

Gene Set-Based Classification Models for Cancer Biology

by

Arezou Rahimi

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of
Doctor of Philosophy

in

Industrial Engineering and Operations Management



KOÇ ÜNİVERSİTESİ

August 5, 2020

Gene Set-Based Classification Models for Cancer Biology

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a doctoral dissertation by

Arezou Rahimi

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Assoc. Prof. Mehmet Gönen (Advisor)

Prof. Ceyda Oğuz

Prof. Füsun Can

Assoc. Prof. Nihal Ata Tutkun

Assist. Prof. Öznur Taştan

Date: _____



*To my dad: my hero,
To my mom: my angel,
And to my lovely brother.*

ABSTRACT

Gene Set-Based Classification Models for Cancer Biology

Arezou Rahimi

**Doctor of Philosophy in Industrial Engineering and Operations
Management**

August 5, 2020

As one of most prevalent and fatal diseases worldwide, cancer has been the focus of biomedical research for many decades. A wide range of different cancers with various causes have been identified and studied, and many treatment methods and drugs have been developed for these cancers. However, many open questions remain in understanding the mechanisms of cancer. With the advent of the high throughput sequencing technologies and the ever-increasing availability of genomic information gathered from cancer patients, researchers have successfully employed genomic information in diagnosis, prognosis and treatment of cancers. Still, given the large number of genes, their considerable correlation and the heterogeneity of cancers, genomic information alone often falls short of generating interpretable models for cancers. To alleviate these effects, pathways can be incorporated into models of cancer. In this thesis, using pathways, we have developed classification methods that produce interpretable models for progression of cancers and survival outcome.

Patients usually experience the growth of a number of tumors in their life-time. However, not all these tumors pose a danger to the patient's life. To communicate the severity of the cancer, practitioners and researchers use the staging convention. The stage of a cancer encodes the level of its spread in the neighboring tissue and the body in general, where early-stages are assigned to local tumors and late-stages to the cancers that have spread in the body. Hence, understanding what drives cancers from early- to late-stages is an important question in understanding the mechanisms of cancer. The survival outlook of a patient is a similar measure for the severity of a cancer and understanding the mechanisms that affect the survival chances of a patient is immensely important in devising treatment strategies.

In this work, we use multiple kernel learning for integrating pathways into the

classification models for cancers. This allows for developing models that are more accurate and far more interpretable compared to models generated by conventional methods that do not use the pathway information. We then extend this method in several directions to improve accuracy and interpretability of the models. In the second part of this thesis, observing that the level of sparsity of the solutions can differ largely from cancer to cancer and is often only indirectly affected by the parameters of the methods, we develop a model with an adjustable measure of sparsity and propose efficient solution methods for solving instances of this problem. In the third part of this thesis, given the known similarities between cancers, we develop a framework for building multitask classification models to improve the classification accuracy of cohorts with limited data using the similar cohorts with abundant data. Finally, we employ optimization techniques such as the cutting-plane method and Benders decomposition to improve the algorithmic performance of these methods to the point that they can be applied successfully to large-scale problems stemming from considering several cancers together in a multitask framework.

The practical application of these methods is examined by applying them to the two aforementioned classification tasks across the Cancer Genome Atlas datasets for 27 cancer types. The results of these experiments indicate that the incorporation of pathways into these classification problems facilitates generating more interpretable and more accurate models, the similarity of cancers can be leveraged using a multitask framework towards the same goals, and the optimization methods developed here allow for solving these large-scale problems in efficient time using parallelization technologies.

ÖZETÇE

Kanser Biyolojisi için Gen Kümesi Tabanlı Sınıflandırma Modelleri

Arezou Rahimi

Endüstri Mühendisliği ve İşletme Yönetimi, Doktora

5 Ağustos 2020

Dünya çapında en yaygın ve ölümcül hastalıklardan biri olan kanser, on yıllardır biyomedikal araştırmaların odak noktası olmuştur. Çeşitli nedenler dolayısıyla ortaya çıkan birçok farklı kanser tanımlanmış, incelenmiş, bu kanserler için birçok tedavi yöntemi ve ilaç geliştirilmiştir. Bununla birlikte, kanser mekanizmalarını anlama konusunda birçok çözümlenmemiş soru bulunmaktadır. Yüksek verimli dizileme teknolojilerinin ortaya çıkması ve kanser hastalarından toplanan genomik bilgilerin sürekli artan kullanılabilirliği ile araştırmacılar, kanserlerin teşhisi, prognozu ve tedavisinde genomik bilgiyi başarıyla kullanmışlardır. Yine de çok sayıda gen, bunların arasındaki yüksek korelasyon ve kanserlerin heterojenliği düşünüldüğünde tek başına genomik bilgi, genellikle kanserler için yorumlanabilir modeller üretmekte yetersiz kalmaktadır. Bu etkileri hafifletmek için, yolaklar, kanser modellerine dahil edilebilir. Bu tezde, yolakları kullanarak kanserlerin ilerlemesi ve hayatta kalma sonuçları için yorumlanabilir modeller üreten sınıflandırma yöntemleri geliştirdik.

Hastaların vücutlarında yaşamları boyunca bir dizi tümör oluşur ve büyür. Ama tümörlerin hepsi hastanın hayatı için tehlike oluşturmaz. Uygulayıcılar ve araştırmacılar kanserin ciddiyetini ifade etmek için evreleme kuralını kullanır. Kanser bu bulunduğu evre, komşu doku ve genel olarak vücuttaki yayılma seviyesini belirler; erken evreler lokal tümörleri, geç evreler ise vücuda yayılım göstermiş kanserleri kapsar. Bu nedenle, kanserleri erken aşamalardan ileri aşamalara neyin yönlendirdiğini anlamak, kanserin mekanizmalarını anlamada önemli bir sorudur. Benzer şekilde bir hastanın sağkalım profili, kanserin şiddetini ölçümlendirebilmek için kullanılabilir ve hastanın hayatta kalma şansını etkileyen mekanizmaları anlamak tedavi stratejileri geliştirmede son derece önemlidir.

Bu çalışmada, kanserler için sınıflandırma modellerine yolakları entegre etmek için çoklu çekirdek öğrenimini kullanıyoruz. Bu yöntem, yolak bilgilerini kullan-

mayan geleneksel yöntemlerle üretilen modellere kıyasla daha doğru ve daha yorumlanabilir modeller geliştirilmesine olanak tanır. Daha sonra modellerin doğruluğunu ve yorumlanabilirliğini geliştirmek için bu yöntemi çeşitli yönlemlere genişletiyoruz. Bu tezin ikinci bölümünde, çözümlerin seyreklik seviyesinin büyük ölçüde kanserden kansere farklılık gösterebileceğini ve genellikle yöntemlerin parametrelerinden sadece dolaylı olarak etkilendiğini gözlemleyerek, ayarlanabilir bir seyreklik ölçüsü olan bir model geliştiriyor ve bu problemin örneklerini çözmek için hızlı çözüm yöntemleri öneriyoruz. Tezin üçüncü bölümünde, kanserler arasında bilinen benzerlikler göz önüne alındığında, çok sayıda veriye sahip benzer kohortları kullanarak sınırlı veriye sahip kohortların sınıflandırma doğruluğunu iyileştirmek ve çok görevli sınıflandırma modelleri oluşturmak için bir çerçeve geliştirdik. Son olarak, bu yöntemlerin algoritmik performansını çok görevli ve bir çok kanserde birlikte düşünmekten kaynaklanan büyük ölçekli problemlere başarılı bir şekilde uygulanabilecekleri noktaya getirmek için kesin düzlem yöntemi ve Benders ayrıştırması gibi eniyileme tekniklerini kullanıyoruz.

Bu yöntemlerin pratik uygulamasını, 27 kanser türü için Kanseri Genom Atlas veri kümelerinde yukarıda belirtilen iki sınıflandırma problemine uygulayarak inceledik. Bu deneylerin sonuçları, bu sınıflandırma problemlerine yolların dahil edilmesinin daha yorumlanabilir ve daha doğru modeller üretmeyi kolaylaştırdığını, kanserlerin benzerliğinin aynı amaçlara yönelik çok görevli bir çerçeve kullanılarak geliştirilebildiğini ve burada geliştirilen eniyileme yöntemlerinin paralelleştirme teknolojilerini kullanarak etkin zamanda büyük ölçekli problemlerin çözümüne olanak verdiğini göstermiştir.

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to my advisor, Assoc. Prof. Mehmet Gönen. I will forever be thankful for his support, patience, motivation, and guidance throughout my PhD, without whom this path would have been impossible to take. It is not often that one finds an advisor who is always available for support and listening to problems, little and big, professional and personal. His technical and personal advice has always been a source of inspiration for me, and doing research on a problem with a tangible impact under his guidance has been a joy and an honor. Besides my advisor, I would also like to thank the rest of my thesis committee: Prof. Ceyda Oğuz, Prof. Füsün Can, Assoc. Prof. Nihal Ata Tutkun, and Assist. Prof. Öznur Taştan, for their constructive comments and insights.

This journey brought many precious friendships to me. It is impossible to mention everyone, but, in particular, I express my warmest gratitude to my friend Siamak, who has always unconditionally and patiently supported me. I would like to extend my thanks to my friends Masoud and Maryam for their motivation and spirit, and Mojtaba for his endless friendship. I feel extremely fortunate for having their friendship. I also thank my office mates at Koç and fellow labmates at the Machine Intelligence and Data Analysis in Science Lab (MidasLab) for their kind support.

In addition, I would like to express my sincere love and appreciation to my dad, who taught me how to stand on my own feet and has supported me spiritually throughout my life, my mom, who taught me love and passion and gave color to my life, and my brother Amir, who has always been there for me.

Finally, I want to thank the Scientific and Technological Research Council of Turkey (TÜBİTAK, EEEAG 117E181), and Koç University for their financial support during my study.

TABLE OF CONTENTS

List of Tables	xiii
List of Figures	xiv
Abbreviations	xviii
Chapter 1: Introduction	1
1.1 Problem Definition	6
1.1.1 Discriminating Early- and Late-Stage Cancers	6
1.1.2 Two-Year Survival Prediction	6
1.2 Existing Approaches	6
1.2.1 Application of Machine Learning in Biomedical Research	7
1.2.2 Study of Progression of Cancer Using Genomic Data	7
1.2.3 Biochemical Pathways	8
1.2.4 Multiple Kernel Learning	9
1.2.5 Multitask Learning	10
1.2.6 Sparsity and Optimization Frameworks	11
Chapter 2: Background	13
2.1 Datasets	13
2.1.1 RNA-Seq Measurements	13
2.1.2 Pathological Stage Information	13
2.1.3 Dataset Construction	15
2.2 Gene Set Databases	17
2.2.1 Hallmark Collection	17
2.2.2 PID Collection	18

2.3	Methods	20
2.3.1	Random Forest	20
2.3.2	Support Vector Machine	21
Chapter 3:	Multiple Kernel Learning on Gene Sets	23
3.1	Multiple Kernel Learning	24
3.2	Group Lasso Multiple Kernel Learning	27
3.3	Gene Set-Based Kernel Functions	27
3.4	Discussion	29
Chapter 4:	Sparse Multiple Kernel Learning with Optimal Kernel Selection	31
4.1	Sparse Multiple Kernel Learning	32
4.1.1	Sparse Multiple Kernel Learning via Binary Kernel Selection	33
4.1.2	Sparse Multiple Kernel Learning via Penalized Kernel Selection	35
4.2	Solution Methods	36
4.2.1	Solving the Binary MKL Problem	36
4.2.2	A Three-Phase Algorithm for Solving the Binary MKL Problem	37
4.2.3	An Efficient Algorithm for Solving the Penalized MKL Problem	39
4.3	Discussion	39
Chapter 5:	Multitask Multiple Kernel Learning	41
5.1	Problem Definition	42
5.2	Co-clustering Step	43
5.3	Multitask Learning Step	46
5.4	Discussion	50
Chapter 6:	Efficient Multitask Multiple Kernel Learning	54
6.1	Introduction	54
6.2	Problem Definition	57

6.3	Linearization Models	59
6.3.1	Variable Linearization	59
6.3.2	Objective Function Linearization	61
6.4	Forest Formulation	66
6.4.1	Symmetry Breaking	68
6.4.2	Cutting-plane Algorithm for Forest Formulation	69
6.5	BDForest: Benders Decomposition for Forest Formulation	69
6.5.1	Formulation Strengthening	74
6.5.2	Pareto-optimal cuts	79
6.5.3	Valid Core Points	79
6.5.4	BDForest Algorithm	81
6.6	Discussion	81
Chapter 7:	Experimental Results	83
7.1	Experimental Settings	83
7.2	Multiple Kernel Learning on Gene Sets	85
7.2.1	Predictive Performance	85
7.2.2	Biological Mechanisms	87
7.3	Sparse Multiple Kernel Learning	91
7.3.1	Predictive Performance	91
7.3.2	Biological Mechanisms	93
7.3.3	Sparsity Analysis	99
7.4	Multitask Multiple Kernel Learning	101
7.4.1	Predictive Performance	102
7.4.2	Biological Mechanisms	104
7.4.3	Computational Complexity	106
7.5	Efficient Multitask Multiple Kernel Learning	106
7.5.1	Computational Complexity	107
7.5.2	Results on the TCGA Datasets	109

Chapter 8: Conclusions	114
Appendix A: Supplementary Materials for Multiple Kernel Learning on Gene Sets	125
A.1 Supplementary Numerical Results For Early- and Late-Stage Discrimination (E2)	125
Appendix B: Supplementary Materials for Sparse Multiple Kernel Learning	127
B.1 Supplementary Numerical Results For Early- and Late-Stage Discrimination (E1)	127
B.2 Supplementary Numerical Results For Early- and Late-Stage Discrimination (E2)	129
B.3 Supplementary Numerical Results for Two-Year Survival Analysis (S1)	135
Appendix C: Supplementary Materials for Multitask Multiple Kernel Learning	137
C.1 Computational Complexity Analysis	137
C.2 Sensitivity to Hyper-parameters	139
C.3 Supplementary Numerical Results For Early- and Late-Stage Discrimination (E2)	140

LIST OF TABLES

2.1	Summary of 20 TCGA cohorts that we used for discriminating early- and late-stage cancers (E1 and E2).	16
2.2	Summary of 20 TCGA cohorts that we used for the two-year survival analysis experiment (S1).	17
7.1	Average number of genes used by SpMKL and other classification models over 100 replications across 15 cancer datasets we used in E1.	100
7.2	Average number of genes used by SpMKL and other classification models over 100 replications across 20 cancer datasets we used in S1.	101
B.1	Mean AUROC values averaged over 100 replications of classification algorithms across 15 cancer datasets we used in E1.	127
B.2	Mean AUROC values averaged over 100 replications of classification algorithms across 18 cancer datasets we used in E2.	129
B.3	Average number of genes used by each classification model over 100 replications across 18 cancer datasets we used in E2.	130
B.4	Mean AUROC values averaged over 100 replications of classification algorithms across 20 cancer datasets we used in S1.	135

LIST OF FIGURES

1.1	Incorporation of pathways to the classification models for cancer using kernel functions	4
2.1	Names and sizes of 50 pathways contained in the Hallmark collection.	18
2.2	Names and sizes of 196 pathways contained in the PID collection. . .	19
3.1	Overview of multiple kernel learning on gene sets.	28
4.1	Flowchart of the different phases in the three-phase algorithm. . . .	37
5.1	Overview of the proposed multitask multiple kernel learning algorithm.	44
5.2	Similarity matrix obtained by solving 100 replications of the MTMKL problem with a co-clustering step on 15 TCGA cohorts.	51
6.1	A toy example for clustering the tasks as a collection of trees.	67
6.2	Symmetry breaking constraints reduce the number of multiple solutions. The three distinct solutions given in (a)–(c) refer to the same cluster; however, only (c) satisfies the symmetry breaking constraints.	69
6.3	Flow of information between different modules of BDForest algorithm.	82
7.1	Overview of the evaluation framework we developed for predicting pathological stages of primary tumors and survival outcome prediction of patients from their gene expression profiles.	84
7.2	Predictive performances of RF , SVM and MKL [H] on 15 cancer datasets in E1 constructed from TCGA cohorts.	86

7.3	Selection frequencies of 50 gene sets in the Hallmark collection as identified by MKL[H] for 15 cancer datasets in E1.	89
7.4	Predictive performances of SpMKL[P], SpMKL[H] and other classification algorithms on 15 cancer datasets in E1 constructed from TCGA cohorts.	95
7.5	Predictive performances of SpMKL[P], SpMKL[H] and other classification algorithms on 20 cancer datasets in S1 constructed from TCGA cohorts.	96
7.6	Selection frequencies of 50 gene sets in the Hallmark collection as identified by SpMKL[H] for 15 cancer datasets in E1.	97
7.7	Selection frequencies of 50 gene sets in the Hallmark collection as identified by SpMKL[H] for 20 cancer datasets in S1.	98
7.8	Sparsity comparison of SpMKL[P] and SpMKL[H] with other MKL algorithms on 15 cancer datasets in E1.	99
7.9	Sparsity comparison of SpMKL[P] and SpMKL[H] with other MKL algorithm on 20 cancer datasets in S1.	100
7.10	Predictive performances of RF, SVM, MKL[H], and MTMKL[H] on 15 cancer datasets constructed from TCGA in E1.	103
7.11	Selection frequencies of 50 gene sets in the Hallmark collection identified by MTMKL[H] for the 15 cancer datasets in experiment E1. . . .	105
7.12	Performance of two versions of Algorithm 6 (OLJ1 and OLJ2) under different values of T , K , and C	108
7.13	Four versions of Algorithm 9 (BDForest) across different values of T and K	109
7.14	Performance of Algorithms 5 (VL), 6 (OLJ1) and 9 (BDFS-P0) across different number of tasks and clusters.	110
7.15	Grouping frequencies between 15 TCGA cohorts over 100 replications of the MTMKLC problem.	111

7.16	Predictive performances of MTMKL [H], BDF [H], and other classification methods on 15 datasets constructed from TCGA in E1.	113
A.1	Predictive performances of RF, SVM and MKL [H] on 18 cancer datasets in E2 constructed from TCGA cohorts.	125
A.2	Selection frequencies of 50 gene sets in the Hallmark collection identified by MKL [H] for 18 cancer datasets in E2.	126
B.1	Selection frequencies of 196 pathways in the PID collection as identified by SpMKL [P] for 15 cancer datasets in E1. For better visualization, only pathways with a minimum frequency of 60 are shown. . . .	128
B.2	Predictive performances of SpMKL [H], SpMKL [P], and other classification algorithms on 18 cancer datasets constructed for E2.	131
B.3	Sparsity comparison of SpMKL [H], SpMKL [P], and other MKL algorithms on 18 cancer datasets constructed for E2.	132
B.4	Selection frequencies of 50 gene sets in the Hallmark collection as identified by SpMKL [H] for 18 cancer datasets in E2.	133
B.5	Selection frequencies of 196 pathways in the PID collection as identified by SpMKL [P] for 18 cancer datasets in E2.	134
B.6	Selection frequencies of 196 pathways in the PID collection as identified by SpMKL [P] for 20 cancer datasets in S1.	136
C.1	Cross-validation, test and total time of RF, SVM, MKL [H], and MTMKL [H] for 15 cancer datasets constructed from TCGA in E1 over 100 replications.	138
C.2	Sensitivity of average cross-validation accuracy to different values of hyper parameters C , λ_1 and λ_2 for 15 cancer datasets in E1.	139
C.3	Predictive performances of RF, SVM, MKL [H], and MTMKL [H] on 18 cancer datasets constructed from TCGA in E2.	141

C.4	Selection frequencies of 50 gene sets in the Hallmark collection as identified by MTMKL[H] for 18 cancer datasets in E2	142
-----	---	-----



ABBREVIATIONS

AUROC	Area Under the Receiver Operating Characteristic Curve
BD	Benders Decomposition
BDForest	Benders Decomposition for Forest Formulation
DTC	Decision Tree Classifier
FMP	Forest Formulation Master Problem
HT	High-Throughput
MILP	Mixed Integer Linear Programming
MKL	Multiple Kernel Learning
MP	Master Problem
MSigDB	Molecular Signatures Database
MTMKL	Multitask Multiple Kernel Learning
MTMKLC	Multitask Multiple Kernel Learning with Clustering
OLMP	Objective Function Linearization Master Problem
PID	Pathway Interaction Database
PSP/DSP	Primal Subproblem / Dual Subproblem
RBMP	Relaxed Benders Master Problem
RF	Random Forest
RNA/DNA	Ribonucleic Acid / Deoxyribonucleic Acid
SIMILP	Semi-Infinite Mixed Integer Linear Program
SpMKL	Sparse Multiple Kernel Learning
SVM	Support Vector Machine
TCGA	The Cancer Genome Atlas
VLMP	Variable Linearization Master Problem

Chapter 1

INTRODUCTION

Cancer is one of the most prevalent and fatal diseases worldwide. Cancer is the cause of a sixth of deaths, and kills more than 10 million people annually. Given its immense impact, cancer has been the focus of researchers for many decades. More than 100 types of cancer have been identified, many causes for cancers have been discovered and a variety of treatment strategies with varying degree of success have been developed. Despite all these efforts, because of the heterogeneity of cancer, its mechanisms have proven hard to determine. Specifically, the practical impact of early treatment highlights the importance of understanding the mechanisms of the progression of cancers. Cancer survival has increased by advances in early detection of cancer that leads to improved treatment. In other words, an earlier stage of the disease at the moment of diagnosis implies higher chances of the patient surviving more than five years. For instance, colon cancer in early stage has a 91% five-year survival rate, and, for some cancers such as breast and prostate cancers, this rate increases to 98% and 100%, respectively. To better understand the progression of cancers, a number of processes have been identified, e.g., the generation of new blood vessels inside tumors and the breakdown of the self-regulated cell death process. However, evaluating the importance of the various known processes across numerous cancer types remains a challenge.

Because of the difficulties in studying complex biomedical phenomena related to cancer, machine learning has been employed in many aspects of cancer research, ranging from early-diagnosis efforts to predicting the effect of various drugs on the tumors. One of the areas where machine learning has been very successful has been the incorporation of genomic information into the different aspects of cancer

research. This has been accelerated by the recent advances in the sequencing technology, granting researchers access to large datasets with a variety of information sources including epi-genetic information and spacial information along with the genomic information. One of the challenges in using machine learning in biomedical research is that the solutions produced by the most accurate and prominent machine learning tools, such as random forests and neural networks, are not easy-to-interpret. Moreover, the use of genomic data is specifically prone to producing very accurate but hard-to-interpret models. This is because there is a very large number of genes, and their activity is often correlated.

In this setting, the identification and study of biochemical pathways creates an opportunity for understanding the mechanisms of cancer. Pathways are series of biochemical interactions that govern the production of biochemical material in the organisms. Changes to the genome resulting from mutations affect the normal function of the pathways which can increase the danger a cancer poses to the patient. These pathways and their relation to various cancer cohorts have been frequently studied individually in the literature. However, a pan-cancer comprehensive study of the role of pathways in progression of cancers and the survival chances of patients remains a gap in the literature that this work aims to fill.

In this work, we aim to answer the following research questions.

- How can pathways be incorporated into classification models for various cancers?
- Can the incorporation of pathways make the models more interpretable and more accurate?
- Can a desired measure of interpretability be forced directly on the models as opposed to indirectly controlling interpretability through other aspects of the model?
- Can information be shared between the models built for various cancers? If so, how can we determine the relevance of the cancers to each other?

- Given the large number of cancers, their heterogeneity and the large number of genes, how can optimization frameworks be employed to answer the aforementioned questions given the practical computational time constraints?

To address the first question we employ the family of multiple kernel learning methods (MKL). In multiple kernel learning, the prediction for a new data point depends on the label of the data points close to it. However, the measure for closeness (the kernel function) is not given a priori, and is constructed as the combination of predefined kernel functions in the process of learning. For incorporating pathways into classification models for cancer, a kernel function is built on the basis of each pathway. More precisely, each kernel function measures the similarity of two tumors only considering the genes involved with that specific pathway. Figure 1.1 gives a schematic representation of this process. Using these kernel functions, we first show that incorporating the pathways into the models for the study of cancer significantly enhances the accuracy of the models. In this setting, the prediction model can be expressed by the coefficients used for combining the kernels referred to here as the kernel profile.

The models generated by incorporation of pathways into the classification models for cancer are significantly more interpretable compared to the models generated based on only genomic information. This can be partly attributed to the sparsity of these solutions. Most of the widely used machine learning methods tune the sparsity of the solutions they generate indirectly using different norms of the kernel profile in the learning process. However, this approach can result in very different levels of sparsity in classification models for different cancers. Given the importance of sparsity in classification models for cancer, we have developed a method for learning sparse models with a user-specified degree of sparsity. The number of kernel functions allowed to have a positive weight in the kernel profile is set by the user and the algorithm is able to optimally select the subset of kernel functions with the desired size that gives the best classification model. The problem of selecting the best subset of kernels and simultaneously setting their weights in a way that results in the best classifier is a bi-level non-linear integer optimization problem. We first

use linearization techniques to transform the problem into a single-level semi-infinite optimization problem and then develop cutting-plane algorithms that can solve the problem iteratively.

