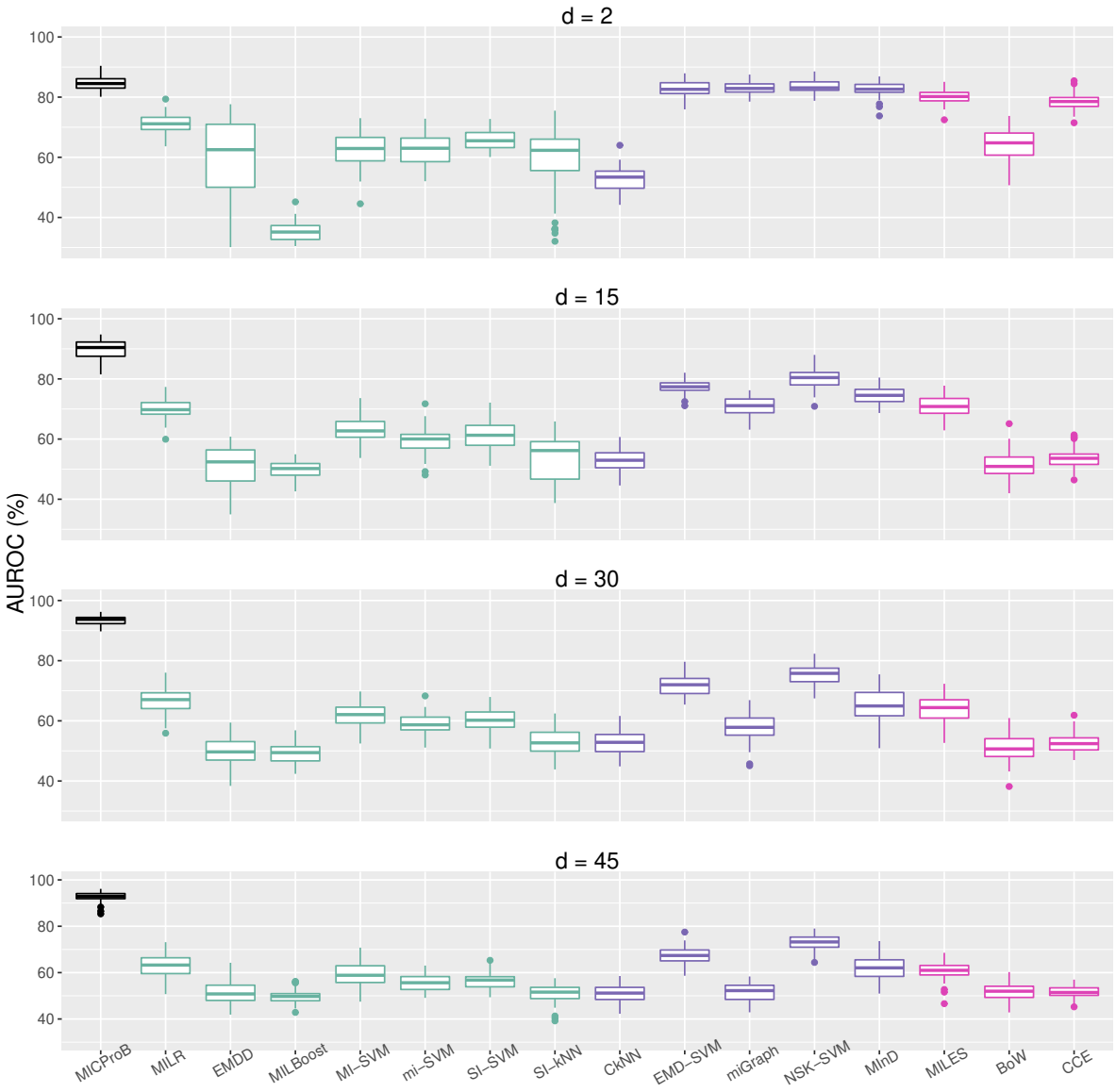


Figure B.3: Simulation evaluation under the PPI framework: AUROC (%) for bag prediction by varying the number of features, evaluated on 50 replicates. Benchmark methods are distinguished by color (green: IS methods; purple: BS methods; magenta: ES methods).

Surprisingly, for MI data generated from the WR framework, the proposed MICProB demonstrates a high degree of robustness, especially when WR ≥ 0.25, as shown in Figure B.4.