Next, we consider the possibility of using the data for different cancer cohorts simultaneously through the multitask learning framework. In multitask learning, the data points are divided into several subsets referred to as tasks, and each task has its own prediction model, yet the prediction models for different tasks can influence each other. In the context of building classification models for cancers, each cancer cohort constitutes a task. In multitask learning, the influence of the prediction models for

different tasks on each other has two aspects. Firstly, which tasks do have an effect on each other? and secondly, how is this influence asserted? The tasks that are more similar are naturally expected to have more similar prediction models so it is reasonable to assume that the amount of influence tasks have over each other's prediction models during training must be related to their similarity. Since there is no commonly-used pan-cancer measure of similarity between the cohorts, such a measure must be derived prior to or in the process of building the prediction models. For a given measure of similarity between the cohorts, their kernel profiles can be forced to be identical if the cohorts are deemed sufficiently similar, or the Euclidean distance between their kernel profiles can be penalized by a factor proportional to the similarity between two cohorts.

In this setting, we use co-clustering of pathways and cohorts as one possible avenue for developing accurate and interpretable classification models. In co-clustering, a number of groups are formed that include both pathways and cohorts. Namely, the pathways and the cohorts that have the closest relation are placed in the same group and the interaction of pathways and cohorts that do not belong to the same cluster is penalized. Moreover, similarity between the kernel profiles of cohorts belonging to a cluster is incentivized. The problem of co-clustering pathways and cohorts is a bi-level non-linear optimization problem and is computationally prohibitive. For this reason, we give an iterative approach to find approximate solutions for this problem and show the effectiveness of this algorithm through a large set of numerical experiments.

Alternatively, as a first step towards devising exact solution methods for large scale multitask multiple kernel problems, we consider the problem of optimally clustering cancer cohorts while the cohorts that belong to the same cluster use identical kernel profiles. We introduce a graph representation for each possible solution for the clustering problem in which the cohorts constitute the nodes and their similarity is represented by the arcs. We further encode each cluster of cohorts as a tree in the graph, which we exploit to eliminate the redundancies in the solution space. Then we use the cutting-plane algorithm and the Benders Decomposition (BD) method

in conjunction with refinements for the BD to solve the problem in an efficient and parallel manner. Our numerical experiments show the efficiency of the BDForest method.

1.1 Problem Definition

1.1.1 Discriminating Early- and Late-Stage Cancers

Through the life of a patient a number of tumors might appear in the patients body. However, not all these tumors necessarily develop into life-threatening disease. For this reason, it is crucial to understand what causes the progression of cancers from early-stage to late-stage tumors. The stage of a cancer indicates its spread in the body, with early-stage tumors being local tumors and late-stage tumors being the tumors that have spread to different parts of the body. A number of mechanisms have been identified that help the tumors to self sustain, avoid the attacks of the immune system and facilitate the spread of the cancer. However, a pan-cancer study aiming to identify the most important mechanisms of the progression of cancers has not been conducted in the literature. For this reason, we consider the problem of discriminating early- and late-stage cancers based on the genomic data available about a tumor as one of the classification problems in this work.

1.1.2 Two-Year Survival Prediction

One of the important measures in the study of cancer is the survival time of the patients. This measure helps practitioners devise treatment strategies and evaluate their effectiveness. Hence, understanding the roles of various pathways in the survival chances of patients is crucial. For this reason, discriminating the patients in terms of their two-year survival is one of the classification tasks considered in this work.

1.2 Existing Approaches

In the following, we review the pertinent literature.

1.2.1 Application of Machine Learning in Biomedical Research

Machine learning algorithms have been very successful in recent decades. Specially, in the field of biomedical research, machine learning algorithms have been used for many applications varying from diagnosis and prognosis to drug-design and treatment (Rifai et al. 2006, Li and Zhou 2007, Ji et al. 2010, Yan et al. 2014, Sehhati et al. 2015, Wang et al. 2018).

Machine learning algorithms such as support vector machines (Cortes and Vapnik 1995) and random forests (Breiman 2001) are among the algorithms that have been applied to different prediction tasks related to diagnosis, prognosis, and treatment of cancer. These algorithms obtained very high predictive performance in several applications (Díaz-Uriarte and De Andres 2006, Pang et al. 2006, Statnikov et al. 2008, Nam et al. 2009, Jagga and Gupta 2014, Bhalla et al. 2017). However, the most important aspect of such automated systems should be extracting relevant and meaningful knowledge from data, which is quite difficult to achieve in very high-dimensional and correlated datasets such as genomic measurements, for follow-up studies.

1.2.2 Study of Progression of Cancer Using Genomic Data

Understanding cancer formation and progression from early to late stages is quite important since preventing and treating cancer at early stages is much easier. This work studies the problem of discriminating early- and late-stage cancers from each other using their gene expression profiles. This problem has been addressed in several previous studies (Broët et al. 2006, Ren et al. 2012, Jagga and Gupta 2014, Bhalla et al. 2017, Telonis et al. 2017).

Broët et al. (2006) tried to identify gene expression features that separate early stages from late stages using a statistical score-based approach on microarray data. Similarly, Jagga and Gupta (2014) and Bhalla et al. (2017) developed correlation-based and threshold-based algorithms, respectively, to pick individual genes that separate early-stage patients from late-stage patients for just a single disease (i.e., kidney renal clear cell carcinoma), where they evaluated the quality of gene ex-

pression features they picked by training standard machine learning algorithms such as support vector machines and random forests on them. This kind of scoring/thresholding metrics might identify predictive gene expression signatures with a limited number of features, but their interpretation is quite difficult because of the high-dimensional and correlated input data. In this high-dimensional regime, machine learning algorithms might select different biomarkers as predictive when they use different subsets of the same patient cohort for a given prediction task (Ein-Dor et al. 2005, 2006).

Hanahan and Weinberg (2000) categorized the main mechanisms involved in the transformation of normal cells into malignant cancerous cells, drawing attention to the presence of these mechanisms in the majority of cancers. Jagga and Gupta (2014) used four different classification methods for detection of early stage clear cell renal carcinoma based on genomic information. Tsai et al. (2014) developed a hybrid of artificial bee colony algorithm and support vector machine to predict the stage of ovarian cancer based on genomic information. Using the result of the classification, they were able to identify candidate genes with previously unknown roles in ovarian cancer.

1.2.3 Biochemical Pathways

Using genomic data in machine learning often suffers from the “curse of dimensionality”, which arises in the face of thousands of features for sample sizes of a few hundreds at most. Hence, without prior knowledge of at least some useful features, a purely data-driven classifier is unlikely to validate on fresh data (Jin 2009). For mitigating these effects, machine learning methods based on pathway information have been developed (Rahimi and Gönen 2018, Manica et al. 2019). Pathways refer to series of biochemical interactions involving different genes and intermediate molecules with the goal of producing specific materials. Many biochemical pathways and their link to specific cancers have been identified in the literature (Schaefer et al. 2009, Liberzon et al. 2015).

Some works have focused on pathways that are common in many cancers (Bergers

and Benjamin 2003). A series of works investigate various pathways for a given cohort (Zhang et al. 2014, Feng et al. 2018), and some works identify single pathway-cohort links (Xiao and Meierhofer 2019)

1.2.4 Multiple Kernel Learning

Multiple kernel learning (MKL) methods aim to generate prediction models that use the best combination of a set of given kernel functions (Gönen and Alpaydm 2011). This property of MKL methods enables the fusion of data gathered from dissimilar domains. MKL problems often fall under the rubric of bi-level optimization problems, where the higher-level problem aims to find the optimal combination of kernels and the low-level problem devises the optimal prediction models based on the kernel decided on by the higher-level problem. Rakotomamonjy et al. (2008) and Xu et al. (2010) proposed iterative methods for optimally solving an MKL problem with an ℓ_1 -norm on the kernel weights.

In recent years, there has been an increasing trend in biomedical research for deriving multiple types of data from the same individual. As a natural fit, MKL algorithms have been widely used in biomedical research for integrating multiple high throughput data sources, with applications including but not limited to drug-target interaction prediction (Gönen 2012, Nascimento et al. 2016), cancer biology (Gönen and Margolin 2014, Seoane et al. 2014), as well as clustering and visualization (Wang et al. 2017).

To generate interpretable classification models, we dramatically decrease the number of genes under consideration using pathways as gene sets. However, instead of identifying a list of gene expression features first and then feeding this feature subset into a machine learning algorithm, we proposed to combine these two steps together with the prior knowledge about pathways/gene sets into a unified model. By coupling these parts, we identified relevant biological processes and learned a classifier only on the selected pathways/gene sets at the same time for a given classification task. To this aim, we used the multiple kernel learning methodology (Gönen and Alpaydm 2011), which was developed to combine multiple feature representa-

tions (i.e., views) or multiple similarity measures (i.e., kernels) in the framework of support vector machines. We created multiple views that correspond to input pathways/gene sets from gene expression profiles, calculated multiple kernel matrices on these views, and combined these kernels in a weighted sum rule for our classifier to effectively discard some of the kernel matrices.

1.2.5 Multitask Learning

Although cancers are very heterogeneous, many share similar characteristics and mechanisms. Hence, an important issue in our setting is the multitask nature of the learning problem (Caruana 1997). Xu and Yang (2011) surveyed different multitask approaches that have been explored in the literature, including enforcing similar model parameters for different tasks, enforcing similar model parameters for tasks with some predefined notion of closeness (e.g., hierarchical relationship in the evolutionary tree), enforcing sparsity and co-clustering. Co-clustering refers to simultaneous clustering of features and tasks. Jacob et al. (2009) considered simultaneous learning of the clustering of tasks and the prediction models for each task. Yousefi et al. (2015) used a quadratic regularization term to make the kernel coefficients of different tasks close to each other. They then solved the resulting optimization problem in an iterative manner by alternating between solving for kernel weights and solving for classifier parameters.

Multitask learning relies on commonalities and interdependence between related tasks (Gu et al. 2014, Pan et al. 2016, Tian et al. 2018). In many multitask learning applications, a measure of similarity between the tasks is known a priori. However, despite the huge literature on cancer, there is no widely accepted measure of similarity between different cancers, and the learning algorithm has to decide on this while building the prediction models. This choice to consider two cancers similar or not adds a number of binary variables to the problem. This increases the complexity of the problem significantly since for every set of integer values, there is a different optimal prediction model for each task. Hence, first we use a heuristic method for solving the problem and later on, to produce solutions that are less sensitive to the

choice and calibration of algorithms and exploiting the recent advances in parallel computing technology, we aim to provide an efficient and exact solution method for similar problems.

1.2.6 Sparsity and Optimization Frameworks

The practical benefit and the interpretability of the models of cancer largely depend on the sparsity of these models. This can be attributed to two main reasons. First, there are a very large number of genes, and cancers are highly heterogeneous. Hence, many genes can play minor or major roles in the progression of cancer in each subgroup of a cancer, where the subgroups of the cancer might not themselves be easily distinguishable. Second, because of the intrinsic properties of the genes and their evolution over time, their expression is highly correlated. This results in various representations for the same model of cancer. In the following, a review of the common solutions to these challenges is presented.

Using the ℓ_1 -norm to reach sparse solutions results in selecting only one feature from a set of highly correlated features. This is not a desirable feature, since genes involved in similar mechanisms are highly correlated in many cases. Hence, they used a combination of hinge loss and ℓ_1 -norm that can give sparse solutions while treating correlated genes similarly. Bertsimas et al. (2017b) proposed using a cutting-plane approach to obtain classification rules with sparse solutions.

In the MKL framework, the choice of ℓ_p -norm enforced on the kernel weights directly affects the sparsity of the kernel weights (Xu et al. 2010). Many studies in the literature focus on models that generate solutions less sparse compared to that of group Lasso MKL (Kloft et al. 2009, Yu et al. 2010). However, in some applications, identifying the most relevant data sources is as important as the predictive performance for generating interpretable solutions, which warrants enforcing sparsity on the kernel weights.

Rahimi and Gönen (2018) apply MKL with an ℓ_1 -norm on the kernel weights for discriminating early- and late-stage cancers by integrating pathways into kernel functions enabling them to generate interpretable solutions. However, for many

cancers, many pathways might be chosen in the final classifier. While studying cancers, it is crucial to identify the intrinsic biological mechanisms in the progression of the disease. Therefore, to increase the interpretability of these discrimination models, further sparsity needs to be enforced on the solution.

To enforce sparsity and multitask learning, we extend the MKL formulation by incorporating integer variables. From an optimization perspective, this problem is a bi-level mixed integer nonlinear optimization problems, which can be intractable for large-scale instances of the problem. Despite the inherent difficulty of solving integer programs, recent theoretical advancements and hardware improvements have contributed to significant speed-ups in solving these optimization problems and have gained attention among the machine learning and data mining community (Bertsimas and King 2015, Bertsimas et al. 2017a). Therefore, in this work, we propose several mixed integer linear models based on transforming the bi-level problems into a collection of single-level problems by introducing appropriate variables and constraints into the problem.

To tackle large-scale instances our multitask learning formulations, we employ Benders Decomposition (Benders 1962, Geoffrion 1972) and acceleration techniques such producing Pareto-optimal cuts (Magnanti and Wong 1981). BD is a partitioning technique for solving large-scale mixed integer programs, which is based on the premise that in some classes of problems, when a set of variables are temporarily fixed, the remaining problem is relatively easier to solve. For recent applications and advancements in BD, the reader is referred to Rahmaniani et al. (2017). The multitask learning problem under consideration lends itself naturally to BD method since when certain variables are fixed, the remaining problem can be solved as a set of independent single-task problems. To the best of our knowledge, this study is the first study that employs BD in the context of multitask learning problems.

Chapter 2

BACKGROUND

2.1 Datasets

In this study, we used several cancer cohorts from the Cancer Genome Atlas (TCGA) project to understand differences between early- and late-stage cancers. The cohorts we used in our experiments are publicly available at <https://portal.gdc.cancer.gov>.

2.1.1 RNA-Seq Measurements

TCGA project reported RNA-Seq measurements of 33 cohorts over more than 10,000 tumors and pre-processed these measurements to have a unified RNA-Seq pipeline. For each cohort, we downloaded HTSeq-FPKM files of all primary tumors from the most recent freeze (i.e., Data Release 25 – July 22, 2020), leading to 9,911 files in total. We decided not to include metastatic tumors reported since their underlying biological mechanisms would be very different than primary tumors.

2.1.2 Pathological Stage Information

Cancer is a complex disorder, one of the important aspects in prognosis of cancer is determining its stage. Stage of cancers describes the size and the location of a cancer and it shows where it has spread. knowing the stage of a cancer is crucial in planing treatment and assessing how well the treatment works. Staging in addition to helping practitioners better understand the state of the disease, helps doctors speak in a common language with the entire medical group. Specially, when researchers study the effect of one treatment against others or investigate different outlooks, having a common language plays a crucial rules. For this purpose, doctors use definitions for stages of cancer provided by the American Joint Committee on Cancer (AJCC)

and the International Union Against Cancer (UICC).

The TNM Classification of Malignant Tumors (TNM) is a system for describing the amount and spread of cancer in a patient's body (Gress et al. 2017). **T** describes the size of the original tumor and any spread of cancer into nearby tissue; **N** describes the spread of cancer to nearby lymph nodes; and **M** reports distant metastasis (spread of cancer to other parts of the body). To classify the progression of cancer, the size and extension of the primary tumor, spread into lymphatic tissue, and the presence of metastases is used.

There are five defined stages for cancers based on their localization, starting from zero to four, increasing by spreading in the body. **Stage 0** refers to cancers where the abnormal cells start to form in tissue, **Stage I** when the tumors are small and without spreading, located in a small area. **Stage II** indicates that the cancer has grown but has not spread. In **Stage III**, the tumor has grown, and it may spread to surrounding tissue or the lymph node. In **Stage IV** or the metastatic stage, the tumor has spread from an initial or primary site to secondary or other organs.

For most of the cancer types, it is common that staging start from one to four while **Stage 0** mostly is defined for breast cancer. Some researchers consider **Stage I** as the early stage, and some consider **Stages I** and **II** as early stages. Patients with the same stage of cancer mostly have similar outlooks, so by knowing how much cancer has spread in the patient's body and where the tumors are located doctors can plan treatments, which is often similar for patients with the same cancer stage. Treatment options can vary based on stage of the cancer, for early stage cancer, typically radiation, surgery or combinations of them can be common options while, for late stage cancer, they may treat by chemotherapy.

TCGA project also provided clinical annotations for cancer patients whose tumors were profiled. Since we were interested in separating early- and late-stage cancers from each other, we checked `pathologic_stage` annotation for each patient from the most recent freeze, and there were 6,958 patients with this information.

2.1.3 Dataset Construction

Our datasets consist of the information about **RNA-Seq** measurements for the tumors of patients from different cancer cohorts described in the Cancer Genome Atlas (TCGA) project. The TCGA database contains data related to more than 10,000 patients from 33 cancer cohorts. Each tumor constitutes a data point, and each gene a feature. Overall, **RNA-Seq** measurements of 19,814 genes were considered. We downloaded the **HTSeq-FPKM** files and clinical annotation files of cancer patients to extract their pathological stages and survival characteristics from the latest freeze of the TCGA project (i.e., **Data Release 25** – July 22, 2020).

To be able to train a predictor for the pathological stage of a primary tumor from its gene expression profile, we need both data sources to be available during training. We extracted primary tumors with available **HTSeq-FPKM** file and **pathologic_stage** annotation for each cohort. We then considered primary tumors with **Stage I** annotation as early-stage (i.e., localized cancers) and the remaining tumors with **Stage II**, **Stage III**, or **Stage IV** annotations as late-stage cancers (i.e., regional or distant spreads). Primary tumors annotated with **Stage X** (i.e., 12 tumors in **BRCA**) and **IS** (i.e., 46 tumors in **TGCT**) were not included in our analyses even if they have their gene expression profiles available. After this step, we only included cohorts with at least 20 tumors both from early- and late-stage categories in our final dataset list.

Table 2.1 gives the list of 15 datasets that were used in our first set of experiments, namely, **E1**, together with details about the number of samples in early- and late-stage cancers. The total number of primary tumours included is 5,547, and dataset sizes vary between 65 (i.e., **KICH**) and 1,067 (i.e., **BRCA**). The percentage of early-stage tumours vary between 5.79% (i.e., 25/432 in **HNSC**) and 67.90% (i.e., 55/81 in **TGCT**). We then considered an alternative labelling of early- and late-stage cancers by assigning primary tumours annotated with **Stage I** or **II** to early-stage (i.e., localised cancers and regional spreads) and primary tumours annotated with **Stage III** or **IV** to late-stage (i.e., distant spreads). In this case, when we only included cohorts with at least 20 tumours both from early- and late-stage categories, we

obtained 18 datasets at the end. Table 2.1 also gives the list of 18 datasets that were used in our second set of experiments, namely, E2, together with details about the number of samples in early- and late-stage cancers. The total number of primary tumours included is 6,038, where the percentage of early-stage tumours vary between 21.99% (i.e., 95/432 in HNSC) and 81.69% (i.e., 406/497 in LUSC).

Table 2.1: Summary of 20 TCGA cohorts that we used for discriminating early- and late-stage cancers (E1 and E2).

Cohort	Disease name	Stage				E1			E2		
		I	II	III	IV	Early	Late	Total	Early	Late	Total
ACC	Adrenocortical carcinoma	9	37	16	15	—	—	—	46	31	77
BLCA	Bladder urothelial carcinoma	2	130	140	134	—	—	—	132	274	406
BRCA	Breast invasive carcinoma	181	619	247	20	181	886	1,067	800	267	1,067
COAD	Colon adenocarcinoma	75	176	128	64	75	368	443	251	192	443
ESCA	Esophageal carcinoma	16	69	49	8	16	126	142	85	57	142
HNSC	Head and neck squamous cell carcinoma	25	70	78	259	25	407	429	95	337	429
KICH	Kidney chromophobe	20	25	14	6	20	45	65	45	20	65
KIRC	Kidney renal clear cell carcinoma	265	57	123	82	265	262	527	322	205	527
KIRP	Kidney renal papillary cell carcinoma	172	21	51	15	172	87	259	193	66	259
LIHC	Liver hepatocellular carcinoma	171	86	85	5	171	176	347	257	90	347
LUAD	Lung adenocarcinoma	274	121	84	26	274	231	505	395	110	505
LUSC	Lung squamous cell carcinoma	244	162	84	7	244	253	497	406	91	497
MESO	Mesothelioma	10	16	44	16	—	—	—	26	60	86
PAAD	Pancreatic adenocarcinoma	21	146	3	4	21	153	174	—	—	—
READ	Rectum adenocarcinoma	30	51	51	24	30	126	156	81	75	156
SKCM	Skin cutaneous melanoma	2	66	27	3	—	—	—	68	30	98
STAD	Stomach adenocarcinoma	53	111	150	38	53	299	352	164	188	352
TGCT	Testicular germ cell tumours	55	12	14	0	55	26	81	—	—	—
THCA	Thyroid carcinoma	281	52	112	55	281	219	500	333	167	500
UVM	Uveal melanoma	0	39	36	4	—	—	—	39	40	79
Total		1,883	3,664	5,547		3,738	2,300	6,038			

For the survival analysis problem, we are interested in predicting whether the survival time of patients is shorter than or exceeds two years, which is a commonly used threshold. Patients with `days_to_death` less than two years were classified as `Dead`, and patients whose `days_to_last_followup`, `days_to_last_known_alive` or `days_to_death` was greater than two years were classified as `Alive`. As detailed in Table 2.2, considering cohorts with at least 20 data points in each class, 20 cohorts with a total of 5,175 patients in them met these conditions.

Table 2.2: Summary of 20 TCGA cohorts that we used for the two-year survival analysis experiment (S1).

Cohort	Disease name	Alive	Dead	Total
BLCA	Bladder urothelial carcinoma	132	142	274
BRCA	Breast invasive carcinoma	594	43	637
CESC	Cervical squamous cell carcinoma and endocervical adenocarcinoma	141	42	183
COAD	Colon adenocarcinoma	207	71	278
ESCA	Esophageal carcinoma	39	48	87
GBM	Glioblastoma multiforme	23	103	126
HNSC	Head and neck squamous cell carcinoma	214	157	371
KIRC	Kidney renal clear cell carcinoma	354	83	437
KIRP	Kidney renal papillary cell carcinoma	149	24	173
LAML	Acute myeloid leukemia	41	71	112
LGG	Brain lower grade glioma	238	56	294
LIHC	Liver hepatocellular carcinoma	140	91	231
LUAD	Lung adenocarcinoma	216	106	322
LUSC	Lung squamous cell carcinoma	222	127	349
MESO	Mesothelioma	30	50	80
OV	Ovarian serous cystadenocarcinoma	234	74	308
PAAD	Pancreatic adenocarcinoma	36	83	119
SARC	Sarcoma	151	54	205
STAD	Stomach adenocarcinoma	99	122	221
UCEC	Uterine corpus endometrial carcinoma	314	49	363
Total		3,579	1,596	5,175

2.2 Gene Set Databases

In addition to predicting the stage of a tumor, we would like to understand the biological mechanisms that differentiate early- and late-stage cancers from each other. For this purpose, we can, for example, use pathways and/or gene sets defined in the literature. We use two collections of pathways from the literature. These collections provide information about groups of genes that have dependencies or similarities in their functions. In the following, we give a brief description of these databases.

2.2.1 Hallmark Collection

Liberzon et al. (2015) have built the **Hallmark** gene set collection by combining and summarizing more than 10,000 gene sets present in the Molecular Signatures Database (MSigDB). This effort has been carried out using automated approaches

and expert opinion with purpose of reducing the redundancy between the gene sets present in MSigDB. As given in Figure 2.1, the **Hallmark** dataset contains 50 pathways with sizes varying between 32 and 200 genes. The pathways collected in this database are related to the development of cells, metabolism, DNA damage and signalling among other categories.

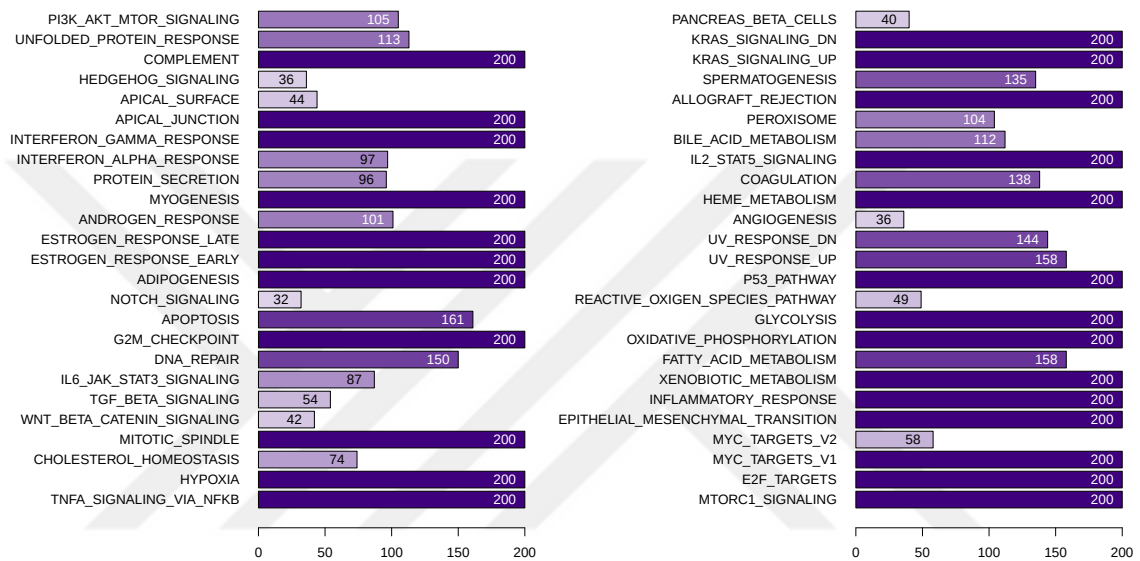


Figure 2.1: Names and sizes of 50 pathways contained in the **Hallmark** collection.

2.2.2 PID Collection

The Pathway Interaction Database (PID) is one of the well known databases in cancer research for signaling pathways, provided by collaboration between the US National Cancer Institute and Nature Publishing Group. It focuses on molecular interactions and reactions that take place in the cells. In addition to cancer research, it is used in other research areas where pathway interactions are of interest including neuroscience, developmental biology and immunology. As given in Figure 2.2, the PID collection contains 196 pathways with sizes varying between 10 and 137 genes. Schaefer et al. (2009) give a summary of this database and comparisons with other databases.

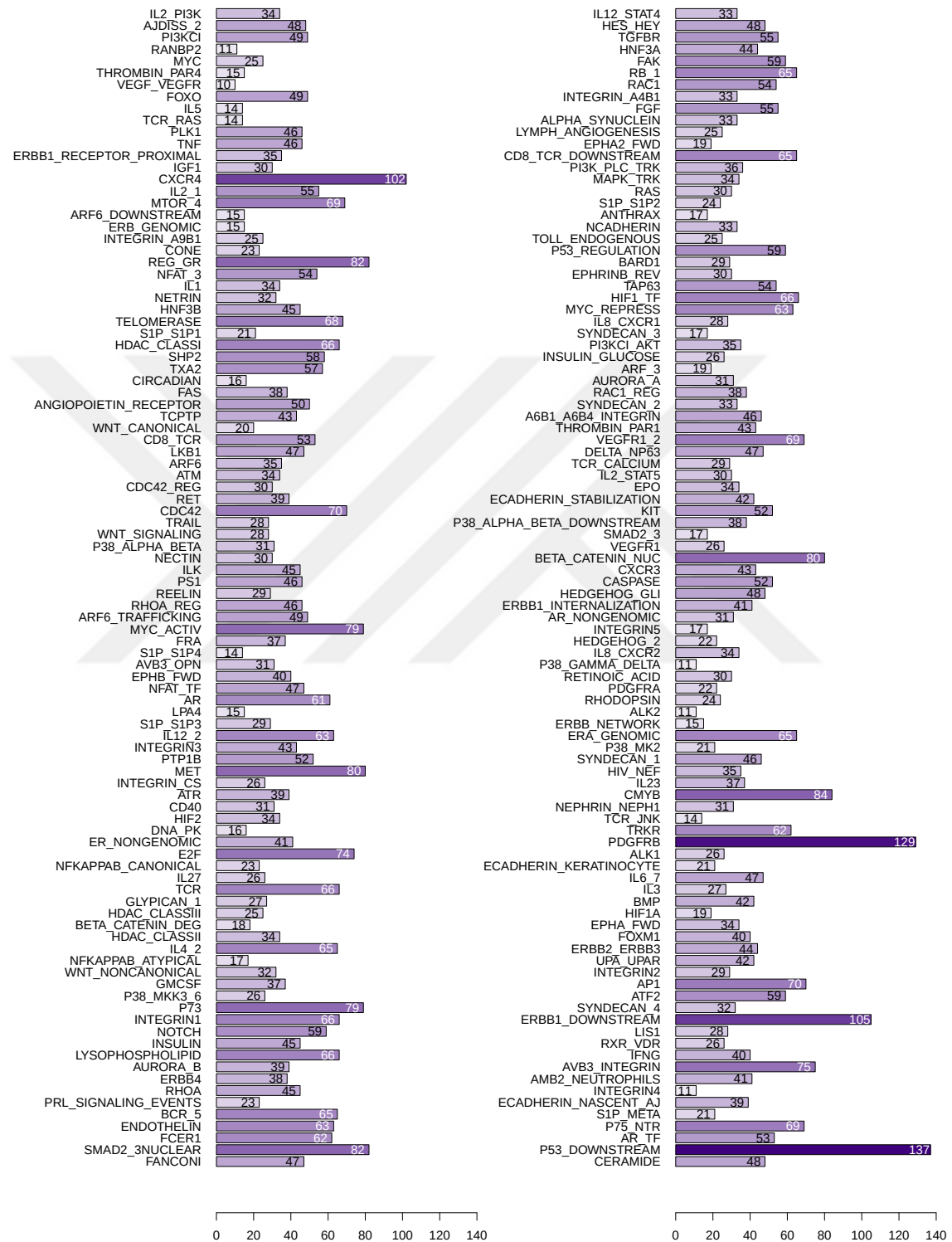


Figure 2.2: Names and sizes of 196 pathways contained in the PID collection.

2.3 Methods

To compare the predictive performance of the methods developed here, we use well known methods that are widely used for classification in biomedical research as the baseline algorithms. In the following, a brief description of these methods is provided.

2.3.1 Random Forest

A *Decision Tree Classifier* (DTC) is a prominent classification algorithm, in which the discrimination function is characterized by a set of sequential decision making rules. DTC is attractive for problems where the decision region is too complex (e.g., high dimensional) but can be approximated by a set of simpler local decision regions (Safavian and Landgrebe 1991). On the other hand, DTCs may suffer from over-fitting to the observable decision region and lack of generalizability on the out-of-sample data points.

To improve generalizability, Breiman (2001) among several other researchers suggested constructing multiple independent DTCs using randomly selected data points and features. By combining these weak decision trees using an ensemble strategy (i.e., letting DTCs vote for the most popular class), more robust classification algorithms known as Random Forests (RFs) can be obtained (Breiman 2001). RFs are chosen as the baseline algorithm since they are widely used in bioinformatics studies, particularly to predict disease phenotypes from genomic measurements (Díaz-Uriarte and De Andres 2006, Pang et al. 2006, Statnikov et al. 2008, Nam et al. 2009). Although they are reported to be very accurate classifiers in terms of predictive performance in several applications, their knowledge extraction capability is quite restricted. Decision tree models in RFs are usually constructed on randomly selected bootstrap samples, which makes knowledge extraction very sensitive to this bootstrapping step. Additionally, in many applications, the rules generated by RFs can be very long and hard to decipher.

2.3.2 Support Vector Machine

One of the most well-known and well-studied approaches to solving classification problems is Support Vector Machine (SVM) (Cortes and Vapnik 1995). Despite the simple idea behind this method, SVM is a powerful algorithm used widely in many applications due to its efficiency, accuracy and relative interpretability (Muller et al. 2001, Tian et al. 2013, Bai et al. 2014). SVM is a supervised learning method and using the labels of data points, the algorithm finds the best hyperplane for classifying data points. The best hyperplane in this algorithm is the one which maximizes the margin between the classes. The data points close to the margin that contribute to forming the discriminant function are called support vectors. Using the support vectors instead of the whole training dataset SVM can be efficient in memory usage. This algorithm can use a technique referred to as the kernel trick, to transform the data points to other feature spaces to model non-linear relations. One of the advantages of using the kernel trick for SVM is its ability to simplify the high dimensionality of the feature space.

Assume that a set of N data points in the form of $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ is given, where $\mathbf{x}_i \in \mathbb{R}^D$ represents the features of data point i , and $y_i \in \mathcal{Y} = \{-1, 1\}$ its class label. SVMs formulate the binary classification as a convex quadratic optimization problem (Cortes and Vapnik 1995). The mathematical details of SVMs is given in the following. The optimization problem of binary classification SVMs can be written as

$$\begin{aligned}
 & \text{minimize } \frac{1}{2} \mathbf{w}^\top \mathbf{w} + C \sum_{i=1}^N \xi_i \\
 & \text{with respect to } \mathbf{w} \in \mathbb{R}^D, \quad \boldsymbol{\xi} \in \mathbb{R}^N, \quad b \in \mathbb{R} \\
 & \text{subject to } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i \\
 & \quad \quad \quad \xi_i \geq 0 \quad \forall i,
 \end{aligned} \tag{2.1}$$

where $\mathbf{w}$ is the set of weights assigned to features, C is a positive regularization parameter, $\boldsymbol{\xi}$ is the set of slack variables, D is the number of input features (e.g.,

the number of genes in gene expression profiles), and b is the intercept parameter. Instead of solving this primal optimization problem, usually the corresponding dual optimization problem is solved (i) to reduce the number of decision variables and (ii) to be able to integrate kernel functions into the model, leading to nonlinear models. The Lagrangian function for this problem is given as

$$\mathcal{L} = \frac{1}{2} \mathbf{w}^\top \mathbf{w} + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i (y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^N \beta_i \xi_i, \quad (2.2)$$

and the derivatives with respect to the decision variables of the primal problem are:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 &\Rightarrow \mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i \\ \frac{\partial \mathcal{L}}{\partial b} = 0 &\Rightarrow \sum_{i=1}^N \alpha_i y_i = 0 \\ \frac{\partial \mathcal{L}}{\partial \xi_i} = 0 &\Rightarrow C = \alpha_i + \beta_i \quad \forall i. \end{aligned}$$

With plugging these back into the Lagrangian function, the objective value of the dual problem can be written as

$$\begin{aligned} &\text{maximize} \quad \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \\ &\text{with respect to } \boldsymbol{\alpha} \in \mathbb{R}^N \\ &\text{subject to} \quad \sum_{i=1}^N \alpha_i y_i = 0 \\ &\quad \quad \quad C \geq \alpha_i \geq 0 \quad \forall i, \end{aligned} \quad (2.3)$$

where there are N decision variables instead of $(D + N + 1)$ decision variables, and the $\mathbf{x}_i^\top \mathbf{x}_j$ term can be replaced with a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ to obtain nonlinear models.

Chapter 3

MULTIPLE KERNEL LEARNING ON GENE SETS

Improving the prognosis, diagnosis and treatment of cancer has been a prominent focus point of researchers for years. The wide diversity of cancers, in terms of the organs they originate from, their symptoms and their known causes have contributed to a persistent difficulty in understanding the mechanisms of cancer.

Given this difficulty, machine learning tools have been widely used for improving the different aspects of cancer treatment. Specially with the ever increasing availability of genomic data, machine learning has been used extensively for exploiting the explanatory power of genomic data in regards to cancer. Using genomic data in machine learning faces a number of challenges. First, genomic data usually has thousands of features while for many applications the number of samples is in the order of tens or hundreds. This might cause many methods to generate models that are prone to over-fitting the data. Secondly, genomic data tends to have highly correlated features, which makes the prediction models to be very sensitive to the specifics of the algorithms, preventing us from reaching robust solutions.

Instead of identifying a list of gene expression features first and then feeding this feature subset to a machine learning algorithm, we proposed to combine these two steps together with the prior knowledge about pathways/gene sets into a unified model. By coupling these parts, we identified relevant biological processes and learned a classifier only on the selected pathways/gene sets at the same time for a given classification task. To this aim, we used the multiple kernel learning methodology (Gönen and Alpaydın 2011), which was developed to combine multiple feature

representations (i.e., views) or multiple similarity measures (i.e., kernels) in the framework of support vector machines. We created multiple views that correspond to input pathways/gene sets from gene expression profiles, calculated multiple kernel matrices on these views, and combined these kernels in a weighted sum rule for our classifier to effectively discard some of them.

Our contributions in this chapter are three-fold:

1. We formulated a multiple kernel learning algorithm on gene sets to identify relevant biological processes and to learn a classification model conjointly.
2. We tested the performance of our proposed algorithm on the problem of separating early- and late-stage cancers from each other using their gene expression profiles, to the best of our knowledge, on the largest number of diseases.
3. We then showed that our algorithm was able to extract meaningful and disease-specific information related to the mechanism of cancer progression.

3.1 Multiple Kernel Learning

The dual formulation of the SVM problem as given in quadratic optimization problem (5.1) allows for incorporating nonlinearities into the SVM model by transforming the feature space using a mapping function $\Phi : \mathcal{X} \rightarrow \mathcal{X}'$ and replacing $\mathbf{x}_i^\top \mathbf{x}_j$ with $k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}_j)$, commonly known as the kernel trick. Consequently, the dual SVM problem can be restated as the following alternative formulation:

$$\begin{aligned}
 & \text{maximize} \quad \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\
 & \text{with respect to} \quad \boldsymbol{\alpha} \in \mathbb{R}^N \\
 & \text{subject to} \quad \sum_{i=1}^N \alpha_i y_i = 0 \\
 & \quad \quad \quad C \geq \alpha_i \geq 0 \quad \forall i.
 \end{aligned} \tag{3.1}$$

We remark that even though $\mathcal{X}'$ is usually a lower dimensional space than the original feature space $\mathcal{X}$, the matrix induced by the kernel function k (i.e., $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$) is always an $N \times N$ matrix. Consequently, the kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ can be a convex combination of P individual kernel functions of the form

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{m=1}^P \eta_m k_m(\mathbf{x}_i, \mathbf{x}_j),$$

with $\eta_m \geq 0$ the weight of kernel k_m , for $m = 1, \dots, P$, such that $\sum_{m=1}^P \eta_m = 1$. Each kernel k_m represents a similarity matrix between the data points, which can be calculated on the same input representation or on different input representations (or views).

Kernel functions are beneficial for machine learning models in that they provide a shortcut to transform the input features to desired form while avoiding complex calculations. Using the kernel trick in SVM leads to efficiently dealing with datasets with large number of features compared to the number of data points. In SVM, the kernel function determines the shape of the discriminant hyperplane. Kernel functions can have different forms such as linear kernels, polynomial kernels and Gaussian (or radial basis) kernels. For a linear kernel function

$$k_L(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$$

the discriminant function is a hyperplane. When the degree is more than one, the more general polynomial kernel of the following form is obtained

$$k_P(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + c)^p,$$

where c is a given parameter and $p \in \mathbb{N}$ is the degree of the polynomial. One of the most general and prominent kernel functions is the Gaussian kernel as defined in the following

$$k_G(\mathbf{x}_i, \mathbf{x}_j) = \exp \left(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_j)}{2\sigma^2} \right),$$

where σ is the width parameter. The value of the Gaussian kernel and other pairwise kernel functions depend on the distance between points. Hence, the pairwise kernels can be interpreted as a similarity measure between the data points.

The predictive performance of SVMs is highly dependent on the kernel function used. The standard approach is to select the best kernel function among a set of candidates using a cross-validation strategy. However, instead of selecting a single kernel function, using a weighted combination of input kernels might give better predictive performance, which is known as MKL (Gönen and Alpaydın 2011). MKL algorithms might combine different kernels calculated on the same input representation or combine kernels calculated on different input representations (i.e., multiview learning). In bioinformatics applications, the same data points can be represented with different measurements (e.g., gene expression, methylation and copy number measurements from the same set of tumors). Instead of combining predictive models trained on each representation (i.e., late integration) or concatenating these input representations into a joint one before learning (i.e., early integration), we can calculate kernel matrices on each representation and learn how to combine them in the predictive algorithm during inference (i.e., intermediate integration).

In this study, we were interested in identifying biological mechanisms that differentiate early- and late-stage cancers from each other. To this aim, we created a separate kernel matrix for each gene set and combined them using an MKL algorithm, namely, group Lasso MKL, which can be used to find a sparse combination (i.e., mostly zero kernel weights due to ℓ_1 -norm) of the input kernels (Xu et al. 2010). This property of group Lasso MKL is particularly important in our context, since the gene sets that correspond to kernel matrices with nonzero weights might be related to the differentiation between early- and late-stage cancers (we shall present gene set-based kernels in Section 3.3.)

3.2 Group Lasso Multiple Kernel Learning

Group Lasso MKL solves the following optimization problem to find the kernel weights and other model parameters simultaneously.

$$\begin{aligned}
& \text{minimize } J(\boldsymbol{\eta}) \\
& \text{with respect to } \boldsymbol{\eta} \in [0, 1]^P \\
& \text{subject to } \sum_{m=1}^P \eta_m = 1,
\end{aligned} \tag{3.2}$$

where $\boldsymbol{\eta}$ is the set of kernel weights, P is the number of input kernels, and $J(\boldsymbol{\eta})$ corresponds to the optimization problem in (3.1), in which $k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{m=1}^P \eta_m k_m(\mathbf{x}_i, \mathbf{x}_j)$. The optimization problem in (3.2) cannot be solved to find optimum $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ variables simultaneously. That is why we use an alternating optimization strategy to optimize them in an iterative manner (Xu et al. 2010). At iteration (t) , we have to first solve the inner optimization problem (i.e., a standard SVM model) to find up-to-date support vector coefficients, namely, $\boldsymbol{\alpha}^{(t)}$, and we then find up-to-date kernel weights, namely, $\boldsymbol{\eta}^{(t+1)}$, using $\boldsymbol{\alpha}^{(t)}$ and the previous kernel weights, namely, $\boldsymbol{\eta}^{(t)}$. Algorithm 1 presents an overview of this procedure. (Xu et al. 2010) also present a generalization of this algorithm for the case where, instead the ℓ_1 -norm kernel weights (i.e., $\sum_{m=1}^P \eta_m = 1$), ℓ_p -norm with $p \geq 1$ is enforced on the kernel weights (i.e., $\|\boldsymbol{\eta}\|_p \leq 1$).

Figure 3.1 also presents a schematic overview of the proposed MKL algorithm. In Section 3.3, we discuss how kernels are produced in our study in more details.

3.3 Gene Set-Based Kernel Functions

In the context of this study, each data point corresponds to a tumor and is represented by the gene expression profile of the tumor. To construct kernels, we use the pathways and/or gene sets identified by various studies and curated in databases such as Hallmark (Liberzon et al. 2015) and PID (Schaefer et al. 2009) collections (see Figures 2.1 and 2.2). These collections provide information about groups of

Algorithm 1 Group Lasso multiple kernel learning algorithm

- 1: $t \leftarrow 1$
- 2: Initialized kernel weights as $\eta_m^{(t)} = \frac{1}{P}$ for each $m = 1, \dots, P$
- 3: **while** not converged **do**
- 4: **SOLVE** dual SVM (3.1) with the up-to-date kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{m=1}^P \eta_m^{(t)} k_m(\mathbf{x}_i, \mathbf{x}_j)$ to obtain the up-to-date support vector coefficients $\boldsymbol{\alpha}^{(t)}$ and the objective function value $J(\boldsymbol{\eta}^{(t)})$
- 5: Find the up-to-date kernel weights $\boldsymbol{\eta}^{(t+1)}$, using $\boldsymbol{\alpha}^{(t)}$ and the previous kernel weights $\boldsymbol{\eta}^{(t)}$ using the following equation

$$\eta_m^{(t+1)} = \frac{\eta_m^{(t)} \sqrt{\sum_{i=1}^N \sum_{j=1}^N \alpha_i^{(t)} \alpha_j^{(t)} y_i y_j k_m(\mathbf{x}_i, \mathbf{x}_j)}}{\sum_{o=1}^P \eta_o^{(t)} \sqrt{\sum_{i=1}^N \sum_{j=1}^N \alpha_i^{(t)} \alpha_j^{(t)} y_i y_j k_o(\mathbf{x}_i, \mathbf{x}_j)}} \quad \forall m \quad (3.3)$$

- 6: **end while**

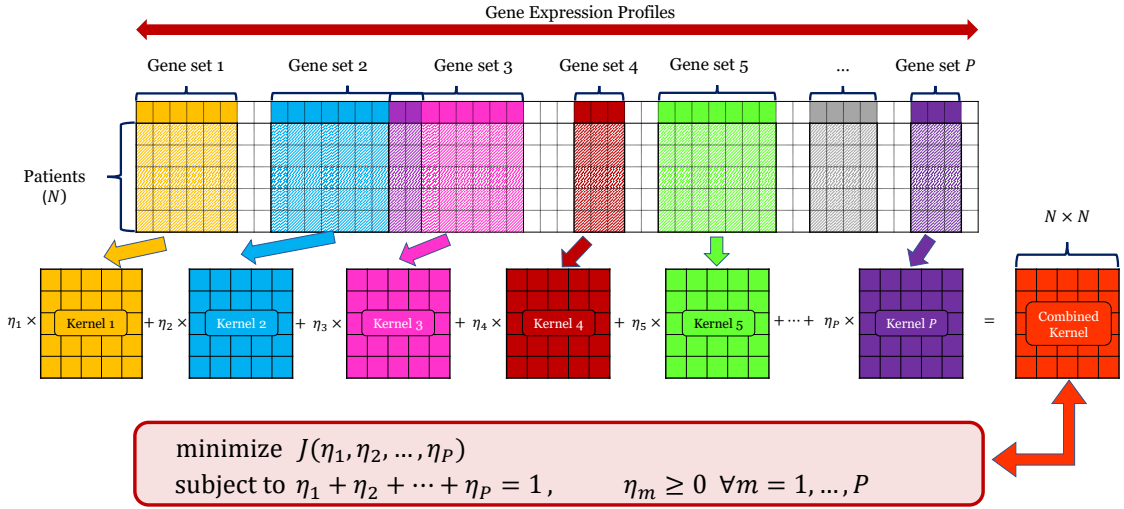


Figure 3.1: Overview of multiple kernel learning on gene sets.

genes that have dependencies or similarities in their functions. Each gene set conveys a specific biological state or process and displays coherent expression in cancers, thereby each representing the dataset from a different perspective.

For a given gene set $\mathcal{G}$, let $\mathbf{x}[\mathcal{G}]$ denote the sub-vector of $\mathbf{x}$ whose indices (i.e.,

genes) are contained in $\mathcal{G}$. Note that $\mathbf{x} \in \mathbb{R}^D$, where D is the number of genes (e.g., around 20,000), while $\mathbf{x}[\mathcal{G}] \in \mathbb{R}^{|\mathcal{G}|}$, which lives in a much lower-dimensional space (e.g., $32 \leq |\mathcal{G}| \leq 200$ for the **Hallmark** gene sets). To calculate a similarity measure between gene expression profiles of tumors based on gene set $\mathcal{G}$, we use the Gaussian kernel defined as

$$k_{\mathcal{G}}(\mathbf{x}_i, \mathbf{x}_j) = \exp \left(-\frac{(\mathbf{x}_i[\mathcal{G}] - \mathbf{x}_j[\mathcal{G}])^\top (\mathbf{x}_i[\mathcal{G}] - \mathbf{x}_j[\mathcal{G}])}{2\sigma_{\mathcal{G}}^2} \right),$$

where we pick the kernel width parameter $\sigma_{\mathcal{G}}$ as the mean of pairwise Euclidean distances between training instances. It is also worth mentioning that since $|\mathcal{G}| \ll D$, computing a kernel matrix based on gene set $\mathcal{G}$ requires significantly fewer arithmetic operations, hence computationally much easier than computing a kernel based on the entire features.

3.4 Discussion

With the advancements in molecular characterization technologies, it has become a standard practice to profile tumor biopsies of cancer patients. These tumor profiles have been extensively used in efforts of understanding molecular mechanisms of cancer formation and progression. For example, if we can successfully determine the molecular mechanisms of progression from early to late stages, we can make use of this information to develop new preventive or therapeutic strategies to stop or slow down this progression. Here, we addressed the problem of discriminating early- and late-stage cancers from each other using their gene expression profiles.

We developed a computational framework (see Figure 7.1) to evaluate the predictive performances of machine learning algorithms on 20 diseases from TCGA collection (see Table 2.1) for this classification task. Our proposed MKL on gene sets algorithm is able to integrate prior knowledge about pathways/gene sets into the model and to extract relative importances of the input gene sets in addition to learning a classification model. The main contribution of our proposed approach comes from performing classification and gene set selection using ℓ_1 -norm regular-

ization conjointly in a unified formulation. By doing so, we can eliminate some gene sets (i.e., noisy or irrelevant gene sets) from the classification model during training instead of using whole gene expression profiles, which leads to more robust and accurate classifiers.



Chapter 4

SPARSE MULTIPLE KERNEL LEARNING WITH OPTIMAL KERNEL SELECTION

Identifying the mechanisms that govern the severity of cancers has an immense importance for selecting effective treatment strategies. In the previous chapters, the study of the severity of cancers, using genomic data together with pathway information and the multiple kernel learning framework has been discussed. However, given the large number of pathways and even larger number of genes involved in the process of cancers becoming more severe, obtaining interpretable models needs machine learning methods with a flexible notion of sparsity. For this reason, the sparse multiple kernel learning (SpMKL) algorithm that produces models with desired degree of sparsity is proposed here. SpMKL is a two-step algorithm that first identifies the most relevant pathways and then uses this knowledge for building more accurate prediction models. SpMKL employs refinements on the cutting-plane framework and other well-known iterative algorithms for solving multiple kernel learning problems. However, for many cancers, many pathways might be chosen in the final classifier.

In this chapter, to generate interpretable solutions for discriminating early- and late-stage tumors as well as for two-year survival outcome prediction, we developed an MKL method that generates extremely sparse discrimination models by identifying the most relevant pathways during the learning process. This chapter has four main contributions:

1. The sparse MKL (SpMKL) method is introduced as a two-step method for enforcing sparsity on multiple kernel learning. The first step is cast as an integer programming problem to identify relevant kernel functions. In the second step, using the level of significance identified for kernel functions in the first

step, an MKL problem with kernel specific linear penalty terms is developed to provide prediction models with the custom level of sparsity surpassing the level reached by methods that use ℓ_1 - or ℓ_2 -norm.

2. A three-phase algorithm is developed based on the cutting-plane method for solving the optimization problem in the first step.
3. A closed-form solution is formulated for optimizing the kernel weights in the second step MKL problem and we show that this iterative method has a performance comparable to that of the closed-form solution method for the penalty-free MKL problem.
4. We use SpMKL for discriminating early- and late-stage tumors as well as for two-year survival outcome prediction for 27 cancer cohorts and compared the discrimination models generated by SpMKL to the models generated by five other methods in terms of their accuracy and sparsity.