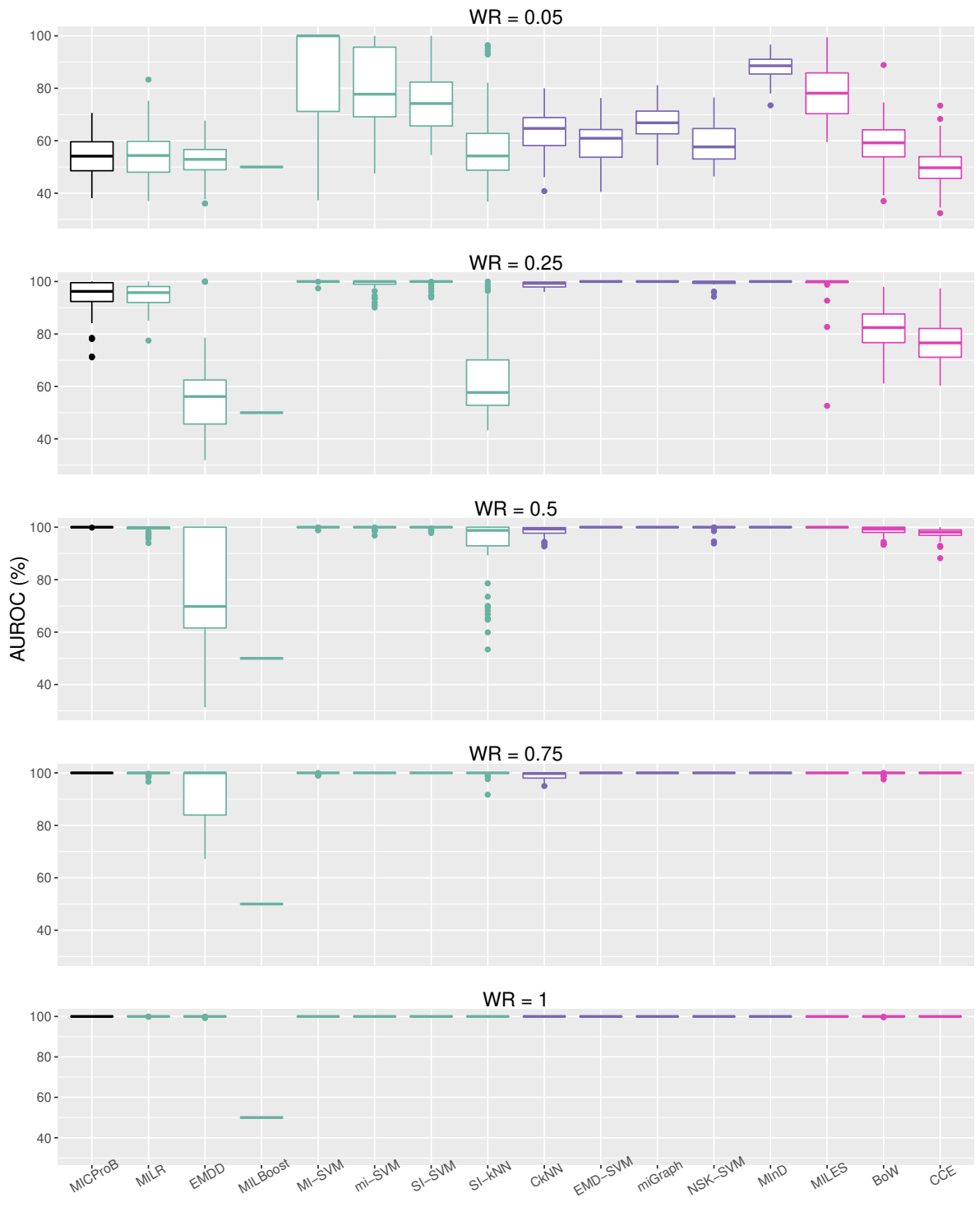


Figure B.4: Simulation evaluation under the WR framework: AUROC (%) for bag prediction by varying WR, evaluated on 50 replicates for robustness checking. Benchmark methods are distinguished by color (green: IS methods; purple: BS methods; magenta: ES methods).

B.1.2. Computational time

We compare the running time for MICProB and the 15 benchmark methods in Figure B.5 using the basic setting described in Section 3.2 of the main manuscript. Admittedly, MICProB is not as computationally efficient as most competing methods due to its sequential sampling procedure. However, regardless of the data-generating mechanism (PPI or WR), MICProB usually performs well, as opposed to all benchmark methods in bag classification and provides unique capacity of identifying primary instances. Furthermore, MICProB enables convenient statistical inference and transparent interpretation that optimization-based methods cannot provide.

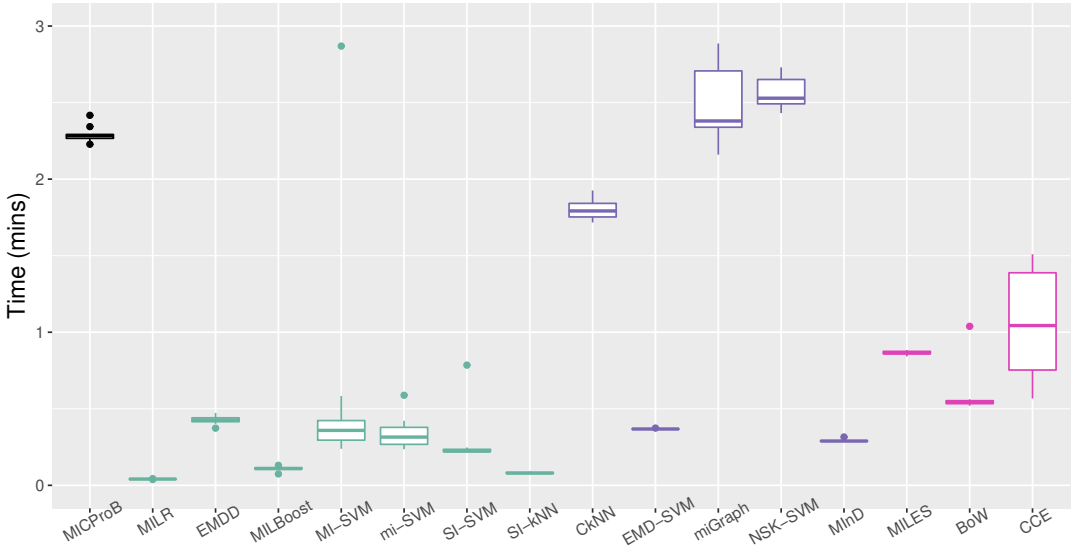


Figure B.5: Simulation evaluation under the PPI framework: computational time under the basic setting ($n = 300$, $m = 10$, $d = 30$, and $\overline{\text{PPI}} = 0.4$) for MICProB and 15 benchmark methods using boxplots (based on 10 replications). We run MICProB on a MacBook Pro with 2.4 GHz 8-Core Intel Core i9 processor and 16GB memory. MICProB iterates 50,000 samples and the chain is thinned by every 50 iterations. For MILR, R 3.6.1 on a partition node with 32 cores and 32GB memory of a computing cluster is used. MILR iterates 500 steps for the EM algorithm and no penalty is imposed. MATLAB 2020a on a partition node with 32 cores and 32GB memory of a computing cluster is used for the remaining 14 methods.

B.2. Additional results for TCGA data

B.2.1. Convergence diagnostics

We illustrate how we apply diagnostic techniques to detect the convergence of MICProB using the THYM dataset from TCGA data example in Section 4.1 of the main manuscript. We run five independent MCMC chains with randomly generated starting points. The trace plots and density plots of the linear predictor $x_{ij}\hat{b}$ for some randomly selected instances, are shown in Figure B.6 and Figure B.7. Clearly, the chains are well-mixed. In addition, we apply the Gelman-Rubin's convergence diagnostics to the linear predictor of these instances. The potential scale reduction factors (PSRF) are all very close to 1 (≤ 1.01). So is the multivariate PSRF. Collectively, these results provide sufficient evidence that the proposed model has properly converged.

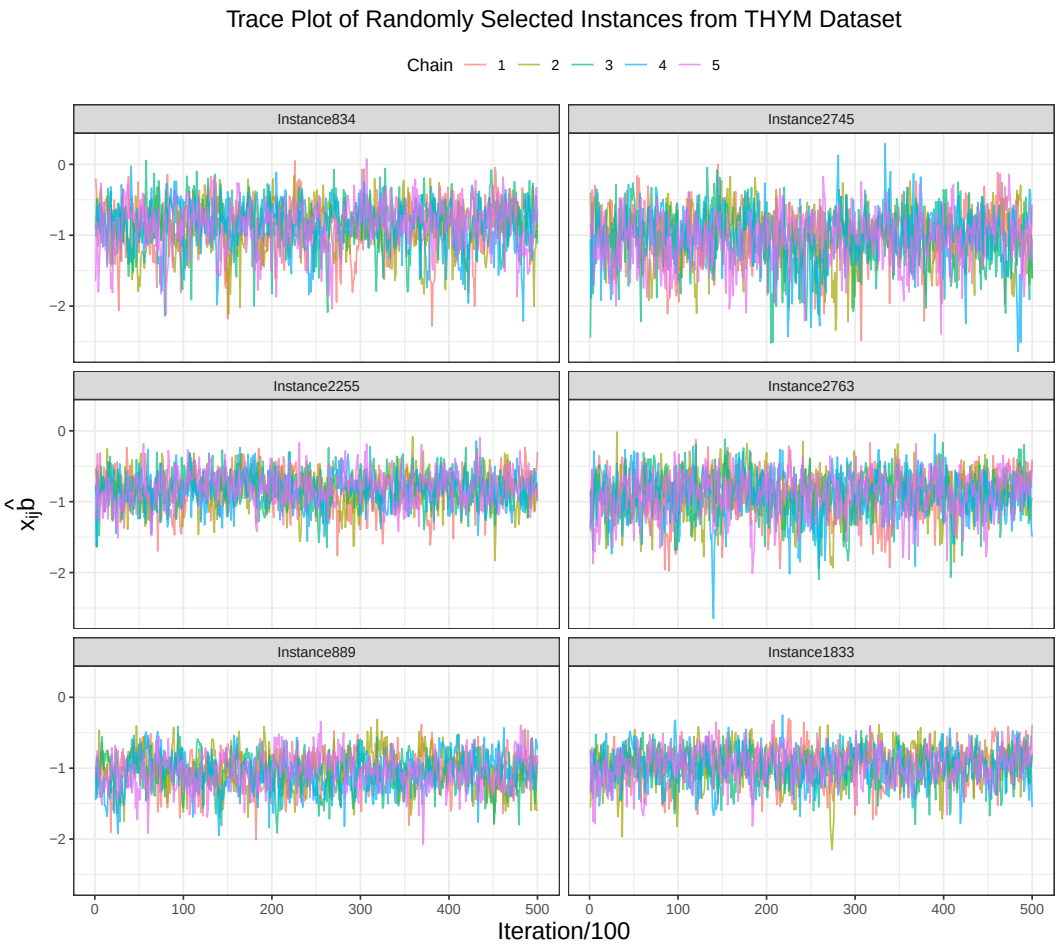


Figure B.6: TCGA data for THYM cancer: trace plots of linear predictor $x_{ij}\hat{b}$ for randomly selected instances. Five chains with randomly generated starting points are shown in different colors.