In summary, we introduce a two-step method to produce extremely sparse kernel weights in MKL framework. The goal of the first step is to produce a measure of significance for the kernel functions. We cast the first step problem as an integer programming problem, where the selection of active kernels is restricted by a desired number of kernel functions using binary variables. Based on the level of significance identified for kernel functions in the first step, we introduce kernel specific linear penalty terms to MKL problem of the second step to enforce sparsity and to improve the predictive performance simultaneously.

4.1 Sparse Multiple Kernel Learning

In statistical learning theory, the goal of a learning algorithm is to obtain a predictor with a small error (or risk) (Vapnik 1992). Since only a few observations of the underlying distribution of the data points is available, one can only use the so-called *empirical risk* as a proxy for the risk. By the law of large numbers, when a large

number of data points are available, empirical risk provides a good estimate of the true risk. However, the caveat of minimizing the empirical risk is that, in practice, we always have a small sample, and the learner may *over-fit* to the training dataset. Over-fitting happens when the learner performs well on the sample data points, but generalizes poorly on the out-of-sample data points, particularly when the sample data is noisy. To overcome over-fitting, a common approach is to control the level of complexity of the learner proportional to the sample size. In many classification algorithms, such as SVMs and MKL, complexity of the learner can be considered as the number of features or kernels used. Therefore, to avoid over-fitting, one can restrict the number of features/kernels with non-zero coefficients.

The number of kernels used can be controlled in two ways: (i) by explicitly restricting the number of kernels allowed to attain a positive weight, or (ii) by penalizing the kernels with positive weights. In this study, we propose a two-step algorithm that combines both approaches. In the first step, we modify the mathematical formulation of the MKL algorithm by adding binary variables that determine whether a kernel is used (binary MKL). We then control the number of kernels used by constraining these binary variables at the desired levels. The output of this step is a set of kernels identified as significant. In the second step (penalized MKL), we use the level of significance identified for each kernel in the first step to penalize the insignificant kernels.

4.1.1 Sparse Multiple Kernel Learning via Binary Kernel Selection

To identify the most informative kernels (i.e., gene sets in this application) with the highest contribution to binary classification problem, sparsity is enforced on MKL by restricting the number of kernels that the model can use. Let G be the number of kernels allowed to attain a nonzero weight in formulation (3.2). This restriction can be enforced by constraining the number of nonzero entries of $\boldsymbol{\eta}$ to G via $\|\boldsymbol{\eta}\|_0 \leq G$; however, adding this constraint to (3.2) makes it a non-convex optimization problem. Alternatively, one can linearize this constraint by introducing a binary variable g_m for each kernel m and adding constraints $\{\sum_{m=1}^P g_m \leq G\}$ and $\{\eta_m \leq g_m \ \forall m\}$ to the

model in (3.2). Consequently, the optimization problem in (3.2) can be reformulated as the following mixed integer programming problem:

$$\begin{aligned}
& \text{minimize } J(\boldsymbol{\eta}) \\
& \text{with respect to } \boldsymbol{\eta} \in [0, 1]^P, \quad \mathbf{g} \in \{0, 1\}^P \\
& \text{subject to } \sum_{m=1}^P \eta_m = 1 \\
& \quad \sum_{m=1}^P g_m \leq G \\
& \quad \eta_m \leq g_m \quad \forall m,
\end{aligned} \tag{4.1}$$

which is referred to here as the binary MKL problem. Note that the optimization problem in (4.1) involves minimizing a convex function over a set of linear constraints, with the additional integrality requirement on the binary variables $\mathbf{g}$. Hence, to solve this problem, a cutting-plane algorithm can be deployed. Since, for each $\boldsymbol{\eta}$, $J(\boldsymbol{\eta})$ is the maximum of the problem in (3.1), we have

$$J(\boldsymbol{\eta}) \geq \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{m=1}^P \eta_m \kappa_m(\boldsymbol{\alpha}) \quad \forall \boldsymbol{\alpha} \in \mathcal{A}, \tag{4.2}$$

where, $\mathcal{A}$ is the set of feasible support vector coefficients defined as

$$\mathcal{A} = \{\boldsymbol{\alpha} \in [0, C]^N : \sum_{i=1}^N \alpha_i y_i = 0\}, \tag{4.3}$$

and, for notational convenience, we define $\kappa_m(\boldsymbol{\alpha})$ as

$$\kappa_m(\boldsymbol{\alpha}) = \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k_m(\mathbf{x}_i, \mathbf{x}_j). \tag{4.4}$$

Consequently, the optimization problem in (4.1) can be rewritten as the following semi-infinite mixed integer linear program (SIMILP), which we refer to as the master

problem (MP).

$$\begin{aligned}
& \text{minimize } \Gamma \\
& \text{with respect to } \Gamma \in \mathbb{R}, \quad \boldsymbol{\eta} \in [0, 1]^P, \quad \mathbf{g} \in \{0, 1\}^P \\
& \text{subject to } \sum_{m=1}^P \eta_m = 1 \\
& \quad \sum_{m=1}^P g_m \leq G \\
& \quad \eta_m \leq g_m \quad \forall m \\
& \quad \Gamma \geq \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{m=1}^P \eta_m \kappa_m(\boldsymbol{\alpha}) \quad \forall \boldsymbol{\alpha} \in \mathcal{A}.
\end{aligned} \tag{4.5}$$

4.1.2 Sparse Multiple Kernel Learning via Penalized Kernel Selection

Provided that a measure of importance of kernels is available (e.g., obtained from the first step), one can use this measure to improve the predictive performance of MKL by shifting the focus of MKL towards using more informative kernels. Let $\theta_m \in [0, 1]$ be a given parameter indicating the level of relevance of kernel m . This information is incorporated into the MKL framework by penalizing the weight of the irrelevant kernels in the form of

$$\begin{aligned}
& \text{minimize } J(\boldsymbol{\eta}) + \rho \sum_{m=1}^P (1 - \theta_m) \eta_m \\
& \text{with respect to } \boldsymbol{\eta} \in [0, 1]^P \\
& \text{subject to } \sum_{m=1}^P \eta_m = 1,
\end{aligned} \tag{4.6}$$

where ρ is a regularization parameter. The optimization problem in (4.6) is referred to as the penalized MKL problem. Note that the only difference between this formulation and the problem in (3.2) is that the objective function includes an additional linear penalty term. Moreover, since $\sum_{m=1}^P \eta_m = 1$, the objective function of (4.6) can be replaced with $J(\boldsymbol{\eta}) - \rho \sum_{m=1}^P \theta_m \eta_m$.

4.2 Solution Methods

We now present solution methods for solving each step of the SpMKL algorithm.

4.2.1 Solving the Binary MKL Problem

Note that MP contains exponentially many constraints, but only a few of these constraints will be active at optimality. Therefore, this problem can be tackled in an iterative manner. Replacing $\mathcal{A}$ with a finite subset $\tilde{\mathcal{A}}$ yields a relaxation of problem in (4.5), thus the optimal value of MP with $\tilde{\mathcal{A}}$ in place of $\mathcal{A}$ provides a lower bound for the binary MKL problem. On the other hand, any feasible solution for problem (4.1) provides an upper bound for the binary MKL problem. Starting with an initial $\tilde{\mathcal{A}}$, the relaxed MP is progressively strengthened by iteratively growing $\tilde{\mathcal{A}}$ (i.e., by adding constraints of the form $\Gamma \geq \sum_{i=1}^N \alpha_i - \sum_{m=1}^P \eta_m \kappa_m(\boldsymbol{\alpha})$) on the fly until the upper and lower bounds are sufficiently close. An overview of this procedure is given in Algorithm 2, in which convergence of the algorithm is controlled by parameters t_{max} (maximum number of iterations) and ζ (optimality gap threshold).

Algorithm 2 Cutting-plane algorithm for solving the binary MKL problem

```

1:  $UB \leftarrow +\infty, LB \leftarrow -\infty, \text{Gap} \leftarrow +\infty, t \leftarrow 1, \tilde{\mathcal{A}} \leftarrow \emptyset$ 
2: while  $\text{Gap} > \zeta$  and  $t \leq t_{max}$  do
3:   SOLVE problem (4.5) with  $\tilde{\mathcal{A}}$  in place of  $\mathcal{A}$  to obtain  $(\Gamma^{(t)}, \boldsymbol{\eta}^{(t)}, \mathbf{g}^{(t)})$ 
4:   SOLVE dual SVM (3.1) with  $\boldsymbol{\eta}^{(t)}$  as the kernel weights to obtain  $\boldsymbol{\alpha}^{(t)}$  and  $J(\boldsymbol{\eta}^{(t)})$ 
5:    $\tilde{\mathcal{A}} \leftarrow \tilde{\mathcal{A}} \cup \{\boldsymbol{\alpha}^{(t)}\}, LB \leftarrow \Gamma^{(t)}$ 
6:   if  $UB > J(\boldsymbol{\eta}^{(t)})$  then
7:      $UB \leftarrow J(\boldsymbol{\eta}^{(t)}), (\boldsymbol{\eta}^*, \mathbf{g}^*) \leftarrow (\boldsymbol{\eta}^{(t)}, \mathbf{g}^{(t)})$ 
8:   end if
9:    $\text{Gap} \leftarrow \frac{UB - LB}{LB}, t \leftarrow t + 1$ 
10: end while
```

4.2.2 A Three-Phase Algorithm for Solving the Binary MKL Problem

Solving an integer programming problem in MP at each iteration of Algorithm 2, especially when P is large, can be computationally prohibitive. Additionally, starting with $\tilde{\mathcal{A}} = \emptyset$ may lead to slow convergence of Algorithm 2. To remedy this, Algorithm 2 is performed in three phases: warm-up, kernel selection and kernel weight fine-tuning. Figure 4.1 gives a schematic view of the phases of the algorithm.

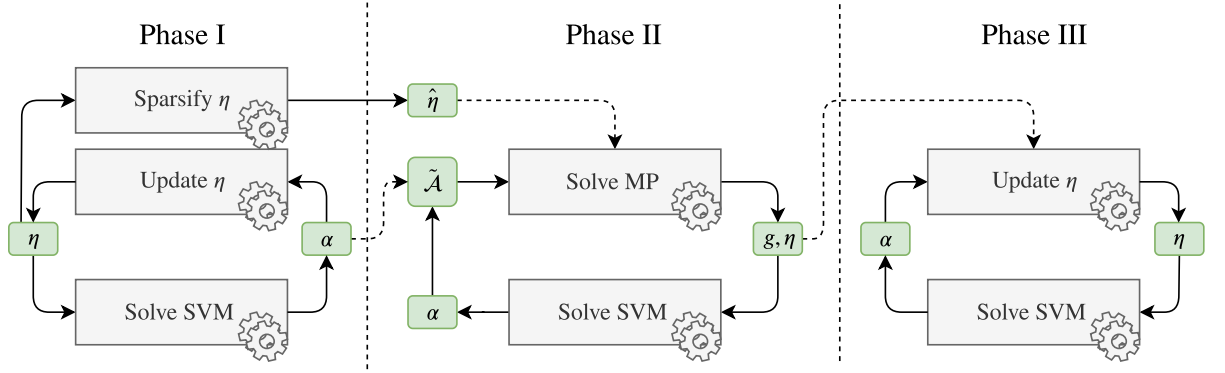


Figure 4.1: Flowchart of the different phases in the three-phase algorithm.

Phase I: Warm-up. The warm-up phase serves two purposes: (i) to find a good initial incumbent sparse solution $\boldsymbol{\eta}^*$ and a corresponding upper bound UB , and (ii) to initialize $\tilde{\mathcal{A}}$ with suitable $\boldsymbol{\alpha}$ values. To find these initial values, instead of enforcing sparsity explicitly using binary variables $\mathbf{g}$, a relaxation of (4.1) is solved by relaxing the integrality requirements on the $\mathbf{g}$ variables. Observe that this relaxation of (4.1) is equivalent to eliminating the $\mathbf{g}$ variables and the associated sparsity constraints from (4.1), which, in turn, is equivalent to the optimization problem in (3.2). The optimization problem in (3.2) can be solved using a cutting-plane algorithm or the iterative algorithm described in Algorithm 1. Deploying either algorithm yields a series of $\{(\boldsymbol{\eta}^{(t)}, \boldsymbol{\alpha}^{(t)})\}$ values. Since $\boldsymbol{\alpha}^{(t)} \in \mathcal{A}$, these support vector coefficients can be directly used to initialize $\tilde{\mathcal{A}}$. Additionally, as described in Proposition 1 below, a sparse solution $\hat{\boldsymbol{\eta}}^{(t)}$ can be recovered with least deviation from $\boldsymbol{\eta}^{(t)}$ by setting the lowest $P - G$ components of $\boldsymbol{\eta}^{(t)}$ to zero and normalizing the value of the remaining components so that the obtained vector is a valid kernel weight vector. Then, the

best $\hat{\boldsymbol{\eta}}^{(t)}$ is used to initialize the incumbent sparse solution $\boldsymbol{\eta}^*$.

Proposition 1. *Let $\boldsymbol{\eta}$ be a given weight vector. Let $\hat{P}$ be the set of G kernels with highest weight in $\boldsymbol{\eta}$, and define $\hat{\boldsymbol{\eta}}$ such that $\hat{\eta}_m = 0$ for each $m \notin \hat{P}$ and $\hat{\eta}_m = \frac{\eta_m}{\sum_{o \in \hat{P}} \eta_o}$ for each $m \in \hat{P}$. Then, $\hat{\boldsymbol{\eta}}$ has the least ℓ_1 distance from $\boldsymbol{\eta}$ among all weight vectors with at most G positive components.*

Proof. For an arbitrary subset $\bar{P}$, define $\Delta(\bar{P})$ as the set of weight vectors that comply with $\bar{P}$, that is $\Delta(\bar{P}) = \{\bar{\boldsymbol{\eta}} \in \mathbb{R}_+^P : \sum_{m=1}^P \bar{\eta}_m = 1, \bar{\eta}_m = 0 \ \forall m \notin \bar{P}\}$. Let $\delta(\bar{P})$ denote the least ℓ_1 distance of $\boldsymbol{\eta}$ from weight vectors complying with $\bar{P}$. Note that for any $\bar{\boldsymbol{\eta}} \in \Delta(\bar{P})$ we have $\|\boldsymbol{\eta} - \bar{\boldsymbol{\eta}}\|_1 = \sum_{m \in \bar{P}} |\eta_m - \bar{\eta}_m| + \sum_{m \notin \bar{P}} \eta_m$. Therefore, $\delta(\bar{P})$ can be computed by $\delta(\bar{P}) = \sum_{m \notin \bar{P}} \eta_m + \min_{\bar{\boldsymbol{\eta}} \in \Delta(\bar{P})} \{\sum_{m \in \bar{P}} |\eta_m - \bar{\eta}_m|\}$. By triangle inequality we have $\sum_{m \in \bar{P}} |\eta_m - \bar{\eta}_m| \geq \sum_{m \in \bar{P}} (\bar{\eta}_m - \eta_m) = 1 - \sum_{m \in \bar{P}} \eta_m$, where the last equality holds because $\sum_{m \in \bar{P}} \bar{\eta}_m = 1$. Therefore, it follows that $\delta(\bar{P}) \geq 2(1 - \rho)$, where $\rho = \sum_{m \in \bar{P}} \eta_m$.

Observe that defining $\bar{\boldsymbol{\eta}}$ such that $\bar{\eta}_m = 0$ for each $m \notin \bar{P}$ and $\bar{\eta}_m = \frac{\eta_m}{\sum_{o \in \bar{P}} \eta_o}$ for each $m \in \bar{P}$ yields $\delta(\bar{P}) = 2(1 - \rho)$ because $\sum_{m \in \bar{P}} |\eta_m - \bar{\eta}_m| = \sum_{m \in \bar{P}} |\eta_m - \frac{\eta_m}{\sum_{o \in \bar{P}} \eta_o}| = (\frac{1}{\rho} - 1)\rho = 1 - \rho$. Consequently, a subset $\hat{P}$ yielding the highest value for $\rho = \sum_{m \in \hat{P}} \eta_m$ together with the sparsified kernel weight $\hat{\boldsymbol{\eta}}$ yield the least ℓ_1 distance from $\boldsymbol{\eta}$. $\square$

Phase II: Kernel selection. The purpose of this phase is to identify the most informative kernels. Note that the solution obtained at the end of Phase I is a feasible solution for problem (4.1); hence, the upper bound obtained at the end of this phase is also a valid upper bound for Algorithm 2. The kernel selection phase is essentially performing Algorithm 2 using the set $\tilde{\mathcal{A}}$, upper bound UB and solution $\boldsymbol{\eta}^*$ obtained at the end of the warm-up phase. Performing Algorithm 2 for an additional t_{kernel} number of iterations yields a selection of kernels $\mathbf{g}^*$, which translates to selecting up to G kernels as the optimal kernels.

Phase III: Kernel weight fine tuning. If Phase II terminates with an optimality gap greater than ζ , the kernels identified by $\mathbf{g}^*$ in Phase II are picked and the optimal weights for the selected kernels are computed using the existing algo-

gorithms for solving regular MKL problem (e.g., the iterative procedure described in Algorithm 1).

4.2.3 An Efficient Algorithm for Solving the Penalized MKL Problem

To solve the optimization problem given in (4.6), the iterative procedure described in Algorithm 1 is adapted to update the kernel weights without using a solver. Because of the linear term in the objective function of (4.6), the kernel weights cannot be directly updated using equation (3.3). Instead, the up-to-date kernel weights $\boldsymbol{\eta}^{(t+1)}$ are computed using (4.7) below.

Proposition 2. *At iteration $t + 1$ of the iterative algorithm for solving penalized MKL problem, the up-to-date kernel weights $\boldsymbol{\eta}^{(t+1)}$ can be computed via*

$$\eta_m^{(t+1)} = \frac{\tilde{f}_m^{(t)}}{\sqrt{\lambda - \rho\theta_m}} \quad \forall m, \quad (4.7)$$

where $\tilde{f}_m^{(t)} = \eta_m^{(t)} \sqrt{\kappa_m(\boldsymbol{\alpha}^{(t)})}$ and λ solves $\sum_{m=1}^P \frac{\tilde{f}_m^{(t)}}{\sqrt{\lambda - \rho\theta_m}} = 1$.

Proof. The equations can be easily derived using the procedure given by Xu et al. (2010). $\square$

Note that if $\rho\theta_m = \bar{\theta} \quad \forall m$, then $\lambda = \bar{\theta} + (\sum_{o=1}^P \eta_o^{(t)} \sqrt{\kappa_o(\boldsymbol{\alpha}^{(t)})})^2$. In particular, if $\bar{\theta} = 0$ then (4.7) is equivalent to (3.3). For general $\rho\theta_m$, λ can be efficiently computed using a simple line search method to solve $\sum_{m=1}^P \frac{\tilde{f}_m^{(t)}}{\sqrt{\lambda - \rho\theta_m}} = 1$.

4.3 Discussion

In this chapter, we proposed SpMKL model for sparse selection of kernel functions for a multiple kernel classification problem and developed efficient algorithms to solve the problem instances in each step of SpMKL. To solve the integer optimization problem in the first step, we proposed a three-phase algorithm based on cutting-plane algorithm to distribute the burden of the different aspects of solving the problem in different phases. To solve the MKL problem with linear penalty terms in the

second step of SpMKL, we developed an efficient alternating algorithm to optimally compute the kernel weights without using an optimization solver.

SpMKL was applied to the problems of cancer stage prediction and two-year survival outcome prediction based on genomic data and pathway information. As presented in Section 7.3, SpMKL was able to predominantly improve the predictive performance compared against various MKL algorithms on gene sets while using fewer genes and pathways. The sparsity of solutions produced by SpMKL was compared to that of SimpleMKL, group Lasso MKL and ℓ_2 -MKL algorithms using the Gini index and statistical tests. SpMKL proved to be efficient in generating discrimination models that are significantly more sparse and more interpretable compared to the results obtained by other methods for these problems.

Given the similarities between the cancer cohorts reported frequently in the literature, the data available for different cancer cohorts can be integrated in a multitask framework. This will allow the data gathered about more common cancers help devise more accurate discrimination models for cancers with scarce data. The following chapters explore this subject.

Chapter 5

MULTITASK MULTIPLE KERNEL LEARNING

Throughout the previous chapters, we have established that identifying which pathways play an important role in progression of the cancer stage is of great interest. Given that there are similarities in the underlying mechanisms of different cancers, in addition to the considerable correlation in the genomic data, there is a need for machine learning methods that can take these aspects of genomic data into account. Furthermore, using machine learning for studying multiple cancer cohorts together with a collection of molecular pathways creates an opportunity for knowledge extraction.

In this chapter, we used machine learning algorithms to build prediction tools with the ability to discriminate early- and late-stage cancers, to give insight into the molecular mechanisms involved in each type of cancer and to identify mechanisms that are common to a set of cancer types. Given these goals, we developed models that are able to discriminate between early- and late-stage cancers using genomic information available early in the treatment process while linking these predictions to relevant biological mechanisms, in addition to deriving groups of cohorts and gene sets for extracting information about the underlying mechanisms of different cancers. MKL algorithms allow for integration of data from different sources. In other words, the tasks of interpreting the biological implications of genomic information in the form of pathways/gene sets and performing diagnosis using genomic information, which were done separately before, can be done jointly using MKL.

5.1 Problem Definition

Although the underlying biological mechanisms of cancer types differ from each other, there are some similarities between cancer types. Therefore, learning about a certain disease might benefit from information about others. The multitask learning framework allows for benefiting from such situations. In multitask learning, different tasks (cohorts) are learned conjointly (Caruana 1997, Xu and Yang 2011). This approach usually improves the generalization performance of tasks with limited data.

Many multitask learning methods rely on a predetermined notion of similarity between the tasks. However, the similarity between the cancer cohorts is still a question of interest. Hence, there is a need for methods that are able to identify these similarities between the cohorts. In addition, merely knowing that two cohorts are similar without any information about why they are similar has limited practical use. To remedy this, the learning method proposed here is able to identify the similarities between the cohorts in addition to identifying the mechanisms that cause the similarity. To this end, we developed a multitask MKL (MTMKL) algorithm with a co-clustering step that is capable of learning clusters of cohorts and gene sets that best explain the distinction between “early”- and “late”-stage cancers.

Our contributions in this chapter are three-fold:

1. We formulated a multitask multiple kernel learning method with a co-clustering model on gene sets to identify relevant biological processes and to learn task-specific classification models conjointly.
2. We devised a cutting-plane algorithm for solving this problem efficiently with given similarity matrices.
3. We employed a hybrid cutting-plane and diversification algorithm for finding the similarity matrices of cohorts and gene sets resulting in more accurate prediction models with more capability for knowledge extraction.

5.2 Co-clustering Step

Despite the prevalence of cancer and the increase in the collected information about it, there are cohorts that have limited data due to their rare occurrence. However, it is possible to improve the prediction models for such cohorts with the help of similar cohorts. For this purpose, similar cohorts need to be identified. Given this setting, we were interested in developing a learning method that can benefit from the gene set information and can model all cohorts conjointly. In the following, we define our problem and discuss the suitable learning algorithms.

Provided T classification tasks (i.e., tumors in our application), let us define the training dataset of task $s \in \{1, \dots, T\}$ be given as $\{(\mathbf{x}_{si}, y_{si})\}_{i=1}^{N_s}$, where $\mathbf{x}_{si}$ is a vector containing gene expression levels and y_{si} is the staging label of patient i 's tumor. $y_{si} = +1$ indicates that patient i 's tumor is in “early” stage, and $y_{si} = -1$ indicates that it is in “late” stage.

As discussed earlier, the SVM formulation does not allow for the direct incorporation of the pathway information into the learning problem. For this purpose, we used the group Lasso MKL method, which helps in identifying the kernels that have the most predictive power. Xu et al. (2010) gives an iterative algorithm with closed-form update rules for group Lasso MKL. In this study, each pathway/gene set corresponds to a subset of the genes (i.e., features) and we build a kernel using each subset, hence the group Lasso MKL method decides which collection of the pathways (i.e., kernels) has the best predictive power (Rahimi and Gönen 2018). For a given task s , the kernelized dual SVM problem 3.1 is formulated as

$$\begin{aligned}
 & \text{maximize } J_s(\boldsymbol{\eta}_s) = \sum_{i=1}^{N_s} \alpha_{si} - \frac{1}{2} \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} \alpha_{si} \alpha_{sj} y_{si} y_{sj} \sum_{m=1}^P \eta_{sm} k_{sm}(\mathbf{x}_{si}, \mathbf{x}_{sj}) \\
 & \text{with respect to } \boldsymbol{\alpha}_s \in \mathbb{R}^{N_s} \\
 & \text{subject to } \sum_{i=1}^{N_s} \alpha_{si} y_{si} = 0 \\
 & \quad C \geq \alpha_{si} \geq 0 \quad \forall i.
 \end{aligned} \tag{5.1}$$

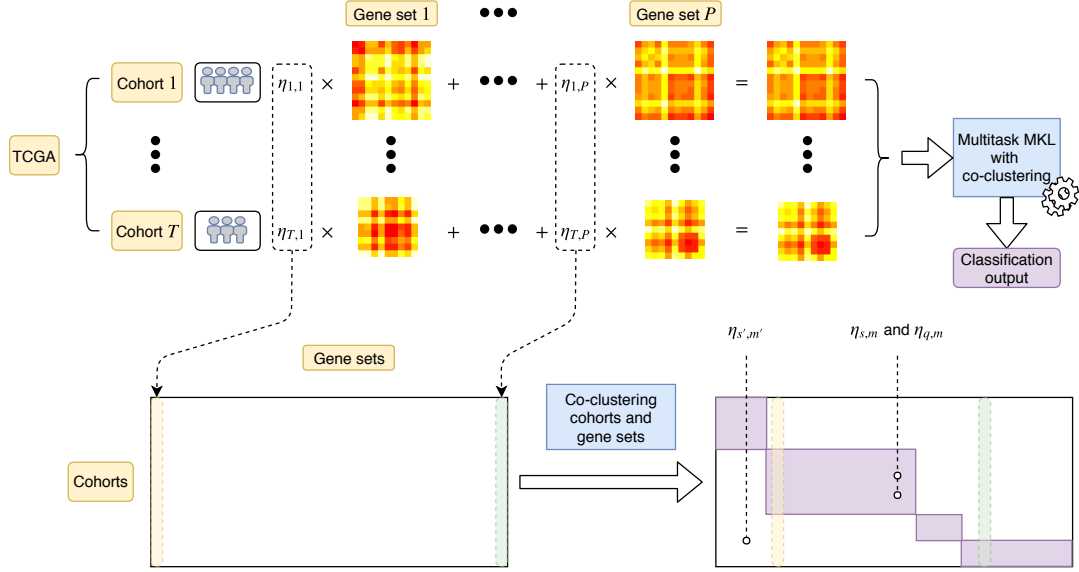


Figure 5.1: Overview of the proposed multitask multiple kernel learning algorithm.