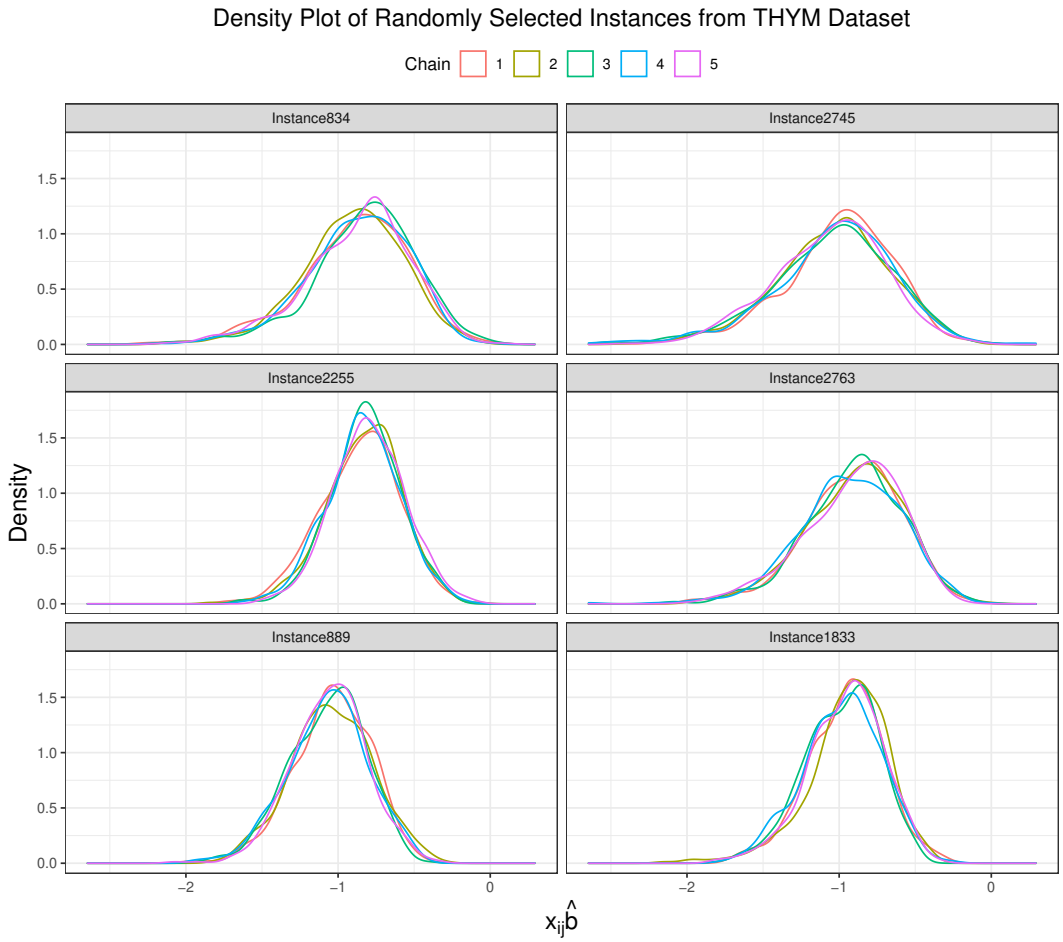


Figure B.7: TCGA data for THYM cancer: density curves of linear predictor $x_{ij}\hat{b}$ for randomly selected instances. Five chains with randomly generated starting points are shown in different colors.

B.2.2. Results from using marginally half-t priors on covariance matrices

We also experiment with placing marginally half-t priors on covariance matrices Σ_β and Σ_b, respectively, as mentioned in Section 5 of the main manuscript. The priors for regression coefficients β and b are specified as

- $\beta|\mu_\beta, \Sigma_\beta \sim \text{MVN}(\mu_\beta, \Sigma_\beta)$, $\Sigma_\beta|s_0, ..., s_d \sim \text{IW}(\nu + (d+1) - 1, 2\nu\text{diag}(1/s_0, ..., 1/s_d))$, $s_r \overset{ind}{\sim} \text{IG}(1/2, 1/S_r^2)$ for $r = 0, ..., d$.
- $b|\mu_b, \Sigma_b \sim \text{MVN}(\mu_b, \Sigma_b)$, $\Sigma_b|q_0, ..., q_d \sim \text{IW}(\nu + (d+1) - 1, 2\nu\text{diag}(1/q_0, ..., 1/q_d))$, $q_r \overset{ind}{\sim}$

IG$(1/2, 1/Q_r^2)$ for $r = 0, ..., d$.

Here, diag$(1/s_0, ..., 1/s_d)$ and diag$(1/q_0, ..., 1/q_d)$ are diagonal matrices with $1/s_0, ..., 1/s_d$ and $1/q_0, ..., 1/q_d$ on the diagonal, respectively; ν, S_r and Q_r for $r = 0, ..., d$ are positive scalars. The hyperparameters are $\mu_\beta, \mu_b, \nu, S_r, Q_r$ for $r = 0, ..., d$. We set $\mu_\beta = \mu_b = (0, 0, ..., 0)$, $\nu = 2$, and $S_r = Q_r = 10^5$. Let $\Omega = (\beta, b, \Delta, Z, U, \Sigma_\beta, \Sigma_b, \{s_r\}_{r=0}^d, \{q_r\}_{r=0}^d)$ denote the collection of all model parameters and latent variables involved. The full probability model is given by

$$\begin{aligned} p(y, \Omega|X) = & \, p(y|Z) \times p(Z|X, \Delta, \beta) \times p(\Delta|U) \times p(U|X, b) \\ & \times p(\beta|\mu_\beta, \Sigma_\beta) \times p(b|\mu_b, \Sigma_b) \\ & \times p(\Sigma_\beta|s_0, ..., s_d) \times p(\Sigma_b|q_0, ..., q_d) \\ & \times \prod_{r=0}^{d} p(s_r|S_r) \times \prod_{r=0}^{d} p(q_r|Q_r) \end{aligned} .$$

Since each parameter/latent variable has its posterior distribution in a closed form, the Gibbs sampler can be used to draw samples from the joint posterior distribution. We can update β, b, Δ, Z and U as described in Section 2.2 and the additional parameters as follows:

- $\Sigma_\beta|\cdots \sim$ IW$(\nu + 2(d+1) - 1, \beta\beta^T + 2\nu$diag$(1/s_0, ..., 1/s_d))$.
- $\Sigma_b|\cdots \sim$ IW$(\nu + 2(d+1) - 1, bb^T + 2\nu$diag$(1/q_0, ..., 1/q_d))$.
- $s_r|\cdots \overset{ind}{\sim}$ IG$(\frac{\nu+(d+1)}{2}, \nu(\Sigma_\beta^{-1})_{rr} + 1/S_r^2)$ for $r = 0, ..., d$, where $(\Sigma_\beta^{-1})_{rr}$ is the (r, r) entry of Σ_β^{-1}.
- $q_r|\cdots \overset{ind}{\sim}$ IG$(\frac{\nu+(d+1)}{2}, \nu(\Sigma_b^{-1})_{rr} + 1/Q_r^2)$ for $r = 0, ..., d$, where $(\Sigma_b^{-1})_{rr}$ is the (r, r) entry of Σ_b^{-1}.

Table B.1 shows that MICProB with original non-hierarchical priors on Σ_β and Σ_b results in higher average performance across seven cancer types. As for the performance on

individual cancers, the half-t priors only wins on OV and ESCA.

		Cancer type						
	Average	KIRC	SKCM	DLBC	ESCA	OV	THYM	STAD
Original	78.3	68.1 (2.5)	69.6 (2.2)	82.2 (5.1)	78.2 (3.7)	76.7 (3.0)	87.7 (2.2)	85.6 (1.5)
Half-t	71.8	59.5 (3.5)	60.1 (2.0)	69.0 (4.7)	79.3 (2.4)	81.0 (1.6)	76.6 (4.4)	77.2 (2.6)

Table B.1: TCGA data: Average AUROC (%) with standard error for predicting bag labels for MICProB with original prior specifications and marginally half-t priors on covariance matrices.

B.3. Additional results for neoantigen data

We describe neoantigen data used in Section 4.2 of the main manuscript using Figure B.8. Panel (a) shows the distribution of numbers of neoantigens from different patients, ranging from 1 to 664 with mean 113. Panel (b) shows distributions of the six covariates that are used to describe neoantigens. Except for mutation type (mut_type) which is binary, all other covariates are continuous.

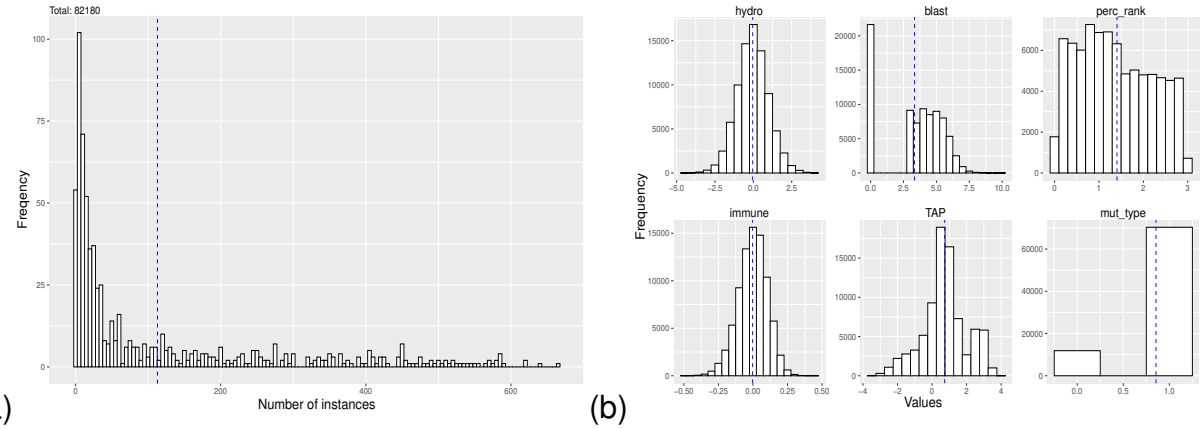


Figure B.8: Neoantigen data: (a) the distribution of numbers of instances (neoantigens) from different bags (patients); (b) distributions of the six neoantigen covariates. Blue dashed line indicates sample mean.

We provide posterior means and interval estimates of coefficients of neoantigen covariates from MICProB in Table B.2. As a statistical method, MILR also provides estimates

for regression coefficients and their standard errors. We use the default setting in the R package `milr` and show results in Table B.2 as well. First, the directions of the intercept and effects of blast, perc_rank, TAP, and mut_type (i.e., the signs of those regression estimates) from MICProB agree with those from MILR. Furthermore, the credible interval perc_rank is positive, which is consistent to the significant estimate from MILR. So does the intercept. For the remaining covariate effects, estimates from both methods are insignificant statistically.

		Intercept	hydro	blast	perc_rank	immune	TAP	mut_type
MILR	Estimate	-4.571	-0.015	0.038	2.479	0.017	-0.084	0.292
	SE	0.151	0.168	0.172	0.249	0.167	0.166	0.427
MICProB	Posterior Mean	-1.162	0.087	0.092	6.473	-0.192	-0.111	0.420
	2.5% Quantile.	-2.020	-0.546	-0.499	5.582	-0.734	-0.723	-0.453
	97.5% Quantile	-0.433	0.726	0.737	7.483	0.436	0.435	1.305

Table B.2: Neoantigen data: the top panel reports estimates of regression coefficients with standard errors from MILR; the bottom panel reports point and interval estimates of the coefficients from MICProB.