Previous research shows that different cancer types, despite having distinct biological mechanisms, have similarities in their mechanisms. To exploit this information, we used a multitask learning formulation, where different tasks are learned simultaneously, allowing for tasks with limited data to benefit from other similar tasks. In this study, each cohort is considered as a distinct task. In multitask learning, a measure of similarity between the tasks is given as an input or is inferred through the learning process. Here, we aim to identify similarities between cancer cohorts (i.e., tasks) in terms of their underlying mechanisms. Hence, the appropriate framework for learning the closeness between the tasks is co-clustering. Co-clustering refers to assigning two heterogeneous sets of items to a number of clusters, such that a cluster can include items from both sets.

Co-clustering builds a predefined number of clusters of cohorts and pathways (i.e., tasks and kernels). Namely, in co-clustering, a cluster contains both cohorts and pathways. Given the contribution of each pathway to prediction for each cohort denoted by η_{sm} (s indexes the tasks, and m indexes the kernels), the joint multitask

multiple kernel learning (MTMKL) and co-clustering problem can be stated as

$$\begin{aligned}
 & \text{minimize} \quad \sum_{s=1}^T \left[J_s(\boldsymbol{\eta}_s) + \lambda_1 \sum_{m=1}^P (1 - \theta_{sm}) \eta_{sm} + \lambda_2 \sum_{q=1}^T \phi_{sq} \|\boldsymbol{\eta}_s - \boldsymbol{\eta}_q\|^2 \right] \\
 & \text{with respect to} \quad \{\boldsymbol{\eta}_s \in \mathbb{R}_+^P\}_{s=1}^T, (\boldsymbol{\theta}, \boldsymbol{\phi}) \in \mathcal{C} \\
 & \text{subject to} \quad \sum_{m=1}^P \eta_{sm} = 1 \quad \forall s,
 \end{aligned} \tag{5.2}$$

where $J_s(\boldsymbol{\eta}_s)$ denotes the objective function for the optimal classification in cohort s for a given $\boldsymbol{\eta}_s$, and is obtained by solving the dual SVM problem in (5.1). $\mathcal{C}$ is the set of feasible co-clustering solutions defined in (5.3), in which χ_{ks} (ψ_{km}) is a binary variable that indicates if cohort s (pathway m) is assigned to cluster k . θ_{sm} is 1 if cohort s and pathway m belong to the same cluster (i.e., $\psi_{km} \times \chi_{ks} = 1$ for some cluster k) and ϕ_{sq} is 1 if cohorts s and q are assigned to the same cluster (i.e., $\chi_{ks} \times \chi_{kq} = 1$ for some cluster k). The constraints ensure that the $\boldsymbol{\eta}$ vector for each cohort as well as the assignment of cohorts/pathways to the clusters are properly defined.

$$\mathcal{C} = \left\{ \begin{array}{ll} \theta_{sm}, \phi_{sq}, \psi_{km}, \chi_{ks} \in \{0, 1\} & \\ \sum_{k=1}^K \psi_{km} = 1 & \forall m \\ \sum_{k=1}^K \chi_{ks} = 1 & \forall s \\ \theta_{sm} \geq \psi_{km} + \chi_{ks} - 1 & \forall (s, m, k) \\ \theta_{sm} \leq \psi_{km} - \chi_{ks} + 1 & \forall (s, m, k) \\ \phi_{sq} \geq \chi_{ks} + \chi_{kq} - 1 & \forall (s, q, k) \\ \phi_{sq} \leq \chi_{ks} - \chi_{kq} + 1 & \forall (s, q, k). \end{array} \right. \tag{5.3}$$

From the two penalty terms of the objective value, the first one penalizes using a gene set for a cohort, if the cohort and the gene set do not belong to the same cluster, whereas the second one penalizes using different kernel weights for cohorts that are in the same cluster. The intended effects of these penalty terms are illustrated in Figure 5.1. As presented in this figure, using the genomic data for each gene set

in each cohort, a kernel function is constructed. Then for each cohort s , the kernel functions are combined using the coefficients η_{sm} . Prediction models for different cohorts affect each other through penalties based on the co-clustering of cohorts and pathways e.g., since cohort s' and pathway m' do not belong to the same cluster, $\eta_{s'm'}$ will be penalized. Likewise, since cohorts s and q belong to the same cluster, the distance between η_{sm} and η_{qm} used as a penalty term.

For a given set of $\{\boldsymbol{\eta}_s\}_{s=1}^T$ values, the co-clustering problem can be reduced to an instance of the Graph Partitioning Problem, which falls under the category of NP-hard problems (Andreev and Racke 2006). Hence, we solve the overall problem via an iterative approach, where, in each iteration, we first solve an MTMKL problem with fixed clusters and then solve a co-clustering problem with fixed kernel weights using a heuristic method.

5.3 Multitask Learning Step

We apply the MTMKL algorithm with a co-clustering step to different validation splits of the dataset and aggregate the co-clustering results for building similarity matrices, then we pass these matrices as inputs to the MTMKL algorithm to assess the predictive performance. In the following, we discuss our MTMKL algorithm with a co-clustering step in more detail.

For a given co-clustering $(\hat{\boldsymbol{\chi}}, \hat{\boldsymbol{\psi}})$ of the cohorts and gene sets, and the corresponding similarity matrices $\hat{\boldsymbol{\phi}}$ and $\hat{\boldsymbol{\theta}}$, the problem is equivalent to an MTMKL problem with additional linear and quadratic penalty terms of the following form:

$$\begin{aligned}
 & \text{minimize} \quad \sum_{s=1}^T \left[J_s(\boldsymbol{\eta}_s) + \lambda_1 \sum_{m=1}^P (1 - \hat{\theta}_{sm}) \eta_{sm} + \lambda_2 \sum_{q=1}^T \hat{\phi}_{sq} \|\boldsymbol{\eta}_s - \boldsymbol{\eta}_q\|^2 \right] \\
 & \text{with respect to} \quad \{\boldsymbol{\eta}_s \in \mathbb{R}_+^P\}_{s=1}^T \\
 & \text{subject to} \quad \sum_{m=1}^P \eta_{sm} = 1 \quad \forall s.
 \end{aligned} \tag{5.4}$$

Note that because of the penalty terms in the objective function of (5.4), the closed form formulation of Xu et al. (2010) can no longer be employed to update the

$\boldsymbol{\eta}_s$ values. Hence, we propose a cutting-plane algorithm for solving the MTMKL problem in (5.4), noting that this cutting-plane approach can be reliably used for MTMKL without co-clustering or any prior knowledge of the similarity between the tasks. The cutting-plane algorithm allows for solving this problem even if the similarity matrices do not correspond to an exact co-clustering of cohorts and gene sets.

Now, we elaborate the proposed cutting-plane method. First, note that problem (5.4) can be equivalently restated as

$$\begin{aligned}
& \text{minimize} \quad \sum_{s=1}^T \Gamma_s + \rho(\boldsymbol{\eta}) \\
& \text{with respect to} \quad \{\boldsymbol{\eta}_s \in \mathbb{R}_+^P\}_{s=1}^T, \boldsymbol{\Gamma} \in \mathbb{R}_+^T \\
& \text{subject to} \quad \sum_{m=1}^P \eta_{sm} = 1 \quad \forall s \\
& \quad \quad \quad \Gamma_s \geq J_s(\boldsymbol{\eta}_s) \quad \forall s
\end{aligned} \tag{5.5}$$

where

$$\rho(\boldsymbol{\eta}) = \lambda_1 \sum_{s=1}^T \sum_{m=1}^P (1 - \hat{\theta}_{sm}) \eta_{sm} + \lambda_2 \sum_{s=1}^T \sum_{q=1}^T \hat{\phi}_{sq} \|\boldsymbol{\eta}_s - \boldsymbol{\eta}_q\|^2.$$

Let $\mathcal{A}_s$ denote the set of feasible solutions of the dual SVM problem in (5.1). For any s and feasible $\boldsymbol{\eta}_s$, $J_s(\boldsymbol{\eta}_s)$, being the optimal value of dual SVM problem in (5.1), implies that, for every $\boldsymbol{\alpha}_s \in \mathcal{A}_s$, the following holds:

$$J_s(\boldsymbol{\eta}_s) \geq \sum_{i=1}^{N_s} \alpha_{si} - \frac{1}{2} \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} \alpha_{si} \alpha_{sj} y_{si} y_{sj} \sum_{m=1}^P \eta_{sm} k_{sm}(\mathbf{x}_{si}, \mathbf{x}_{sj}).$$

Therefore, problem (5.5) (or equivalently (5.4)) can be restated as

$$\begin{aligned}
& \text{minimize} \quad \sum_{s=1}^T \Gamma_s + \rho(\boldsymbol{\eta}) \\
& \text{with respect to} \quad \{\boldsymbol{\eta}_s \in \mathbb{R}_+^P\}_{s=1}^T, \boldsymbol{\Gamma} \in \mathbb{R}_+^T \\
& \text{subject to} \quad \sum_{m=1}^P \eta_{sm} = 1 \quad \forall s \\
& \quad \quad \quad \Gamma_s \geq \sum_{i=1}^{N_s} \alpha_{si} - \frac{1}{2} \sum_{m=1}^P \eta_{sm} \kappa_{sm}(\boldsymbol{\alpha}_s) \quad \forall \boldsymbol{\alpha}_s \in \mathcal{A}_s,
\end{aligned} \tag{5.6}$$

where $\kappa_{sm}(\boldsymbol{\alpha}_s) = \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} \alpha_{si} \alpha_{sj} y_{si} y_{sj} k_{sm}(\mathbf{x}_{si}, \mathbf{x}_{sj})$.

We refer to the quadratic program in (5.6) as the master problem (MP). Note that for any proper subsets of $\mathcal{A}_s$, MP provides a lower-bound on the optimal value of (5.4). On the other hand, for any feasible $\{\boldsymbol{\eta}_s\}_{s=1}^T$, $\sum_s J_s(\boldsymbol{\eta}_s) + \rho(\boldsymbol{\eta})$ provides an upper-bound on the optimal value of (5.4). Therefore, starting with initial subsets of $\mathcal{A}_s$, we iteratively grow these sets by alternating between the MP and the dual SVM problems until the optimal solution is obtained. The proposed cutting-plane algorithm is detailed in Algorithm 3, where UB and LB are the upper- and lower-bounds on the optimal value of (5.4), respectively. ζ and $t_{\max}$ are parameters determining the convergence tolerance of the algorithm.

Note that for a fixed value of $\boldsymbol{\eta}_s$ the dual SVM problems can be solved independently for each s . Hence, we are able to solve the multitask problem to optimality by solving the single-task problems. This property allows for distributing the computational burden of a large multitask problem over parallel machines. In addition, since each single-task problem is solved separately, the latest advances in solving single-task problems can be employed for the multitask problem. We use group Lasso MKL method for solving the single-task problems.

For solving the co-clustering problem in a reasonable time, we use a heuristic algorithm. To decrease the risk of converging to local optima, we choose a predefined number of heterogeneous co-clustering solutions in each step and find the optimal kernel weights for each of these co-clusterings. Hence, in each step, the choice of the

Algorithm 3 Cutting-plane algorithm pseudo-code

```

1:  $UB \leftarrow +\infty, LB \leftarrow -\infty, t \leftarrow 1$ 
2:  $\mathcal{A}_s \leftarrow \emptyset$ 
3: Terminate  $\leftarrow \mathbf{FALSE}$ 
4: while (Terminate = FALSE) do
5:     SOLVE MP (5.6) and obtain  $\boldsymbol{\eta}_s$  and  $z_{MP}^*$ 
6:      $LB \leftarrow z_{MP}^*$ 
7:     for  $s \in \{1 \dots T\}$  do
8:         SOLVE dual SVM (5.1) for  $s$ , and obtain  $\boldsymbol{\alpha}_s$  and  $J_s(\boldsymbol{\eta}_s)$ 
9:          $\mathcal{A}_s \leftarrow \mathcal{A}_s \cup \{\boldsymbol{\alpha}_s\}$ 
10:    end for
11:     $UB \leftarrow \min\{UB, \sum_s J_s(\boldsymbol{\eta}_s) + \rho(\boldsymbol{\eta})\}$ 
12:    Gap  $\leftarrow \frac{UB - LB}{LB}$ 
13:    if Gap  $\leq \zeta$  or  $t \geq t_{\max}$  then
14:        Terminate  $\leftarrow \mathbf{TRUE}$ 
15:    end if
16:     $t \leftarrow t + 1$ 
17: end while

```

optimal co-clustering is not solely based on the penalty terms. Algorithm 4 gives a description of this process.

Many studies in the multitask learning literature use a given measure of similarity between the tasks. Such a measure of similarity, referred to as the similarity matrix, can come from different sources, e.g., evolutionary tree (if the data points belong to different species) or expert opinion. Co-clustering of cohorts and gene sets can be a basis for deriving such similarity matrices relating cohorts to other cohorts ($\boldsymbol{\phi}$) and cohorts to gene sets ($\boldsymbol{\theta}$). For this purpose, we perform the co-clustering multiple times on samples of the data and aggregate the results in the form of frequency matrices $\bar{\boldsymbol{\phi}}$ and $\bar{\boldsymbol{\theta}}$, i.e., the entries of $\bar{\boldsymbol{\phi}}$ and $\bar{\boldsymbol{\theta}}$ show how often two cohorts (or a cohort and a gene set) have been assigned to the same cluster. Figure 5.2 illustrates such a cohort-cohort similarity matrix.

Algorithm 4 MTMKL algorithm with a co-clustering step pseudo-code

```

1: Initialize  $\hat{\boldsymbol{\eta}}_s$  with an arbitrary set of  $\boldsymbol{\eta}_s$  vectors
2: Derive a co-clustering  $(\hat{\boldsymbol{\chi}}, \hat{\boldsymbol{\psi}})$  based on  $\hat{\boldsymbol{\eta}}_s$ 
3:  $OPT \leftarrow (\hat{\boldsymbol{\eta}}, \hat{\boldsymbol{\chi}}, \hat{\boldsymbol{\psi}})$ 
4:  $z_{OPT} \leftarrow \sum_s J_s(\hat{\boldsymbol{\eta}}_s) + \rho(\hat{\boldsymbol{\eta}})$ 
5: while (stopping criterion is not met) do
6:   Derive  $B$  best heterogeneous co-clustering solutions using the diversification
     algorithm based on  $\hat{\boldsymbol{\eta}}_s$ 
7:   for each co-clustering solution  $(\boldsymbol{\chi}, \boldsymbol{\psi})_b$  do
8:     SOLVE Algorithm 3 and obtain  $(\boldsymbol{\eta}_s)_b$  vectors for  $(\boldsymbol{\chi}, \boldsymbol{\psi})_b$ 
9:      $z_b \leftarrow \sum_s J_s((\boldsymbol{\eta}_s)_b) + \rho((\boldsymbol{\eta})_b)$ 
10:   end for
11:    $\hat{b} = \arg \min_b \{z_b\}$ 
12:    $(\hat{\boldsymbol{\eta}}, \hat{\boldsymbol{\chi}}, \hat{\boldsymbol{\psi}}) \leftarrow (\boldsymbol{\eta}, \boldsymbol{\chi}, \boldsymbol{\psi})_{\hat{b}}$ 
13:   if  $z_{\hat{b}} < z_{OPT}$  then
14:      $OPT \leftarrow (\hat{\boldsymbol{\eta}}, \hat{\boldsymbol{\chi}}, \hat{\boldsymbol{\psi}})$ 
15:      $z_{OPT} \leftarrow z_{\hat{b}}$ 
16:   end if
17: end while

```

With these similarity matrices at hand, we used a cutting-plane algorithm for solving an MTMKL problem with given similarity matrices, aiming to derive more robust prediction models. Our computational experiments show that using these similarity matrices indeed improves predictive accuracy. In the Section 7.4, we give the results of this approach along with comparisons against other methods.

5.4 Discussion

With the collection of genomic information from a vast number of patients in recent decades, there have been many successful efforts in using this information in understanding cancer and ultimately improving cancer treatment. However, the high complexity of genomic data has limited its explanation power. In recent years, to

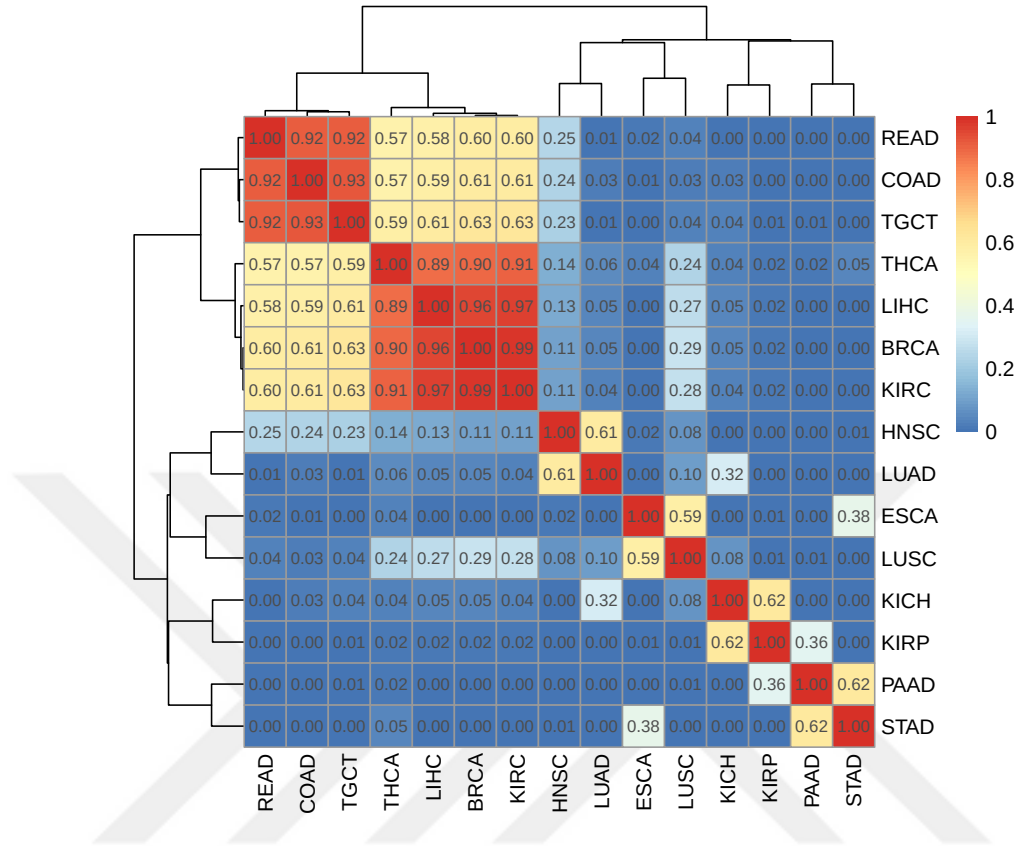


Figure 5.2: Similarity matrix obtained by solving 100 replications of the MTMKL problem with a co-clustering step on 15 TCGA cohorts.

fill this gap, researchers focused on identifying pathways that play a role in the progression of cancer.

In Chapter 3, we formulated a method for using the genomic information along with pathway information to improve prediction models for progression of cancer. To quantify the progression of cancer, we used the concept of staging and focused on predicting the stage of a cancer based on genomic data. By applying this method to various cancer cohorts, we were able to identify similarities in cancers based on their mechanisms. Namely, if similar pathways have the most predictive power for two cancers, it is plausible to infer some similarity between those two cancer types. However, in our previous study, the cohorts were examined separately and these separate analyses were aggregated afterwards, which did not benefit from the potential of learning prediction models conjointly. In this work, we improved

the prediction and knowledge extraction capabilities of our models using multitask learning and co-clustering of cohorts and pathways.

Multitask learning refers to enforcing some similarity between the prediction models for similar tasks. If, in fact, the tasks are similar, they will act as additional data points for each other, which leads to improved predictive accuracy. Our results show that multitask learning leads to significant improvements in prediction models based on genomic data and pathways. This considerable advantage of multitask learning requires either prior knowledge of the similarities between cancers or a way of identifying the similarities. Since such prior knowledge is not available, we have employed co-clustering of cancers and pathways to extract this information from the data.

In this context, co-clustering refers to building groups of cancers and pathways. Namely, the contribution of a pathway to the prediction model for a cohort will be penalized if they do not belong to the same group. In addition, if two cohorts are not in the same group, their prediction models are allowed to be different. This approach allows multitask learning between two tasks if the tasks are similar enough, and does not pick solutions with limited knowledge extraction capacity.

Since the problem of co-clustering the cohorts and pathways, and learning their prediction models was not addressed by the existing methods, we have formulated a new algorithm. This solution method based on the cutting-plane algorithm can be highly parallelized, allowing for large collection of datasets to be analyzed in a reasonable time.

After learning the clustering solutions of many replications, we aggregated these findings to devise a measure of similarity between the cohorts and pathways (similarity matrices) and fed these similarity matrices into a multitask learning algorithm, which is exact when the similarity matrices are given. Our results show significant improvements in terms of predictive power, further suggesting that the relations extracted here reflect important similarities between subsets of cancer types.

The cutting-plane algorithm proposed here can be extended towards building algorithms capable of finding the exact solution for joint clustering and multitask

learning in other domains.



Chapter 6

EFFICIENT MULTITASK MULTIPLE KERNEL LEARNING

Multitask multiple kernel learning algorithms combine the capabilities of incorporating different data sources into the prediction model and using the data from one task to improve the accuracy on others. However, these methods do not necessarily produce interpretable results. Restricting the solutions to the set of interpretable solutions increases the computational burden of the learning problem significantly, leading to computationally prohibitive run times for some important biomedical applications. That is why we propose a multitask multiple kernel learning formulation with clustering of tasks and develop a highly time-efficient solution approach for it. Our solution method is based on Benders decomposition and treating the clustering problem as finding a given number of tree structures in a graph, hence it is named the forest formulation. We use our method to discriminate early- and late-stage cancers using genomic data and gene sets, and compare our algorithm against two other algorithms. The two other algorithms are based on different approaches for linearization of the problem while all the algorithms make use of the cutting-plane method. Our results indicate that as the number of tasks and/or the number of desired clusters increase, the forest formulation becomes increasingly favorable in terms of computational performance.

6.1 Introduction

In biomedical applications, interpretability is essential for designing drugs, devising treatment strategies, and biomarker discovery for clinical decision-making (Rifai et al. 2006, Bhalla et al. 2017). Producing interpretable results often implies im-

posing restrictions on the models, which in turn, increases the computational effort required for solving the constrained optimization problems.

We consider the problem of clustering learning tasks where the tasks in the same cluster share weights for kernel functions, and the prediction models are developed using these kernel functions. Enforcing similar prediction models for some of the tasks makes this problem a multitask learning problem. Multitask learning refers to sharing data between the tasks, thereby strengthening the prediction models for tasks with limited data (Zhang 2014). In multitask learning, if prior knowledge about the similarities between the tasks is available, it can be used directly in the learning process, otherwise these similarities must also be inferred from the data. Clustering the tasks and learning the prediction models conjointly allows considering situations where a measure of similarity between the tasks is not available.

Kernel functions allow for building non-linear models and can be computationally helpful when the number of features is too large. MKL methods aim to generate prediction models with the best combination of a set of kernel functions (Gönen and Alpaydın 2011), thereby incorporating different representations of the data and obtaining interpretable solutions. Multitask MKL problem with clustering is therefore a bi-level optimization problem, where the higher-level problem decides on the clusters and the kernel weights, and the lower-level problems learn the optimal prediction models for each task based on the kernels decided on by the higher-level problem.

From an optimization perspective, this problem falls under the rubric of bi-level mixed integer nonlinear optimization problems, which can be intractable for large-scale instances of the problem. Despite the inherent difficulty of solving integer programs, recent theoretical advancements and hardware improvements have contributed to significant speed-ups in solving these optimization problems and have gained attention among the machine learning and data mining community (Bertsimas and King 2015, Bertsimas et al. 2017a). Therefore, in this chapter, we propose three mixed integer linear models based on transforming the bi-level problem into a collection of single-level problems by introducing appropriate variables and con-

straints into the problem. We refer to the first two formulations as variable linearization and objective linearization models. To efficiently solve these models, we employ cutting-plane algorithms. The last model, which we refer to as the forest formulation, is based on formulating the clustering problem as finding a number of distinct trees in a graph. We then propose a tailored BD algorithm for the forest formulation, which we refer to as the BDForest algorithm. We further accelerate this algorithm via symmetry breaking techniques, strong formulation, and Pareto-optimal cuts.

Multitask learning relies on commonalities and interdependence between related tasks (Gu et al. 2014, Pan et al. 2016, Tian et al. 2018). In many multitask learning applications, a measure of similarity between the tasks is known a priori. However, despite the huge literature on cancer, there is no widely accepted measure of similarity between different cancers, and the learning algorithm has to decide on this while building the prediction models. This choice to consider two cancers similar or not adds a number of binary variables to the problem. This increases the complexity of the problem significantly since for every set of integer values, there is a different optimal prediction model for each task. In Chapter 5, we introduced a heuristic method for solving the problem. To produce solutions that are less sensitive to the choice and calibration of algorithms and exploiting the recent advances in parallel computing technology, we aim to provide an efficient and exact solution method for similar problems.

The main contributions of this work is fourfold:

1. We introduce and formulate the *multitask multiple kernel learning problem with clustering* (MTMKLC) as a convex mixed integer optimization problem. We propose three mixed integer linear reformulations and outer-approximation (cutting-plane) algorithms for solving these optimization problems.
2. We develop the highly time-efficient *BDForest* algorithm based on BD for solving the MTMKLC problem. While a high number of clusters brings on additional difficulty to this problem, we show that the BDForest algorithm is capable of handling these problems efficiently independently of the number of

clusters.

3. Using the BDForest algorithm, we provide interpretable exact solutions to the problem of discriminating early- and late-stage cancers for 15 cancer cohorts based on genomic data of 5,547 patients and gene sets information. Previously there were no exact solution methods capable of solving the multitask multiple kernel learning problem with clustering.
4. We give comprehensive time comparisons between the algorithms under different parameter settings, which can be used for selecting the most proper algorithm for a given application.

The outline of this chapter is as follows. In Section 6.2, we give the formulation of the multitask multiple kernel learning problem with clustering. Section 6.3 presents two alternative linearization formulations for the MTMKLC problem and cutting-plane algorithms for solving instances of these formulations. We present a novel formulation for the MTMKLC problem by treating the clusters as trees in a graph in Section 6.4. Section 6.5 presents our solution method based on BD with several acceleration techniques for solving instances of the forest formulation. Section 6.6 concludes the chapter.

6.2 Problem Definition

Support vector machine (SVM) and its variants have widely been used for many classification tasks, because of their generalization ability, interpretability of their solutions, and their knowledge extraction capacities (Cortes and Vapnik 1995, Muller et al. 2001, Tian et al. 2013, Bai et al. 2014). Assume that T binary classification tasks are given, each task $s = 1, \dots, T$ with N_s data points of the form $\{(\mathbf{x}_{si}, y_{si})\}_{i=1}^{N_s}$, where $\mathbf{x}_{si} \in \mathbb{R}^{D_s}$ is the vector of the feature values for data point i and $y_{si} \in \{-1, 1\}$ is the class label of data point i . As shown in Section 3.1, we may formulate a

kernelized dual SVM problem as

$$J_s(\boldsymbol{\eta}_s) = \underset{\boldsymbol{\alpha}_s \in \mathcal{A}_s}{\text{maximize}} \left\{ \sum_{i=1}^{N_s} \alpha_{si} - \frac{1}{2} \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} \alpha_{si} \alpha_{sj} y_{si} y_{sj} \sum_{m=1}^P \eta_{sm} k_{sm}(\mathbf{x}_{si}, \mathbf{x}_{sj}) \right\}, \quad (6.1)$$

which we refer to as the dual SVM (DSVM) problem. In this formulation, P is the number of kernel functions, $\boldsymbol{\eta}_s$ is a given weight vector (profile) of the kernel functions, and $\mathcal{A}_s$ is the set of feasible support vector coefficients defined as

$$\mathcal{A}_s = \left\{ \boldsymbol{\alpha} \in [0, C]^{N_s} : \sum_{i=1}^{N_s} \alpha_{si} y_{si} = 0 \right\} \quad \forall s = 1, \dots, T.$$

Recall that MKL aims at choosing the optimal kernel weights $\boldsymbol{\eta}_s$ by minimizing the objective value in (6.1). We consider a generalization of MKL, where T tasks are divided into K subgroups (clusters), and the tasks assigned to each cluster are required to have prediction models based on the same kernel weights. Each cluster $l \in \{1, \dots, K\}$ consists of a number of tasks and has its own kernel profile denoted by $\boldsymbol{\eta}_l$. Every task $s \in \{1, \dots, T\}$ has to be assigned to one cluster and use the kernel profile of its assigned cluster to learn a prediction model. The kernel profile of task s is therefore computed by $\boldsymbol{\eta}_s = \sum_{l=1}^K z_{sl} \boldsymbol{\eta}_l$, where z_{sl} is a binary variable determining if task s is assigned to cluster l . Using these notations, we formulate the MTMKLC problem as

$$\underset{\{\boldsymbol{\eta}_l \in \Delta\}_{l=1}^K, \mathbf{Z} \in \mathcal{C}}{\text{minimize}} \sum_{s=1}^T J_s \left(\sum_{l=1}^K z_{sl} \boldsymbol{\eta}_l \right), \quad (6.2)$$

where

$$\Delta = \left\{ \boldsymbol{\eta} \in \mathbb{R}_+^P : \sum_{m=1}^P \eta_m = 1 \right\}$$

is the unit simplex, and

$$\mathcal{C} = \left\{ \mathbf{Z} \in \{0, 1\}^{T \times K} : \sum_{l=1}^K z_{sl} = 1 \quad \forall s, \quad \sum_{s=1}^T z_{sl} \geq 1 \quad \forall l \right\}$$

is the set of feasible clusters.

6.3 Linearization Models

The MTMKLC problem (6.2) involves two types of non-linearity: non-linear functions $J_s(\cdot)$ and bilinear terms of the form $z_{sl}\boldsymbol{\eta}_l$, where binary variables $\mathbf{Z}$ are multiplied by continuous variables $\{\boldsymbol{\eta}_l\}_{l=1}^K$. In the following, we show how to linearize and solve problem (6.2) using cutting-plane algorithms. For notational convenience, for a given task s and respective support vector coefficients $\boldsymbol{\alpha}_s \in \mathcal{A}_s$, we define notations $|\boldsymbol{\alpha}_s| = \sum_{i=1}^{N_s} \alpha_{si}$ and

$$\kappa_{sm}(\boldsymbol{\alpha}_s) = \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} \alpha_{si} \alpha_{sj} y_{si} y_{sj} k_{sm}(\mathbf{x}_{si}, \mathbf{x}_{sj}).$$

6.3.1 Variable Linearization

Let $\mathcal{J}_s$ be an over-estimator of $J_s(\cdot)$. For each arbitrary kernel weight profile $\boldsymbol{\eta}_l$, since $J_s(\boldsymbol{\eta}_l)$ is the optimal value of problem (6.1), we have

$$J_s(\boldsymbol{\eta}_l) \geq |\boldsymbol{\alpha}_s| - \frac{1}{2} \sum_{m=1}^P \kappa_{sm}(\boldsymbol{\alpha}_s) \eta_{lm} \quad \forall \boldsymbol{\alpha}_s \in \mathcal{A}_s.$$

Therefore, problem (6.2) can be equivalently formulated as the following SIMILP:

$$\text{minimize } \sum_{s=1}^T \mathcal{J}_s \tag{6.3}$$

$$\text{with respect to } \{\boldsymbol{\eta}_l \in \Delta\}_{l=1}^K, \mathbf{Z} \in \mathcal{C}, \boldsymbol{\mathcal{J}} \in \mathbb{R}_+^T \tag{6.4}$$

$$\text{subject to } \mathcal{J}_s + \frac{1}{2} \sum_{m=1}^P \kappa_{sm}(\boldsymbol{\alpha}_s) \sum_{l=1}^K z_{sl} \eta_{lm} \geq |\boldsymbol{\alpha}_s| \quad \forall \boldsymbol{\alpha}_s \in \mathcal{A}_s. \tag{6.5}$$

To linearize the bilinear terms $z_{sl}\eta_{lm}$ in (6.5), define vector $\boldsymbol{\eta}_{sl} = z_{sl}\boldsymbol{\eta}_l$. Observe that since $\mathbf{Z}$ is binary, $z_{sl} = 0 \rightarrow \boldsymbol{\eta}_{sl} = \mathbf{0}$ and $z_{sl} = 1 \rightarrow \boldsymbol{\eta}_{sl} = \boldsymbol{\eta}_l$. These relationships

can be expressed in linear form using the following constraints:

$$\eta_{slm} \leq z_{sl} \quad \forall(s, l, m) \quad (6.6)$$

$$\eta_{slm} \leq \eta_{lm} \quad \forall(s, l, m) \quad (6.7)$$

$$\eta_{slm} \geq \eta_{lm} + z_{sl} - 1 \quad \forall(s, l, m). \quad (6.8)$$

In the following, we show how some of these constraints can be replaced with stronger constraints or can be eliminated entirely. First, since $\boldsymbol{\eta}_l \in \Delta$, we have $\sum_{m=1}^P \eta_{lm} = 1$, therefore (6.6) can be replaced with the following stronger compact constraint:

$$\sum_{m=1}^P \eta_{slm} = z_{sl} \quad \forall(s, l). \quad (6.9)$$

Second, note that, whenever $z_{sl} = 1$, (6.9) implies that $\sum_{m=1}^P \eta_{slm} = \sum_{m=1}^P \eta_{lm} = 1$. This property in conjunction with (6.7) imply that whenever $z_{sl} = 1$, we have to have $\boldsymbol{\eta}_{sl} = \boldsymbol{\eta}_l$. Thus, constraint (6.8) can be safely eliminated. Consequently, problem (6.3)–(6.5) is formulated as the following SIMILP:

$$\begin{aligned} & \text{minimize } \sum_{s=1}^T \mathcal{J}_s \\ & \text{with respect to } \{\boldsymbol{\eta}_l \in \Delta\}_{l=1}^K, \{\boldsymbol{\eta}_{sl} \in \Delta\}_{s=1, l=1}^{T, K}, \mathbf{Z} \in \mathcal{C}, \boldsymbol{\mathcal{J}} \in \mathbb{R}_+^T \\ & \text{subject to } \eta_{slm} \leq \eta_{lm} \quad \forall(s, l, m) \\ & \quad \sum_{m=1}^P \eta_{slm} = z_{sl} \quad \forall(s, l) \\ & \quad \mathcal{J}_s + \frac{1}{2} \sum_{m=1}^P \kappa_{sm}(\boldsymbol{\alpha}_s) \sum_{l=1}^K \eta_{slm} \geq |\boldsymbol{\alpha}_s| \quad \forall \boldsymbol{\alpha}_s \in \mathcal{A}_s, \end{aligned} \quad (6.10)$$

which we refer to as the variable linearization master problem (VLMP). This problem can be solved using a cutting-plane method, where $\mathbf{Z}$, $\{\boldsymbol{\eta}_l \in \Delta\}_{l=1}^K$, and $\{\boldsymbol{\eta}_{sl} \in \Delta\}_{s=1, l=1}^{T, K}$ are the master problem variables, and the cuts of the form (6.10) are generated in the course of the algorithm by solving respective dual SVM problems. Note that, for a fixed value of $\mathbf{Z}$ and $\{\boldsymbol{\eta}_l \in \Delta\}_{l=1}^K$, the problem can be decomposed

by each s , therefore the dual SVM problems can be solved in parallel. A pseudo-code for the proposed cutting-plane algorithm is detailed in Algorithm 5, in which convergence is controlled by the parameters t_{max} (maximum number of iterations) and ζ (optimality gap threshold).

Algorithm 5 Cutting-plane algorithm for solving VLMP

```

1:  $UB \leftarrow +\infty, LB \leftarrow -\infty, t \leftarrow 1, \mathcal{A}_s \leftarrow \emptyset$ 
2: Terminate  $\leftarrow$  FALSE
3: while Terminate = FALSE do
4:   SOLVE VLMP and obtain  $(\{\boldsymbol{\eta}_l\}_{l=1}^K, \mathbf{Z})$  and  $\mathcal{J}$ 
5:    $LB \leftarrow \sum_{s=1}^T \mathcal{J}_s$ 
6:   for  $s = 1, \dots, T$  do
7:     SOLVE DSVM (6.1) with  $\hat{\boldsymbol{\eta}}_s = \sum_{l=1}^K z_{sl} \boldsymbol{\eta}_l$  to obtain optimal value  $v_s = J_s(\hat{\boldsymbol{\eta}}_s)$  and optimal solution  $\boldsymbol{\alpha}_s$ .
8:      $\mathcal{A}_s \leftarrow \mathcal{A}_s \cup \{\boldsymbol{\alpha}_s\}$  (Add a cut of the form (6.10) to VLMP)
9:   end for
10:   $UB \leftarrow \min\{UB, \sum_{s=1}^T v_s\}$ 
11:  Gap  $\leftarrow (UB - LB)/LB$ 
12:  if Gap  $\leq \zeta$  or  $t \geq t_{max}$  then
13:    Terminate  $\leftarrow$  TRUE
14:  end if
15:   $t \leftarrow t + 1$ 
16: end while

```

6.3.2 Objective Function Linearization

We now present another way of linearizing problem (6.2). Since $\mathbf{Z}$ is binary, the constraint $\sum_{l=1}^K z_{sl} = 1$ implies that

$$\sum_{s=1}^T J_s \left(\sum_{l=1}^K z_{sl} \boldsymbol{\eta}_l \right) = \sum_{s=1}^T \sum_{l=1}^K z_{sl} J_s(\boldsymbol{\eta}_l),$$

which means that, instead of multiplying the clustering variables (i.e., $\mathbf{Z}$) with the kernel weights (i.e., $\{\boldsymbol{\eta}_l\}_{l=1}^K$), we can multiply the clustering variables with the J functions. To linearize the term $z_{sl} J_s(\boldsymbol{\eta}_l)$, we define variable $\mathcal{J}_{sl}$ to be equal to 0

when $z_{sl} = 0$ and equal to $J_s(\cdot)$ when $z_{sl} = 1$. Moreover, let $\bar{J}_s$ be a valid upper bound on the value of $J_s(\cdot)$. These values are problem specific and can be calculated for each problem instance. The relationship between $\mathbf{Z}$, $\mathbf{J}$ and $\mathcal{J}$ can be stated in linear form as

$$\mathcal{J}_{sl} \leq z_{sl} \bar{J}_s \quad \forall (s, l) \quad (6.11)$$

$$\mathcal{J}_{sl} \leq J_s(\cdot) \quad \forall (s, l) \quad (6.12)$$

$$\mathcal{J}_{sl} \geq J_s(\cdot) + (z_{sl} - 1) \bar{J}_s \quad \forall (s, l)$$

$$\mathcal{J}_{sl} \geq 0 \quad \forall (s, l).$$

Since problem (6.2) minimizes $z_{sl} J_s(\cdot)$ (or equivalently $\mathcal{J}_{sl}$), constraints (6.11) and (6.12) can be discarded. The model with linearized objective function can be formulated as

$$\begin{aligned} & \text{minimize} \quad \sum_{l=1}^K \sum_{s=1}^T \mathcal{J}_{sl} \\ & \text{with respect to} \quad \{\boldsymbol{\eta}_l \in \Delta\}_{l=1}^K, \mathbf{Z} \in \mathcal{C}, \{\mathcal{J}_s \in \mathbb{R}_+^K\}_{s=1}^T \\ & \text{subject to} \quad \mathcal{J}_{sl} \geq J_s(\boldsymbol{\eta}_l) + (z_{sl} - 1) \bar{J}_s \quad \forall (s, l). \end{aligned} \quad (6.13)$$

Recall that $J_s(\boldsymbol{\eta}_l)$ is equal to the optimal value of the dual SVM problem (6.1) for any arbitrary $\boldsymbol{\eta}_l$. Hence, we have

$$J_s(\boldsymbol{\eta}_l) \geq |\boldsymbol{\alpha}_s| - \frac{1}{2} \sum_{m=1}^P \kappa_{sm}(\boldsymbol{\alpha}_s) \eta_{lm} \quad \forall \boldsymbol{\alpha}_s \in \mathcal{A}_s. \quad (6.14)$$

Combining (6.13) and (6.14), we obtain the following SIMILP:

$$\begin{aligned}
& \text{minimize} && \sum_{l=1}^K \sum_{s=1}^T \mathcal{J}_{sl} \\
& \text{with respect to} && \{\boldsymbol{\eta}_l \in \Delta\}_{l=1}^K, \mathbf{Z} \in \mathcal{C}, \{\mathcal{J}_s \in \mathbb{R}_+^K\}_{s=1}^T \\
& \text{subject to} && \mathcal{J}_{sl} + \frac{1}{2} \sum_{m=1}^P \eta_{lm} \kappa_{sm}(\boldsymbol{\alpha}_s) \geq \\
& && |\boldsymbol{\alpha}_s| + (z_{sl} - 1) \bar{\mathcal{J}}_s \quad \forall (s, l, \boldsymbol{\alpha}_s \in \mathcal{A}_s),
\end{aligned} \tag{6.15}$$

which we refer to as the objective function linearization master problem (OLMP).

Next, we explain how feasible values for parameters $\bar{\mathcal{J}}_s$ can be found.

Proposition 3. *For each task s , define $\mathcal{N}_s^+$ and $\mathcal{N}_s^-$ as the set of data points with positive and negative labels, respectively. Then, $\bar{\mathcal{J}}_s = 2C \min\{|\mathcal{N}_s^+|, |\mathcal{N}_s^-|\}$ is a valid upper bound on $J_s(\boldsymbol{\eta})$ for any arbitrary $\boldsymbol{\eta} \in \Delta$.*

Proof. For arbitrary $\boldsymbol{\eta}$, let $\boldsymbol{\alpha}_s(\boldsymbol{\eta}) \in \mathcal{A}_s$ be the optimal solution of dual SVM problem (6.1), that is

$$J_s(\boldsymbol{\eta}) = |\boldsymbol{\alpha}_s(\boldsymbol{\eta})| - \frac{1}{2} \sum_{m=1}^P \kappa_{sm}(\boldsymbol{\alpha}_s(\boldsymbol{\eta})) \eta_m.$$

Since the kernel functions κ_{sm} are positive semi-definite, $\kappa_{sm}(\boldsymbol{\alpha}_s) \geq 0$ for each m, s and $\boldsymbol{\alpha}_s$, implying

$$J_s(\boldsymbol{\eta}) \leq |\boldsymbol{\alpha}_s(\boldsymbol{\eta})|.$$

Additionally, since $\boldsymbol{\alpha}_s(\boldsymbol{\eta}) \in \mathcal{A}_s$, we have $\sum_{i \in \mathcal{N}_s^-} \alpha_{si}(\boldsymbol{\eta}) = \sum_{i \in \mathcal{N}_s^+} \alpha_{si}(\boldsymbol{\eta})$, therefore

$$|\boldsymbol{\alpha}_s(\boldsymbol{\eta})| = \sum_{i=1}^{N_s} \alpha_{si}(\boldsymbol{\eta}) = \begin{cases} 2 \sum_{i \in \mathcal{N}_s^-} \alpha_{si}(\boldsymbol{\eta}) \leq 2C |\mathcal{N}_s^-| \\ 2 \sum_{i \in \mathcal{N}_s^+} \alpha_{si}(\boldsymbol{\eta}) \leq 2C |\mathcal{N}_s^+| \end{cases},$$

which yields $J_s(\boldsymbol{\eta}) \leq |\boldsymbol{\alpha}_s(\boldsymbol{\eta})| \leq 2C \min\{|\mathcal{N}_s^+|, |\mathcal{N}_s^-|\}$. $\square$

Proposition 4. *For each task s and kernel index m , define $J_{sm}^* = J_s(\mathbf{e}_m)$ where $\mathbf{e}_m \in \mathbb{R}^P$ is the m^{th} basic vector of $\mathbb{R}^P$. Then, $\bar{\mathcal{J}}_s = \max\{J_{s1}^*, J_{s2}^*, \dots, J_{sP}^*\}$ is a valid upper bound on $J_s(\boldsymbol{\eta})$ for any arbitrary $\boldsymbol{\eta} \in \Delta$.*

Proof. Define $J_s^* = \max_{\boldsymbol{\eta} \in \Delta} J_s(\boldsymbol{\eta})$. We show that $\bar{J}_{2s} = J_s^*$. Note that J_s^* can be rewritten as

$$\max_{\boldsymbol{\eta} \in \Delta} \max_{\boldsymbol{\alpha} \in \mathcal{A}_s} \sum_{i=1}^{N_s} \alpha_i - \frac{1}{2} \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} \alpha_i \alpha_j y_{si} y_{sj} \sum_{m=1}^P \eta_m k_{sm}(\mathbf{x}_{si}, \mathbf{x}_{sj}). \quad (6.16)$$

Since (6.16) is a joint maximization on both $\boldsymbol{\eta}$ and $\boldsymbol{\alpha}$, we can exchange the order of maximizations and rearrange (6.16) as

$$J_s^* = \max_{\boldsymbol{\alpha} \in \mathcal{A}_s} \hat{J}_s(\boldsymbol{\alpha}), \quad (6.17)$$

where

$$\hat{J}_s(\boldsymbol{\alpha}) = \max_{\boldsymbol{\eta} \in \Delta} \sum_{i=1}^{N_s} \alpha_i - \frac{1}{2} \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} \alpha_i \alpha_j y_{si} y_{sj} \sum_{m=1}^P \eta_m k_{sm}(\mathbf{x}_{si}, \mathbf{x}_{sj}),$$

which is a linear program in $\boldsymbol{\eta}$. Hence, for any arbitrary $\boldsymbol{\alpha}$, the $\hat{J}_s(\boldsymbol{\alpha})$ is attained at one of the extreme points of Δ , which are the basic vectors of $\mathbb{R}^P$. Therefore, we can compute $\hat{J}_s(\boldsymbol{\alpha})$ via

$$\hat{J}_s(\boldsymbol{\alpha}) = \max_m \sum_{i=1}^{N_s} \alpha_i - \frac{1}{2} \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} \alpha_i \alpha_j y_{si} y_{sj} k_{sm}(\mathbf{x}_{si}, \mathbf{x}_{sj}).$$

Thus, (6.17) is equivalent to

$$\begin{aligned} J_s^* &= \max_{\boldsymbol{\alpha} \in \mathcal{A}_s} \max_m \sum_{i=1}^{N_s} \alpha_i - \frac{1}{2} \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} \alpha_i \alpha_j y_{si} y_{sj} k_{sm}(\mathbf{x}_{si}, \mathbf{x}_{sj}) \\ &= \max_m \max_{\boldsymbol{\alpha} \in \mathcal{A}_s} \sum_{i=1}^{N_s} \alpha_i - \frac{1}{2} \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} \alpha_i \alpha_j y_{si} y_{sj} k_{sm}(\mathbf{x}_{si}, \mathbf{x}_{sj}) \\ &= \max_m J_{sm}^* = \bar{J}_{2s}. \end{aligned}$$

Consequently, since $\bar{J}_{2s} = \max_{\boldsymbol{\eta} \in \Delta} J_s(\boldsymbol{\eta})$, $\bar{J}_{2s}$ is a valid upper bound on $J_s(\boldsymbol{\eta})$ for any arbitrary $\boldsymbol{\eta}$. $\square$

Corollary 1. $\bar{J}_{2s}$ is a tight upper bound on $J_s(\cdot)$. There is no other value $\bar{J}_s^* < \bar{J}_{2s}$

such that $J_s(\boldsymbol{\eta}) \leq \bar{J}_s^*$ for each $\boldsymbol{\eta} \in \Delta$.

Mixed integer linear program (MILP) solvers tend to rely on the continuous relaxation of subproblems, thus strength of a MILP involving big-M inequalities depends on the tightness of the big-M values. Since $\bar{J}_{2s} \leq \bar{J}_{1s}$, setting $\bar{J}_s = \bar{J}_{2s}$ in OLMP provides a stronger formulation. However, obtaining $\bar{J}_{2s}$ values requires solving $T \times P$ dual SVM problems which can be computationally expensive, whereas $\bar{J}_{1s}$ can be obtained instantly.

As per VLMP, OLMP can be solved using a cutting-plane method, where $\mathbf{Z}$ and $\{\boldsymbol{\eta}_l\}_{l=1}^K$ are the master problem variables, and the cuts of the form (6.15) are generated by solving respective dual SVM problems. The algorithm is slightly different from Algorithm 5 and is detailed in Algorithm 6.