BIBLIOGRAPHY

[1] James H Albert and Siddhartha Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association*, 88(422):669–679, 1993.

[2] Robert A Amar, Daniel R Dooly, Sally A Goldman, and Qi Zhang. Multiple-instance learning of real-valued data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 3–10, 2001.

[3] Jaume Amores. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence*, 201:81–105, 2013.

[4] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems*, pages 577–584, 2003.

[5] Annabella Astorino, Antonio Fuduli, and Manlio Gaudioso. A Lagrangian relaxation approach for binary multiple instance classification. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9):2662–2671, 2019.

[6] William R Atchley, Jieping Zhao, Andrew D Fernandes, and Tanja Drüke. Solving the protein sequence metric problem. *Proceedings of the National Academy of Sciences*, 102(18):6395–6400, 2005.

[7] Boris Babenko, Piotr Dollár, Zhuowen Tu, and Serge Belongie. Simultaneous learning and alignment: Multi-instance and multi-pose learning. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, 2008.

[8] Charles Bergeron, Jed Zaretzki, Curt Breneman, and Kristin P Bennett. Multiple instance ranking. In *Proceedings of the 25th International Conference on Machine Learning*, pages 48–55, 2008.

[9] Daria Beshnova, Jianfeng Ye, Oreoluwa Onabolu, Benjamin Moon, Wenxin Zheng, Yang-Xin Fu, James Brugarolas, Jayanthi Lea, and Bo Li. De novo prediction of cancer-associated T cell receptors for noninvasive cancer detection. *Science Translational Medicine*, 12(557), 2020.

[10] Dmitriy A Bolotin, Mikhail Shugay, Ilgar Z Mamedov, Ekaterina V Putintseva, Maria A Turchaninova, Ivan V Zvyagin, Olga V Britanova, and Dmitriy M Chudakov. MiTCR: Software for T-cell receptor sequencing data analysis. *Nature Methods*, 10(9):813, 2013.

[11] Forrest Briggs, Balaji Lakshminarayanan, Lawrence Neal, Xiaoli Z Fern, Raviv Raich, Sarah JK Hadley, Adam S Hadley, and Matthew G Betts. Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach. *The Journal of the Acoustical Society of America*, 131(6):4640–4650, 2012.

[12] Lauren Averett Byers and Charles M Rudin. Small cell lung cancer: Where do we go from here? *Cancer*, 121(5):664–672, 2015.

[13] Marc-André Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77:329–353, 2018.

[14] Gavin C Cawley and Nicola LC Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *The Journal of Machine Learning Research*, 11:2079–2107, 2010.

[15] Ping-Yang Chen, Ching-Chuan Chen, Chun-Hao Yang, Sheng-Mao Chang, and Kuo-Jung Lee. milr: Multiple-instance logistic regression with lasso penalty. *The R Journal*, 9(1):446, 2017.

[16] Yixin Chen, Jinbo Bi, and James Ze Wang. Miles: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1931–1947, 2006.

[17] Veronika Cheplygina, David MJ Tax, and Marco Loog. Multiple instance learning with bag dissimilarities. *Pattern Recognition*, 48(1):264–275, 2015.

[18] Daniel L Clarke-Pearson. Screening for ovarian cancer. *New England Journal of Medicine*, 361(2):170–177, 2009.

[19] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

[20] James Foulds and Eibe Frank. A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, 25(1):1–25, 2010.

[21] Eibe Frank and Xin Xu. Applying propositional learning algorithms to multi-instance data. Technical report, Department of Computer Science, University of Waikato, 2003.

[22] Jerome H Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

[23] Thomas Gärtner, Peter A Flach, Adam Kowalczyk, and Alexander J Smola. Multi-instance kernels. In *Proceedings of the 19th International Conference on Machine Learning*, pages 179–186, 2002.

[24] Manlio Gaudioso, Giovanni Giallombardo, Giovanna Miglionico, and Eugenio Vocaturo. Classification in the multiple instance learning framework via spherical separation. *Soft Computing*, 24(7):5071–5077, 2020.

[25] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. Chapman Hall, London, 2013.

[26] Sergei I Grivennikov, Florian R Greten, and Michael Karin. Immunity, inflammation, and cancer. *Cell*, 140(6):883–899, 2010.

[27] Christopher Harris, Beth Croce, and Ashleigh Xie. Thymoma. *Annals of Cardiothoracic Surgery*, 4(6):576, 2015.

[28] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[30] Yang Hu, Mingjing Li, and Nenghai Yu. Multiple-instance ranking: learning to rank images for image retrieval. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[31] Alan Huang and Matthew P Wand. Simple marginally noninformative prior distributions for covariance matrices. *Bayesian Analysis*, 8(2):439–452, 2013.

[32] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2127–2136, 2018.

[33] Kota Iwahori, Sunitha Kakarla, Mireya P Velasquez, Feng Yu, Zongzhen Yi, Claudia Gerken, Xiao-Tong Song, and Stephen Gottschalk. Engager T cells: a new class of antigen-specific T cells that redirect bystander T cells. *Molecular Therapy*, 23(1):171–178, 2015.

[34] Ya-bin Jin, Wei Luo, Guo-yi Zhang, Kai-rong Lin, Jin-huan Cui, Xiang-ping Chen, Ying-ming Pan, Xiao-fan Mao, Jun Tang, and Yue-jian Wang. TCR repertoire profiling of tumors, adjacent normal tissues, and peripheral blood predicts survival in nasopharyngeal carcinoma. *Cancer Immunology, Immunotherapy*, 67(11):1719–1730, 2018.

[35] Michael H Kershaw, Jennifer A Westwood, and Phillip K Darcy. Gene-engineered T cells for cancer therapy. *Nature Reviews Cancer*, 13(8):525–541, 2013.

[36] Diether Lambrechts, Els Wauters, Bram Boeckx, Sara Aibar, David Nittner, Oliver Burton, Ayse Bassez, Herbert Decaluwé, Andreas Pircher, Kathleen Van den Eynde, et al. Phenotype molding of stromal cells in the lung tumor microenvironment. *Nature Medicine*, 24(8):1277–1289, 2018.

[37] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, pages 562–570, 2015.

[38] Bo Li, Taiwen Li, Binbin Wang, Ruoxu Dou, Jian Zhang, Jun S Liu, and X Shirley Liu. Ultrasensitive detection of TCR hypervariable-region sequences in solid-tissue RNA-seq data. *Nature Genetics*, 49(4):482–483, 2017.

[39] Jianfang Liu, Tara Lichtenberg, Katherine A Hoadley, Laila M Poisson, Alexander J Lazar, Andrew D Cherniack, Albert J Kovatich, Christopher C Benz, Douglas A Levine, Adrian V Lee, et al. An integrated TCGA pan-cancer clinical data resource to drive high-quality survival outcome analytics. *Cell*, 173(2):400–416, 2018.

[40] Tianshi Lu, Ze Zhang, James Zhu, Yunguan Wang, Peixin Jiang, Xue Xiao, Chantale Bernatchez, John V Heymach, Don L Gibbons, Jun Wang, et al. Deep learning-based prediction of the T cell receptor–antigen binding specificity. *Nature Machine Intelligence*, pages 1–12, 2021.

[41] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In *Advances in Neural Information Processing Systems*, pages 570–576, 1998.

[42] Diana Miao, Claire A Margolis, Wenhua Gao, Martin H Voss, Wei Li, Dylan J Martini, Craig Norton, Dominick Bossé, Stephanie M Wankowicz, Dana Cullen, et al. Genomic correlates of response to immune checkpoint therapies in clear cell renal cell carcinoma. *Science*, 359(6377):801–806, 2018.

[43] Cancer Genome Atlas Research Network. Comprehensive molecular characterization of clear cell renal cell carcinoma. *Nature*, 499(7456):43–49, 2013.

[44] Michael A Newton, Amine Noueiry, Deepayan Sarkar, and Paul Ahlquist. Detecting differential gene expression with a semiparametric hierarchical mixture method. *Biostatistics*, 5(2):155–176, 2004.

[45] Sachiko Okamoto, Junichi Mineno, Hiroaki Ikeda, Hiroshi Fujiwara, Masaki Yasukawa, Hiroshi Shiku, and Ikunoshin Kato. Improved expression and reactivity of transduced tumor-specific TCRs in human lymphocytes by specific silencing of endogenous TCR. *Cancer Research*, 69(23):9003–9011, 2009.

[46] Jared Ostmeyer, Scott Christley, Inimary T Toby, and Lindsay G Cowell. Biophysicochemical motifs in T-cell receptor sequences distinguish repertoires from tumor-infiltrating lymphocyte and adjacent healthy tissue. *Cancer Research*, 79(7):1671–1680, 2019.

[47] Drew Pardoll. Does the immune system see tumors as foreign or self? *Annual Review of Immunology*, 21(1):807–839, 2003.

[48] Seongoh Park, Xinlei Wang, Johan Lim, Guanghua Xiao, Tianshi Lu, and Tao Wang. Bayesian multiple instance regression for modeling immunogenic neoantigens. *Statistical Methods in Medical Research*, 29(10):3032–3047, 2020.

[49] Deepak Pathak, Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional multi-class multiple instance learning. *arXiv preprint arXiv:1412.7144v4*, 2014.

[50] Nicholas G Polson, James G Scott, and Jesse Windle. Bayesian inference for logistic models using Pólya–Gamma latent variables. *Journal of the American statistical Association*, 108(504):1339–1349, 2013.

[51] Gwenolé Quellec, Guy Cazuguel, Béatrice Cochener, and Mathieu Lamard. Multiple-instance learning for medical image and video analysis. *IEEE Reviews in Biomedical Engineering*, 10:213–234, 2017.

[52] Colin Raffel and Daniel PW Ellis. Feed-forward networks with attention can solve some long-term memory problems. *arXiv preprint arXiv:1512.08756*, 2015.

[53] David H Raulet and Nadia Guerra. Oncogenic stress sensed by the immune system: role of natural killer cell receptors. *Nature Reviews Immunology*, 9(8):568–580, 2009.

[54] Soumya Ray and Mark Craven. Supervised versus multiple instance learning: An empirical comparison. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 697–704, 2005.

[55] Soumya Ray and David Page. Multiple instance regression. In *Proceedings of the 18th Internatinal Conference on Machine Learning*, pages 425–432, 2001.

[56] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[57] Markus G Rudolph, Robyn L Stanfield, and Ian A Wilson. How TCRs bind MHCs, peptides, and coreceptors. *Annual Review of Immunology*, 24:419–466, 2006.