Algorithm 6 Cutting-plane algorithm for solving OLMP

```

1: Compute  $\bar{J}_s$  values using Proposition 3 or 4.
2:  $UB \leftarrow +\infty, LB \leftarrow -\infty, t \leftarrow 1, \mathcal{A}_s \leftarrow \emptyset$ 
3: Terminate  $\leftarrow$  FALSE
4: while Terminate = FALSE do
5:   SOLVE OLMP and obtain  $(\{\boldsymbol{\eta}_l\}_{l=1}^K, \mathbf{Z})$  and  $\mathcal{J}$ 
6:    $LB \leftarrow \sum_{s=1}^T \sum_{l=1}^K \mathcal{J}_{sl}$ 
7:   for  $s = 1, \dots, T$  do
8:     SOLVE DSVM (6.1) with  $\hat{\boldsymbol{\eta}}_s = \sum_{l=1}^K z_{sl} \boldsymbol{\eta}_l$  to obtain optimal value  $v_s = J_s(\hat{\boldsymbol{\eta}}_s)$  and optimal solution  $\boldsymbol{\alpha}_s$ .
9:      $\mathcal{A}_s \leftarrow \mathcal{A}_s \cup \{\boldsymbol{\alpha}_s\}$  (Add  $K$  cuts of the form (6.15) to OLMP)
10:  end for
11:   $UB \leftarrow \min\{UB, \sum_s v_s\}$ 
12:   $\text{Gap} \leftarrow (UB - LB)/LB$ 
13:  if  $\text{Gap} \leq \zeta$  or  $t \geq t_{max}$  then
14:    Terminate  $\leftarrow$  TRUE
15:  end if
16:   $t \leftarrow t + 1$ 
17: end while

```

6.4 Forest Formulation

The solution spaces of the linearized models introduced in Section 6.3 are highly symmetric and grow exponentially by the number of clusters (i.e., K). These drawbacks limit applicability of these models to small instances of the MTMKLC problem. To mitigate this dependency and to eliminate the symmetric solutions, we introduce a novel way of enforcing group assignments by modelling each group as a tree. Let u_{sq} be a binary variable determining whether there exists an edge between nodes s and q (i.e., tasks s and q are assigned to the same group). A forest is an undirected graph whose connected components are trees. Figure 6.1 illustrates a toy example in which, with 8 tasks ($T = 8$) and 3 clusters ($K = 3$), the clusters have been formed using three tree structures with a total of $T - K = 8 - 3 = 5$ edges. In general, a forest of T nodes and K components must contain exactly $T - K$ edges; hence, to correctly define a forest of K components using $\mathbf{U}$ variables, we must satisfy two conditions:

1. The graph must contain exactly $T - K$ edges,
2. The graph must be acyclic.

To meet the first condition, binary $\mathbf{U}$ must satisfy

$$\sum_{s=1}^{T-1} \sum_{q=s+1}^T u_{sq} = T - K, \quad (6.18)$$

where we have assumed that $s < q$ because the graph is undirected. Additionally, to satisfy the second condition, we must ensure that no subset of nodes contains a subtour:

$$\sum_{s \in \delta} \sum_{q \in \delta: s < q} u_{sq} \leq |\delta| - 1 \quad \forall \delta \subset \{1, \dots, T\}: |\delta| \leq T - K. \quad (6.19)$$

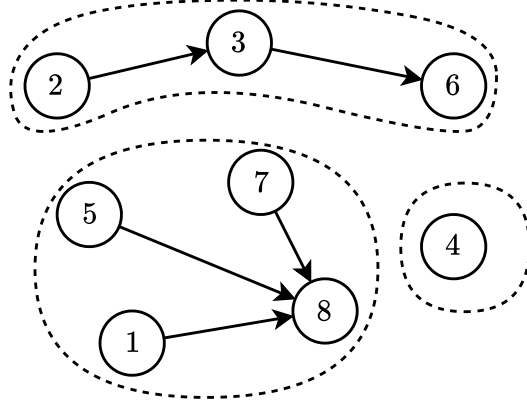


Figure 6.1: A toy example for clustering the tasks as a collection of trees.

Let $\mathcal{F}$ be the set of binary solutions satisfying these two conditions, that is

$$\mathcal{F} = \{\mathbf{U} \in \{0, 1\}^{T \times T} : (6.18), (6.19)\}. \quad (6.20)$$

Using these variables and the forest defining constraints, we can reformulate problem (6.2) as

$$\begin{aligned} & \text{minimize} \quad \sum_{s=1}^T J_s(\boldsymbol{\eta}_s) \\ & \text{with respect to} \quad \mathbf{U} \in \mathcal{F}, \quad \{\boldsymbol{\eta}_s \in \Delta\}_{s=1}^T \\ & \text{subject to} \quad \eta_{sm} - \eta_{qm} \leq 1 - u_{sq} \quad \forall (s, q, m) : s < q. \end{aligned} \quad (6.21)$$

In this formulation, the constraints ensure that when tasks s and q are connected, they share the same kernel weights. Note that by these constraints, all nodes that belong to the same connected component (i.e., cluster) will share the same kernel weights, thus this model correctly formulates the underlying MTMKLC problem.

An important observation to be made concerns the number of subtour elimination constraints of the form (6.19), which grows exponentially by the number of tasks. Additionally, each subset of nodes corresponds to multiple spanning trees. Therefore, these spanning trees can be viewed as symmetric solutions. Symmetric solutions put

unnecessary computational burden on the solution algorithm, and eliminating such solutions reduces the size of the solution space thus helping the algorithm prune the suboptimal solutions more efficiently. In the following, we show how subtour elimination constraints can be replaced with symmetry breaking constraint.

6.4.1 Symmetry Breaking

We eliminate symmetric solutions by restricting the trees to linear chains (i.e., requiring the in-degree and out-degree of nodes to be at most one):

$$\sum_{q=s+1}^T u_{sq} \leq 1 \quad \forall s, \quad \sum_{q=1}^{s-1} u_{qs} \leq 1 \quad \forall s. \quad (6.22)$$

Note that since we restrict the direction of edge (s, q) to $s < q$, symmetry breaking constraints (6.22) establish a one-to-one mapping between subsets of nodes and the respective chains. Figure 6.2 depicts the effect of these constraints. Additionally, as shown in Proposition 5, employing symmetry breaking constraints not only reduces the size of solution space by eliminating symmetric solutions, but also yields feasible binary solutions. Hence, we may replace subtour elimination constraints (6.19) with symmetry breaking constraints (6.22), which results in the following definition of $\mathcal{F}$:

$$\mathcal{F} = \{\mathbf{U} \in \{0, 1\}^{T \times T} : (6.18), (6.22)\}. \quad (6.23)$$

Proposition 5. *Symmetry breaking constraints eliminate subtours.*

Proof. By contradiction, assume that a subtour of length l exists, and assume that nodes $\{s_1, s_2, \dots, s_l\}$ form this cycle. Without loss of generality assume that $s_1 < s_2 < \dots < s_l$. Since an edge between nodes s and q exists only if $s < q$, the edges that form the cycle must be $\{(s_1, s_2), (s_2, s_3), \dots, (s_{l-1}, s_l), (s_1, s_l)\}$. But this would imply that out-degree of s_1 (and in-degree of s_l) is greater than 2, contradicting with (6.22). $\square$

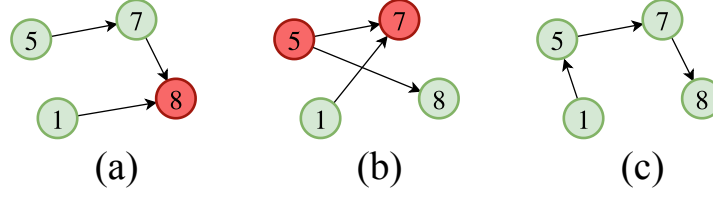


Figure 6.2: Symmetry breaking constraints reduce the number of multiple solutions. The three distinct solutions given in (a)–(c) refer to the same cluster; however, only (c) satisfies the symmetry breaking constraints.

6.4.2 Cutting-plane Algorithm for Forest Formulation

As per VLMP and OLMP, problem (6.21) can be expressed as an SIMILP of the form:

$$\begin{aligned}
 & \text{minimize} \quad \sum_{s=1}^T \mathcal{J}_s \\
 & \text{with respect to} \quad \mathbf{U} \in \mathcal{F}, \{\boldsymbol{\eta}_s \in \Delta\}_{s=1}^T, \boldsymbol{\mathcal{J}} \in \mathbb{R}_+^T \\
 & \text{subject to} \quad \eta_{sm} - \eta_{qm} \leq 1 - u_{sq} \quad \forall (s, q, m): s < q \\
 & \quad \quad \quad \mathcal{J}_s + \frac{1}{2} \sum_{m=1}^P \eta_{sm} \kappa_{sm}(\boldsymbol{\alpha}_s) \geq |\boldsymbol{\alpha}_s| \quad \forall \boldsymbol{\alpha}_s \in \mathcal{A}_s,
 \end{aligned} \tag{6.24}$$

which we refer to as the forest master problem (FMP). We solve FMP by relaxing the constraints (6.24) and producing them as needed using the iterative procedure described in Algorithm 7, in which FMP_t is FMP at iteration t with $\mathcal{A}_{st}$ in place of $\mathcal{A}_s$.

6.5 BDForest: Benders Decomposition for Forest Formulation

The bottleneck in Algorithm 7 is solving the integer program FMP_t at each iteration t of this algorithm. Benders Decomposition (BD) is a decomposition technique for solving large-scale MILPs (Geoffrion 1972). This method lends itself to problems where, when a set of complicating variables are temporarily fixed, the remaining

Algorithm 7 Cutting-plane algorithm for forest formulation

```

1:  $UB \leftarrow +\infty, LB \leftarrow -\infty, t \leftarrow 1, \mathcal{A}_{st} \leftarrow \emptyset$ 
2: Terminate  $\leftarrow \mathbf{FALSE}$ 
3: while Terminate = FALSE do
4:   SOLVE FMPt; obtain  $(\mathbf{U}, \{\boldsymbol{\eta}_s\}_{s=1}^T)$  and  $\mathcal{J}$ 
5:    $LB \leftarrow \sum_s \mathcal{J}_s$ 
6:   for  $s = 1, \dots, T$  do
7:     SOLVE DSVM (6.1) using  $\boldsymbol{\eta}_s$ ; obtain  $\boldsymbol{\alpha}_s$  and  $J_s(\boldsymbol{\eta}_s)$ 
8:      $\mathcal{A}_{s,t+1} \leftarrow \mathcal{A}_{s,t} \cup \{\boldsymbol{\alpha}_s\}$ 
9:   end for
10:   $UB \leftarrow \min\{UB, \sum_s J_s(\boldsymbol{\eta}_s)\}$ 
11:  Gap  $\leftarrow \frac{UB - LB}{LB}$ 
12:  if Gap  $\leq \zeta$  or  $t \geq t_{max}$  then
13:    Terminate  $\leftarrow \mathbf{TRUE}$ 
14:  end if
15:   $t \leftarrow t + 1$ 
16: end while

```

problem is relatively easier to solve. With $\mathbf{U}$ fixed at $\hat{\mathbf{U}} \in \mathcal{F}$, FMP_t reduces to

$$\text{minimize } \sum_{s=1}^T \mathcal{J}_s$$

$$\text{with respect to } \{\boldsymbol{\eta}_s \in \mathbb{R}_+^P\}_{s=1}^T, \mathcal{J} \in \mathbb{R}_+^T$$

$$\text{subject to } \sum_{m=1}^P \eta_{sm} = 1 \quad \forall s \quad (6.25)$$

$$\eta_{qm} - \eta_{sm} \geq \hat{u}_{sq} - 1 \quad \forall (s, q, m): s < q \quad (6.26)$$

$$\mathcal{J}_s + \frac{1}{2} \sum_{m=1}^P \eta_{sm} \kappa_{sm}(\boldsymbol{\alpha}_s) \geq |\boldsymbol{\alpha}_s| \quad \forall \boldsymbol{\alpha}_s \in \mathcal{A}_{st}, \quad (6.27)$$

which we refer to as the primal subproblem (PSP). Let θ_s , λ_{sqm} and $\beta_{s\boldsymbol{\alpha}_s}$ be the dual variables associated with constraints (6.25), (6.26) and (6.27), respectively. The

dual subproblem (DSP) is formulated as

$$\begin{aligned}
& \text{maximize} \quad \sum_{s=1}^T \left[\theta_s + \sum_{\alpha_s \in \mathcal{A}_{st}} \beta_s \alpha_s |\alpha_s| + \sum_{q=s+1}^T \sum_{m=1}^P \lambda_{sqm} (\hat{u}_{sq} - 1) \right] \\
& \text{with respect to } \boldsymbol{\theta} \in \mathbb{R}^T, \{\boldsymbol{\lambda}_m \in \mathbb{R}_+^{T \times T}\}_{m=1}^P, \{\boldsymbol{\beta}_s \in \mathbb{R}_+^{|\mathcal{A}_{st}|}\}_{s=1}^T \\
& \text{subject to} \quad \sum_{\alpha_s \in \mathcal{A}_{st}} \beta_s \alpha_s \leq 1 \quad \forall s \\
& \quad \theta_s + \sum_{q=1}^{s-1} \lambda_{qsm} - \sum_{q=s+1}^T \lambda_{sqm} \\
& \quad \quad + \frac{1}{2} \sum_{\alpha_s \in \mathcal{A}_{st}} \beta_s \alpha_s \kappa_{sm}(\alpha_s) \leq 0 \quad \forall (s, m). \tag{6.28}
\end{aligned}$$

Let Θ_t denote the set of extreme points of DSP. Note that $\hat{\mathbf{U}}$ only appears in the objective function of DSP and Θ_t does not depend on $\hat{\mathbf{U}}$. Additionally, since DSP is a linear program, its optimal solution is attained at one of the points in Θ_t . Therefore, FMP_t can be equivalently restated as

$$\begin{aligned}
& \text{minimize} \quad \mu \\
& \text{with respect to } \mu \in \mathbb{R}, \mathbf{U} \in \mathcal{F} \\
& \text{subject to } \mu \geq \mathcal{B}_t(\mathbf{U} | \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\lambda}) \quad \forall (\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\lambda}) \in \Theta_t, \tag{6.30}
\end{aligned}$$

which is known as the Benders master problem (BMP). In this formulation, constraints of the form (6.30) are called the Benders optimality cuts defined as

$$\mathcal{B}_t(\mathbf{U} | \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\lambda}) = \sum_{s=1}^T \left[\theta_s + \sum_{\alpha_s \in \mathcal{A}_{st}} \beta_s \alpha_s |\alpha_s| + \sum_{q=s+1}^T \sum_{m=1}^P \lambda_{sqm} (u_{sq} - 1) \right].$$

BMP_t is an equivalent formulation for FMP_t, but contains exponentially many constraints of the form (6.30). However, only a few of these constraints will be active at the optimal solution of BMP_t. In BD algorithm, instead of solving BMP_t,

a relaxation of this problem (RBMP) is solved by replacing Θ_t with a subset $\tilde{\Theta} \subset \Theta_t$:

$$\begin{aligned} & \text{minimize } \mu \\ & \text{with respect to } \mu \in \mathbb{R}, \mathbf{U} \in \mathcal{F} \\ & \text{subject to } \mu \geq \mathcal{B}_t(\mathbf{U}|\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\lambda}) \quad \forall (\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\lambda}) \in \tilde{\Theta}, \end{aligned}$$

which produces a trial solution $\hat{\mathbf{U}}$. Passing $\hat{\mathbf{U}}$ to DSP, we obtain a Benders cut of the form (6.30), which is then added to RBMP to strengthen the relaxation. As described in Algorithm 8, we repeat this procedure until the optimal solution to BMP_t (or FMP_t) is attained.

Algorithm 8 BD algorithm for solving FMP_t

```

1:  $UB_t \leftarrow +\infty, LB_t \leftarrow -\infty$ 
2:  $\tilde{\Theta} \leftarrow \emptyset, (\mathbf{U}^*, \{\boldsymbol{\eta}_s^*\}_{s=1}^T, \mathcal{J}^*) \leftarrow \emptyset$ 
3:  $\tau \leftarrow 1, \text{Terminate} \leftarrow \mathbf{FALSE}$ 
4: while  $\text{Terminate} = \mathbf{FALSE}$  do
5:   SOLVE  $\text{RBMP}_t(\tilde{\Theta})$ ; obtain  $(\hat{\mu}, \hat{\mathbf{U}})$ 
6:    $LB_t \leftarrow \hat{\mu}$ 
7:   SOLVE  $\text{PSP}_t$  using  $\hat{\mathbf{U}}$ ; obtain primal solution  $(\{\hat{\boldsymbol{\eta}}_s\}_{s=1}^T, \hat{\mathcal{J}})$  and dual solution  $(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\lambda}})$ 
8:    $\tilde{\Theta} \leftarrow \tilde{\Theta} \cup \{(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\lambda}})\}$ 
9:   if  $UB_t > \sum_s \hat{\mathcal{J}}_s$  then
10:     $(\mathbf{U}^*, \{\boldsymbol{\eta}_s^*\}_{s=1}^T, \mathcal{J}^*) \leftarrow (\hat{\mathbf{U}}, \{\hat{\boldsymbol{\eta}}_s\}_{s=1}^T, \hat{\mathcal{J}})$ 
11:     $UB_t \leftarrow \sum_s \hat{\mathcal{J}}_s$ 
12:   end if
13:    $\text{Gap} \leftarrow (UB_t - LB_t)/LB_t$ 
14:   if  $\text{Gap} \leq \zeta$  or  $\tau \geq \tau_{max}$  then
15:      $\text{Terminate} \leftarrow \mathbf{TRUE}$ 
16:   end if
17:    $\tau \leftarrow \tau + 1$ 
18: end while

```

Solving Algorithm 8 at each iteration of Algorithm 7 may be computationally prohibitive. Moreover, the information obtained in performing BD at iteration t of Algorithm 7 may be used to warm start the BD in the subsequent iterations of Algorithm 7. In fact, as we show in the following, a single iteration of Algorithm 8 is enough to help Algorithm 7 move forward, and so we can integrate these two algorithms into a unified BD algorithm as explained below.

Lemma 1. *At iteration t of Algorithm 7, for arbitrary $(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\lambda}) \in \Theta_t$, let $\mu \geq \mathcal{B}_t(\mathbf{U}|\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\lambda})$ be a Benders cut produced for solving BMP_t . Then, this cut is valid for BMP_h for any $h > t$.*

Proof. Let $(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\lambda}}) \in \Theta_t$ be an arbitrary solution produced at iteration t . Note that $\mathcal{A}_{st} \subseteq \mathcal{A}_{sh}$ for $h > t$. Define $(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\beta}})$ by setting $\tilde{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}$, $\tilde{\boldsymbol{\lambda}} = \hat{\boldsymbol{\lambda}}$. Moreover, let $\tilde{\beta}_{s\boldsymbol{\alpha}_s} = \hat{\beta}_{s\boldsymbol{\alpha}_s}$ if $\boldsymbol{\alpha}_s \in \mathcal{A}_{st}$ and $\tilde{\beta}_{s\boldsymbol{\alpha}_s} = 0$ if $\boldsymbol{\alpha}_s \in \mathcal{A}_{sh} \setminus \mathcal{A}_{st}$. Note that $(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\beta}})$ is a feasible solution for DSP_h at iteration h of Algorithm 7 and that $\mathcal{B}_t(\mathbf{U}|\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\lambda}}) = \mathcal{B}_h(\mathbf{U}|\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\beta}})$. Therefore, $\mu \geq \mathcal{B}_h(\mathbf{U}|\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\beta}})$, implying validity of this constraint for BMP_h . $\square$

Lemma 2. *Let $\tilde{\Theta}$ be an arbitrary subset of Θ_t , and let $v(\tilde{\Theta})$ be the optimal value of $\text{RBMP}_t(\tilde{\Theta})$. Then, $v(\tilde{\Theta})$ is a valid lower bound on the optimal value of problem (6.21).*

Proof. It is a direct result of the fact that $\text{RBMP}_t(\tilde{\Theta})$ is a relaxation of BMP_t (or equivalently FMP_t), which in turn is a relaxation of problem (6.21). $\square$

Lemma 1 implies that if we solve FMP_t using the proposed BD algorithm, we can reuse the produced Benders cuts in the subsequent iterations of Algorithm 7. Moreover, by Lemma 2, any lower bound obtained during the iterations of Algorithm 8 is a valid lower bound for the subsequent iterations of Algorithm 7. These properties allow for developing a unified BD scheme for solving both cutting-plane algorithm and the inner BD algorithm in a single run, which we refer to as the BD-Forest algorithm described in Algorithm 9. In developing this algorithm, we employ several acceleration techniques as detailed below.

6.5.1 Formulation Strengthening

BD usually suffers from slow convergence, which is often attributed to the weakness of the Benders cuts (Magnanti and Wong 1981). As noted by (Magnanti and Wong 1981), stronger formulation strengthens the Benders cuts (i.e., yields sharper cuts). In the following, we show how FMP can be reformulated using stronger constraints. The relationship between u_{sq} , η_s , and η_q directly impacts the strength of the forest formulation. Currently, we have established this relationship via:

$$1 - u_{sq} \geq \eta_{sm} - \eta_{qm} \quad \forall (s, q, m) : s < q. \quad (6.31)$$

Note that the following constraint is also valid:

$$1 - u_{sq} \geq \eta_{qm} - \eta_{sm} \quad \forall (s, q, m) : s < q. \quad (6.32)$$

Cumulative valid inequalities. In our first approach for strengthening the forest formulation, we note that for any $\hat{P} \subset \{1, \dots, P\}$, the following constraints are valid inequalities for FMP:

$$\begin{aligned} 1 - u_{sq} &\geq \sum_{m \in \hat{P}} (\eta_{sm} - \eta_{qm}) & \forall (s, q) : s < q \\ 1 - u_{sq} &\geq \sum_{m \in \hat{P}} (\eta_{qm} - \eta_{sm}) & \forall (s, q) : s < q. \end{aligned} \quad (6.33)$$

However, since there are exponentially many such constraints, we will focus on a specific set of these constraints. In particular, for each m , let $\hat{P}_m = \{1, \dots, m\}$. Then (6.33) reduces to

$$\begin{aligned} 1 - u_{sq} &\geq \sum_{n=1}^m (\eta_{sn} - \eta_{qn}) & \forall (s, q, m) : s < q \\ 1 - u_{sq} &\geq \sum_{n=1}^m (\eta_{qn} - \eta_{sn}) & \forall (s, q, m) : s < q, \end{aligned} \quad (6.34)$$

which we refer to as the cumulative valid inequalities. As given in Proposition 6

below, cumulative valid inequalities can replace the normal constraints (6.31) and (6.32).

Proposition 6. *Cumulative valid inequalities are not dominated by the normal constraints (6.31) and (6.32). Additionally, cumulative valid inequalities are sufficient for establishing the relationship between tasks and clusters.*

Proof. For arbitrary $\boldsymbol{\eta}_s$ and $\boldsymbol{\eta}_q$, define functions $f_1(\boldsymbol{\eta}_s, \boldsymbol{\eta}_q)$ and $f_2(\boldsymbol{\eta}_s, \boldsymbol{\eta}_q)$ as

$$f_1(\boldsymbol{\eta}_s, \boldsymbol{\eta}_q) = \maximize_m |\eta_{sm} - \eta_{qm}|$$

$$f_2(\boldsymbol{\eta}_s, \boldsymbol{\eta}_q) = \maximize_m \left| \sum_{n=1}^m \eta_{sn} - \eta_{qn} \right|.$$

Using these functions, constraints (6.31) and (6.32) are equivalent to

$$1 - u_{sq} \geq f_1(\boldsymbol{\eta}_s, \boldsymbol{\eta}_q). \quad (6.35)$$

Similarly, cumulative inequalities read as

$$1 - u_{sq} \geq f_2(\boldsymbol{\eta}_s, \boldsymbol{\eta}_q). \quad (6.36)$$

Hence, to show that cumulative constraints are not dominated by the normal constraints, it suffices to show that there exists a solution $(\hat{\boldsymbol{\eta}}_s, \hat{\boldsymbol{\eta}}_q)$ for which $f_1(\hat{\boldsymbol{\eta}}_s, \hat{\boldsymbol{\eta}}_q) < f_2(\hat{\boldsymbol{\eta}}_s, \hat{\boldsymbol{\eta}}_q)$. Observe that $f_1(\boldsymbol{\eta}_s, \boldsymbol{\eta}_q) = f_2(\boldsymbol{\eta}_s, \boldsymbol{\eta}_q)$ for $P \leq 2$. Also, it is not hard to see that $f_2(\boldsymbol{\eta}_s, \boldsymbol{\eta}_q) \leq f_1(\boldsymbol{\eta}_s, \boldsymbol{\eta}_q)$ for $P = 3$. For $P \geq 4$, let $\hat{\boldsymbol{\eta}}_s = (0.1, 0.2, 0.3, 0.4, 0, \dots)$ and $\hat{\boldsymbol{\eta}}_q = (0, 0.05, 0.4, 0.55, 0, \dots)$. Note that $f_1(\hat{\boldsymbol{\eta}}_s, \hat{\boldsymbol{\eta}}_q) = 0.15 < f_2(\hat{\boldsymbol{\eta}}_s, \hat{\boldsymbol{\eta}}_q) = 0.25$, which means the cumulative constraints are not dominated by the normal constraints.