[58] Yusuke Sato, Tetsuichi Yoshizato, Yuichi Shiraishi, Shigekatsu Maekawa, Yusuke Okuno, Takumi Kamura, Teppei Shimamura, Aiko Sato-Otsubo, Genta Nagae, Hiromichi Suzuki, et al. Integrated molecular analysis of clear-cell renal cell carcinoma. *Nature Genetics*, 45(8):860–867, 2013.

[59] Aatur D Singhi, Eugene J Koay, Suresh T Chari, and Anirban Maitra. Early detection of pancreatic cancer: opportunities and challenges. *Gastroenterology*, 156(7):2024–2040, 2019.

[60] Reiji Teramoto and Hisashi Kashima. Prediction of protein-ligand binding affinities using multiple instance learning. *Journal of Molecular Graphics and Modelling*, 29(3):492–497, 2010.

[61] Eugenio Vocaturo and Ester Zumpano. Multiple instance learning approaches for melanoma and dysplastic nevi images classification. In *IEEE International Conference on Machine Learning and Applications*, pages 1396–1401, 2020.

[62] Eugenio Vocaturo, Ester Zumpano, Giovanni Giallombardo, and Giovanna Miglionico. 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 and Applications*, pages 1–9, 2020.

[63] Jun Wang and Jean-Daniel Zucker. Solving multiple-instance problem: A lazy learning approach. In *Proceedings of the 17th International Conference on Machine Learning*, pages 1119–1126, 2000.

[64] Tao Wang, Rong Lu, Payal Kapur, Bijay S Jaiswal, Raquibul Hannan, Ze Zhang, Ivan Pedrosa, Jason J Luke, He Zhang, Leonard D Goldstein, et al. An empirical approach leveraging tumorgrafts to dissect the tumor microenvironment in renal cell carcinoma identifies missing link to prognostic inflammatory factors. *Cancer Discovery*, 8(9):1142–1155, 2018.

[65] Xiaofang Wang, Hui Xiong, Daning Liang, Zhenzhen Chen, Xiqing Li, and Kun Zhang. The role of SRGN in the survival and immune infiltrates of skin cutaneous melanoma (SKCM) and SKCM-metastasis patients. *BMC Cancer*, 20:1–8, 2020.

[66] Xinggang Wang, Yongluan Yan, Peng Tang, Xiang Bai, and Wenyu Liu. Revisiting multiple instance neural networks. *Pattern Recognition*, 74:15–24, 2018.

[67] Zhuang Wang, Vladan Radosavljevic, Bo Han, Zoran Obradovic, and Slobodan Vucetic. Aerosol optical depth prediction from satellite observations by multiple instance regression. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 165–176, 2008.

[68] Sarah K Whiteside, Jeremy P Snook, Matthew A Williams, and Janis J Weis. Bystander T cells: a balancing act of friends and foes. *Trends in Immunology*, 39(12):1021–1035, 2018.

[69] Danyi Xiong, Ze Zhang, Tao Wang, and Xinlei Wang. A comparative study of multiple instance learning methods for cancer detection using T-cell receptor sequences. *Computational and Structural Biotechnology Journal*, 19:3255, 2021.

[70] Zheng-Jun Zha, Xian-Sheng Hua, Tao Mei, dong Wang, Guo-Jun Qi, and Zengfu Wang. Joint multi-label multi-instance learning for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[71] Dan Zhang, Fei Wang, Luo Si, and Tao Li. Maximum margin multiple instance clustering with applications to image and text clustering. *IEEE Transactions on Neural Networks*, 22(5):739–751, 2011.

[72] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007.

[73] Min-Ling Zhang and Zhi-Hua Zhou. Multi-instance clustering with applications to multi-instance prediction. *Applied Intelligence*, 31(1):47–68, 2009.

[74] Qi Zhang and Sally A Goldman. EM-DD: An improved multiple-instance learning technique. In *Advances in Neural Information Processing Systems*, pages 1073–1080, 2002.

[75] Ze Zhang, Danyi Xiong, Xinlei Wang, Hongyu Liu, and Tao Wang. Mapping the functional landscape of T cell receptor repertoires by single-T cell transcriptomics. *Nature Methods*, 18(1):92–99, 2021.

[76] Zhi-Li Zhang and Min-Ling Zhang. Multi-instance multi-label learning with application to scene classification. In *Advances in Neural Information Processing Systems*, pages 1609–1616, 2007.

[77] Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. Multi-instance learning by treating instances as non-iid samples. In *Proceedings of the 26th International Conference on Machine Learning*, pages 1249–1256, 2009.

[78] Zhi-Hua Zhou and Jun-Ming Xu. On the relation between multi-instance learning and semi-supervised learning. In *Proceedings of the 24th international conference on Machine learning*, pages 1167–1174, 2007.

[79] Zhi-Hua Zhou and Min-Ling Zhang. Solving multi-instance problems with classifier ensemble based on constructive clustering. *Knowledge and Information Systems*, 11(2):155–170, 2007.

[80] Zhi-Hua Zhou, Min-Ling Zhang, Sheng-Jun Huang, and Yu-Feng Li. Multi-instance multi-label learning. *Artificial Intelligence*, 176(1):2291–2320, 2012.

[81] Ji Zhu, Saharon Rosset, Robert Tibshirani, and Trevor J Hastie. 1-norm support vector machines. In *Advances in Neural Information Processing Systems*, pages 49–56, 2004.