We now show that cumulative valid inequalities are sufficient for establishing the relationship between tasks and clusters. Note that if $u_{sq} = 0$, then both type of constraints are inactive. Therefore, we assume that $u_{sq} = 1$ in the following. We prove the statement by induction. For $m = 1$, cumulative constraints are equivalent to normal constraints. Now, assume that the statement is true for $m = k < P$ and

we show that it is true for $m = k + 1$. If $u_{sq} = 1$, then, for $m = k$, we must have

$$\begin{cases} 0 \geq \sum_{n=1}^k (\eta_{sn} - \eta_{qn}) \\ 0 \geq \sum_{n=1}^k (\eta_{qn} - \eta_{sn}) \end{cases} \Rightarrow 0 = \sum_{n=1}^k (\eta_{sn} - \eta_{qn}),$$

which implies that for $m = k + 1$

$$\sum_{n=1}^{k+1} (\eta_{sn} - \eta_{qn}) = \eta_{sm} - \eta_{qm} + \sum_{n=1}^k (\eta_{sn} - \eta_{qn}) = \eta_{sm} - \eta_{qm} \leq 0,$$

which implies constraint (6.31). The same can be shown for constraint (6.32). $\square$

Strong valid inequalities. In our second approach, instead of using normal constraints (6.31) and (6.32) or the cumulative constraints (6.34), we establish the relationship between tasks and clusters by introducing new variables. Define variable φ_{sqm} as the minimum of η_{sm} and η_{qm} when $u_{sq} = 1$ and 0 otherwise. Using these variables, the relationship between u_{sq} , $\boldsymbol{\eta}_s$ and $\boldsymbol{\eta}_q$ can be restated via

$$u_{sq} = \sum_{m=1}^P \varphi_{sqm} \quad \forall (s, q): s < q \quad (6.37)$$

$$0 \leq \varphi_{sqm} \leq \eta_{sm} \quad \forall (s, q, m): s < q \quad (6.38)$$

$$0 \leq \varphi_{sqm} \leq \eta_{qm} \quad \forall (s, q, m): s < q. \quad (6.39)$$

Note that when $u_{sq} = 1$, (6.37) implies that $\sum_{m=1}^P \varphi_{sqm} = 1$. Therefore, (6.38) together with the fact that $\sum_{m=1}^P \eta_{sm} = 1$ imply that $\varphi_{sqm} = \eta_{sm}$ for each m . With a similar reasoning, using (6.39), we have $\varphi_{sqm} = \eta_{qm}$, which establishes the required relationship. Moreover, when $u_{sq} = 0$ and $\varphi_{sqm} = 0$, the constraints become redundant.

Proposition 7. Constraints (6.37)–(6.39) dominate normal constraints (6.31)–(6.32).

Proof. First, note that constraints (6.37)–(6.39) yield

$$u_{sq} = \sum_{m=1}^P \varphi_{sqm} \leq \sum_{m=1}^P \min\{\eta_{sm}, \eta_{qm}\}. \quad (6.40)$$

We may rewrite $\sum_{m=1}^P \min\{\eta_{sm}, \eta_{qm}\}$ as

$$\begin{aligned} \sum_{m=1}^P \min\{\eta_{sm}, \eta_{qm}\} &= 1 - \frac{1}{2} \left(2 - 2 \sum_{m=1}^P \min\{\eta_{sm}, \eta_{qm}\} \right) \\ &= 1 - \frac{1}{2} \sum_{m=1}^P (\eta_{sm} - \min\{\eta_{sm}, \eta_{qm}\} + \eta_{qm} - \min\{\eta_{sm}, \eta_{qm}\}) \\ &= 1 - \frac{1}{2} \sum_{m=1}^P (\max\{0, \eta_{sm} - \eta_{qm}\} + \max\{0, \eta_{qm} - \eta_{sm}\}) = 1 - \frac{1}{2} \sum_{m=1}^P |\eta_{sm} - \eta_{qm}|, \end{aligned}$$

where we have used $\sum_{m=1}^P \eta_{sm} = \sum_{m=1}^P \eta_{qm} = 1$. Hence, (6.40) is equivalent to

$$1 - u_{sq} \geq \frac{1}{2} \sum_{m=1}^P |\eta_{sm} - \eta_{qm}|. \quad (6.41)$$

Therefore, it suffices to show that the right-hand-side value of (6.41) is greater than that of (6.35). Define m^* as

$$m^* = \arg \max_m |\eta_{sm} - \eta_{qm}|.$$

Since $\boldsymbol{\eta}_s \in \Delta$ and $\boldsymbol{\eta}_q \in \Delta$, we have

$$\begin{aligned} \sum_{m=1}^P |\eta_{sm} - \eta_{qm}| &= |\eta_{sm^*} - \eta_{qm^*}| + \sum_{m \neq m^*} |\eta_{sm} - \eta_{qm}| \\ &\geq |\eta_{sm^*} - \eta_{qm^*}| + \left| \sum_{m \neq m^*} \eta_{sm} - \eta_{qm} \right| \\ &= |\eta_{sm^*} - \eta_{qm^*}| + |1 - \eta_{sm^*} - (1 - \eta_{qm^*})| = 2|\eta_{sm^*} - \eta_{qm^*}|, \end{aligned}$$

which yields

$$\sum_{m=1}^P |\eta_{sm} - \eta_{qm}| \geq 2 \max_m |\eta_{sm} - \eta_{qm}|,$$

implying

$$1 - u_{sq} \geq \frac{1}{2} \sum_{m=1}^P |\eta_{sm} - \eta_{qm}| \geq \max_m |\eta_{sm} - \eta_{qm}|,$$

which completes the proof. $\square$

When constraint set (6.26) is replaced with constraints (6.37)–(6.39), we refer to the resulting formulation as the *strong* forest formulation (SFMP). Consequently, associating dual variables λ_{sq} , ν_{sqm} and ω_{sqm} with constraints (6.37)–(6.39), the modified DSP is reformulated as

$$\text{maximize} \quad \sum_{s=1}^T \left[\theta_s + \sum_{\alpha_s \in \mathcal{A}_{st}} \beta_s \alpha_s |\alpha_s| + \sum_{q=s+1}^T \lambda_{sq} \hat{u}_{sq} \right]$$

with respect to $\boldsymbol{\theta} \in \mathbb{R}^T$, $\{\beta_s \in \mathbb{R}_+^{|\mathcal{A}_{st}|}\}_{s=1}^T$, $\boldsymbol{\lambda} \in \mathbb{R}^{T \times T}$, $\{\omega_m \in \mathbb{R}_+^{T \times T}, \boldsymbol{\nu}_m \in \mathbb{R}_+^{T \times T}\}_{m=1}^P$

$$\text{subject to} \quad \sum_{\alpha_s \in \mathcal{A}_{st}} \beta_s \alpha_s \leq 1 \quad \forall s \quad (6.42)$$

$$\begin{aligned} \theta_s + \frac{1}{2} \sum_{\alpha_s \in \mathcal{A}_{st}} \beta_s \alpha_s \kappa_{sm}(\alpha_s) \\ + \sum_{q=1}^{s-1} \omega_{qsm} + \sum_{q=s+1}^T \nu_{sqm} \leq 0 \quad \forall (s, m) \end{aligned} \quad (6.43)$$

$$\lambda_{sq} - \nu_{sqm} - \omega_{sqm} \leq 0 \quad \forall (s, q, m): s < q, \quad (6.44)$$

which yields a cut of the form $\mu \geq \mathcal{B}(\mathbf{U}|\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\nu}, \boldsymbol{\omega})$ where

$$\mathcal{B}(\mathbf{U}|\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\nu}, \boldsymbol{\omega}) = \sum_{s=1}^T (\theta_s + \sum_{\alpha_s \in \mathcal{A}_{st}} \beta_s \alpha_s |\alpha_s|) + \sum_{s=1}^T \sum_{q=s+1}^T \lambda_{sq} u_{sq}.$$

6.5.2 Pareto-optimal cuts

For an arbitrary solution $\hat{\mathbf{U}}$, if DSP possesses alternative optimal solutions, each optimal solution can be used to produce a Benders optimality cut, but with potentially different strengths. As introduced by (Magnanti and Wong 1981), Benders cut $\tilde{\mathcal{B}}(\mathbf{U})$ dominates (is stronger than) Benders cut $\hat{\mathcal{B}}(\mathbf{U})$ if $\tilde{\mathcal{B}}(\mathbf{U}) \geq \hat{\mathcal{B}}(\mathbf{U}) \quad \forall \mathbf{U} \in \mathcal{F}$ with strict inequality satisfied for at least one solution $\mathbf{U} \in \mathcal{F}$. If no other cut dominates Benders cut $\mathcal{B}(\mathbf{U})$, we say this cut is Pareto-optimal (or non-dominated). Let $v(\hat{\mathbf{U}})$ be the optimal value of the subproblem for a given solution $\hat{\mathbf{U}}$. (Magnanti and Wong 1981) showed that a Pareto-optimal cut can be chosen among the alternative optimal solutions of DSP by solving the following auxiliary subproblem (PO1):

$$\begin{aligned} & \text{maximize} \quad \sum_{s=1}^T \left[\theta_s + \sum_{\alpha_s \in \mathcal{A}_{st}} \beta_s \alpha_s |\alpha_s| + \sum_{q=s+1}^T \sum_{m=1}^P \lambda_{sqm} (\tilde{u}_{sq} - 1) \right] \\ & \text{with respect to} \quad \boldsymbol{\theta} \in \mathbb{R}^T, \{\boldsymbol{\lambda}_m \in \mathbb{R}_+^{T \times T}\}_{m=1}^P, \{\boldsymbol{\beta}_s \in \mathbb{R}_+^{|\mathcal{A}_{st}|}\}_{s=1}^T \\ & \text{subject to} \quad (6.28), (6.29) \\ & \quad \sum_{s=1}^T \left[\theta_s + \sum_{\alpha_s \in \mathcal{A}_{st}} \beta_s \alpha_s |\alpha_s| + \sum_{q=s+1}^T \sum_{m=1}^P \lambda_{sqm} (\hat{u}_{sq} - 1) \right] = v(\hat{\mathbf{U}}), \end{aligned}$$

where $\tilde{\mathbf{U}}$ is called a core point which belongs to $\text{relint}(\mathcal{F})$ (i.e., relative interior of convex hull of $\mathcal{F}$). Similarly, a Pareto-optimal cut based on the strong formulation can be produced by solving the following auxiliary subproblem (PO2):

$$\begin{aligned} & \text{maximize} \quad \sum_{s=1}^T \left[\theta_s + \sum_{\alpha_s \in \mathcal{A}_{st}} \beta_s \alpha_s |\alpha_s| + \sum_{q=s+1}^T \lambda_{sq} \tilde{u}_{sq} \right] \\ & \text{with respect to} \quad \boldsymbol{\theta} \in \mathbb{R}^T, \{\boldsymbol{\beta}_s \in \mathbb{R}_+^{|\mathcal{A}_{st}|}\}_{s=1}^T, \boldsymbol{\lambda} \in \mathbb{R}^{T \times T}, \{\boldsymbol{\omega}_m \in \mathbb{R}_+^{T \times T}, \boldsymbol{\nu}_m \in \mathbb{R}_+^{T \times T}\}_{m=1}^P \\ & \text{subject to} \quad (6.42), (6.43), (6.44) \\ & \quad \sum_{s=1}^T \left[\theta_s + \sum_{\alpha_s \in \mathcal{A}_{st}} \beta_s \alpha_s |\alpha_s| + \sum_{q=s+1}^T \lambda_{sq} \hat{u}_{sq} \right] = v(\hat{\mathbf{U}}) \end{aligned}$$

6.5.3 Valid Core Points

We now show how a core point can be computed using Proposition 8 below.

Proposition 8. Let $\xi \in (0, \frac{K-1}{K}]$ be an arbitrary number. Define $\tilde{\mathbf{U}}$ as $\tilde{u}_{sq} = c\xi^{q-s}$, where c is a scalar defined as

$$c = \frac{T - K}{\frac{\xi}{1-\xi} \left(T - 1 - \frac{\xi}{1-\xi} (1 - \xi^{T-1}) \right)}.$$

Then, $\tilde{\mathbf{U}} \in \text{relint}(\mathcal{F})$ and so defines a valid core point.

Proof. First, we need to show that $\tilde{\mathbf{U}}$ satisfies (6.18) as equality. We have

$$\begin{aligned} \sum_{s=1}^T \sum_{q=s+1}^T \tilde{u}_{sq} &= c \sum_{s=1}^{T-1} \sum_{q=s+1}^T \xi^{q-s} = c \sum_{s=1}^{T-1} \sum_{n=1}^{T-s} \xi^n \\ &= c \sum_{s=1}^{T-1} \frac{\xi}{1-\xi} (1 - \xi^{T-s}) = c \frac{\xi}{1-\xi} \left(T - 1 - \sum_{n=1}^{T-1} \xi^n \right) \\ &= c \frac{\xi}{1-\xi} \left(T - 1 - \frac{\xi}{1-\xi} (1 - \xi^{T-1}) \right) = T - K. \end{aligned}$$

We further show that $0 < \tilde{u}_{sq} < 1$ and that $\tilde{\mathbf{U}}$ satisfies (6.22) as strict inequalities. Since $0 < \tilde{u}_{sq} < 1$ is implied by (6.22), we only need to prove strict inequality for (6.22). In the following, we show that $\sum_{q=s+1}^T u_{sq} \leq 1$ is satisfied as strict inequality for $\tilde{\mathbf{U}}$, the same can be shown for $\sum_{q=1}^{s-1} u_{qs} \leq 1$ similarly. For each s we have

$$\begin{aligned} \sum_{q=s+1}^T \tilde{u}_{sq} &= c \sum_{q=s+1}^T \xi^{q-s} = c \sum_{n=1}^{T-s} \xi^n = c \frac{\xi}{1-\xi} (1 - \xi^{T-s}) \\ &= \frac{T - K}{T - 1 - \frac{\xi}{1-\xi} (1 - \xi^{T-1})} (1 - \xi^{T-s}) < 1. \end{aligned}$$

This holds because $\xi \leq \frac{K-1}{K}$ implies that $K \geq 1 + \frac{\xi}{1-\xi}$, therefore $T - K < T - 1 - \frac{\xi}{1-\xi} (1 - \xi^{T-1})$. Additionally, $0 < 1 - \xi^{T-s} < 1$ since $0 < \xi < 1$. $\square$

In our computations, we simply set $\xi = 0.5$ for which Proposition 8 yields the following core point.

Corollary 2. $\tilde{\mathbf{U}}$ with $\tilde{u}_{sq} = \frac{T-K}{T-2+0.5^{T-1}} 0.5^{q-s}$ is a core point.

6.5.4 BDForest Algorithm

Incorporating strong formulation, Pareto-optimal cuts, and symmetry breaking technique yields the accelerated BD algorithm given in Algorithm 9, which we refer to as the BDForest algorithm. In this algorithm, RBMP is formulated based on the definition of $\mathcal{F}$ as given in (6.23). Figure 6.3 illustrates the overall flow of this algorithm.

Algorithm 9 BDForest: Accelerated BD algorithm for the forest formulation

```

1:  $UB \leftarrow +\infty; LB \leftarrow -\infty; t \leftarrow 1; \mathcal{A}_s \leftarrow \emptyset; \Theta_t \leftarrow \emptyset; \text{Terminate} \leftarrow \mathbf{FALSE}$ 
2: Compute core point  $\tilde{\mathbf{U}}$  using Proposition 8.
3: while Terminate = FALSE do
4:   SOLVE RBMP $_t(\Theta_t)$ ; obtain  $(\hat{\mu}, \hat{\mathbf{U}})$ 
5:    $LB \leftarrow \hat{\mu}$ 
6:   SOLVE PSP using  $\hat{\mathbf{U}}$  to obtain  $(\hat{\mathcal{J}}, \{\hat{\boldsymbol{\eta}}_s\}_{s=1}^T)$ 
7:    $v(\hat{\mathbf{U}}) \leftarrow \sum_s \hat{\mathcal{J}}_s$ 
8:   SOLVE PO1 (or PO2) using  $\hat{\mathbf{U}}$  to obtain  $(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\lambda}})$ 
9:    $\Theta^{t+1} \leftarrow \Theta_t \cup \{(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\lambda}})\}$ 
10:  for  $s = 1, \dots, T$  do
11:    SOLVE DSVM (6.1) using  $\hat{\boldsymbol{\eta}}_s$ ; obtain  $\hat{\boldsymbol{\alpha}}_s$  and  $J_s(\hat{\boldsymbol{\eta}}_s)$ 
12:     $\mathcal{A}_s \leftarrow \mathcal{A}_s \cup \{\hat{\boldsymbol{\alpha}}_s\}$ 
13:  end for
14:   $UB \leftarrow \min\{UB, \sum_s J_s(\hat{\boldsymbol{\eta}}_s)\}; \text{Gap} \leftarrow \frac{UB - LB}{LB}$ 
15:  if  $\text{Gap} \leq \zeta$  or  $t \geq t_{max}$  then
16:    Terminate  $\leftarrow \mathbf{TRUE}$ 
17:  end if
18:   $t \leftarrow t + 1$ 
19: end while

```

6.6 Discussion

In this work, we propose the time-efficient BDForest algorithm for solving large-scale instances of the multitask multiple kernel learning problems together with clustering. In many multitask machine learning methods, a measure of similarity between the tasks is assumed to be given. Such an assumption largely simplifies the optimization problems. However, in many important applications (e.g., cancer

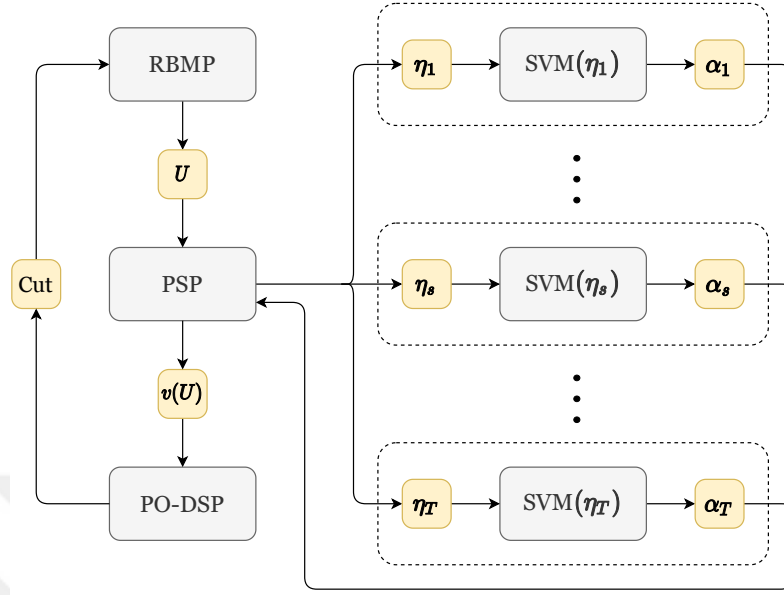


Figure 6.3: Flow of information between different modules of BDForest algorithm.

research) there is limited prior knowledge or consensus about the similarity of the tasks. Hence, grouping mechanisms have to be included to define the similarity measure and to learn the best prediction models simultaneously.

The optimization challenges of the MTMKLC problem are not limited to the integer variables used for clustering of tasks, the other challenges are (i) there are multiple feasible solutions referring to the same clustering, and (ii) the multiple kernel problem is a bi-level problem on its own. We propose the forest formulation in addition to symmetry breaking constraints to solve the problem of redundant solutions. Moreover, the bi-level problem for each task can be turned into a single level problem using cutting planes. Both remedies add a large number of constraints to the problem that cannot be generated explicitly. Hence, we use Benders decomposition for generating and adding the constraints as they are needed. We also employ several enhancement techniques for accelerating this algorithm.

Chapter 7

EXPERIMENTAL RESULTS

To test the predictive performance of MKL on gene sets algorithm and our extensions to MKL introduced in this thesis, namely, SpMKL, MTMKL and BDForest in discriminating early- and late-stage cancers, we performed two sets of experiments E1 and E2 respectively on 15 and 18 datasets that we constructed from TCGA cohorts for two alternative labelling strategies (see Table 2.1) by comparing against two baseline algorithms, namely, RFs and SVMs. We compared against RFs since they were frequently used in phenotype prediction tasks of several bioinformatics applications. We compared against SVMs since MKL on gene sets algorithm is mainly based on SVMs, and we wanted to see the effect of integrating the pathway/gene set information into the classification algorithm. For the single-task MKL and SpMKL algorithms, we also performed experiments on two-year survival analysis, i.e., S1 (see Table 2.2).

7.1 Experimental Settings

For each dataset, we picked 80% of the tumors as the training set, and we set aside the remaining 20% as the test set. While doing so, we tried to keep the negative and positive class ratios in training and test sets almost equal (i.e., stratification). The training set was normalized to have zero mean and unit standard deviation, and the test set was then normalized using the mean and the standard deviation of the original training set. We repeated this procedure 100 times to obtain more robust results and reported the final results over these 100 replications. In each replication, the hyper-parameters for RFs, SVMs and MKL on gene sets and other variants of MKL were selected using a four-fold inner cross-validation on the training set.

Figure 7.1 shows the overall unbiased performance evaluation framework we de-

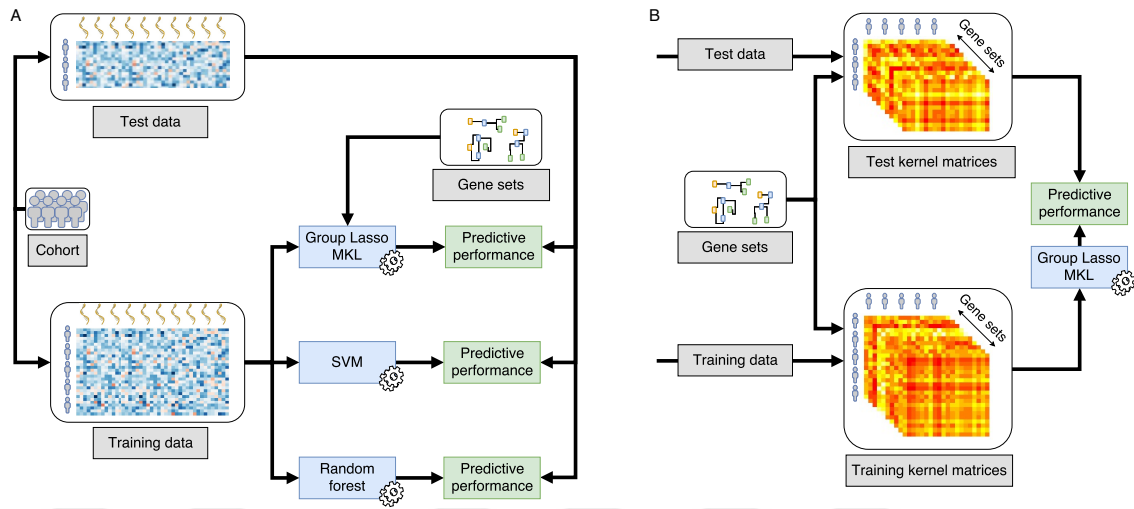


Figure 7.1: Overview of the evaluation framework we developed for predicting pathological stages of primary tumors and survival outcome prediction of patients from their gene expression profiles.

veloped in this study. On the TCGA datasets we constructed, we compared three main machine learning algorithms, namely, random forests (RFs), support vector machines (SVMs), and multiple kernel learning (MKL) on gene sets (Figure 7.1A). Figure 7.1B illustrates integrating pathways/gene sets into the classification algorithm, where we calculate a kernel matrix using the expression features of genes that are included in each pathway/gene set during training or testing. RFs and SVMs use gene expression profiles of tumors to predict their pathological stages. However, in addition to gene expression profiles, MKL also uses a gene set database and extracts additional knowledge about the differences between early- and late-stage cancers in terms of gene sets by discarding some of them in the final classifier.

To compare the predictive performances of the algorithms, we used area under the receiver operating characteristic curve (AUROC). AUROC is used to summarize the receiver operating characteristic curve, which is a curve of true positives as a function of false positives while the threshold to predict class labels changes. Larger AUROC values correspond to better predictive performance.

For RFs, we used `randomForestSRC` R package version 2.5.1 (Ishwaran and Kogalur 2017). We picked the number of trees to grow parameter `ntree` from the set

$\{500, 1000, \dots, 2500\}$ using the four-fold inner cross-validation strategy described.

For SVMs and MKL on gene sets, we used our own implementations in R, which uses MOSEK version 8.1.0.34 to solve quadratic optimization problems (MOSEK ApS 2017). As presented in Section 3.3, to calculate a similarity measure between gene expression profiles of tumors, we used the Gaussian kernel defined as

$$k_G(\mathbf{x}_i, \mathbf{x}_j) = \exp \left(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_j)}{2\sigma^2} \right),$$

where we picked the kernel width parameter σ as the mean of pairwise Euclidean distances between training instances. We selected the regularization parameter C from the set $\{10^{-4}, 10^{-3}, \dots, 10^{+5}\}$ using the four-fold inner cross-validation strategy described.

For MKL on gene sets, we performed 200 iterations to guarantee the convergence since the algorithm usually converges in tens of iterations. Note that the Gaussian kernel functions were calculated on subsets of gene expression profiles by looking at the genes included in the corresponding gene sets, and the kernel width parameters were selected accordingly.

7.2 Multiple Kernel Learning on Gene Sets

7.2.1 Predictive Performance

On 15 cancer datasets for the first set of experiments E1, we compared three machine learning algorithms, namely, random forest (RF), support vector machine (SVM) and multiple kernel learning on the Hallmark gene sets (MKL[H]), in terms of their predictive performances.

Figure 7.2 shows the predictive performances of RF, SVM and MKL[H] on each cohort separately. In this figure, the violin plots compare the AUROC values of the algorithms over 100 replications. SVM and MKL[H] are compared against RF using a two-tailed paired t -test to check whether there is a significant difference between their performances. For p -value results, **red**: RF is better; **green**: SVM is better; **blue**: MKL[H] is better; black: no difference. The **orange** dashed lines show the baseline

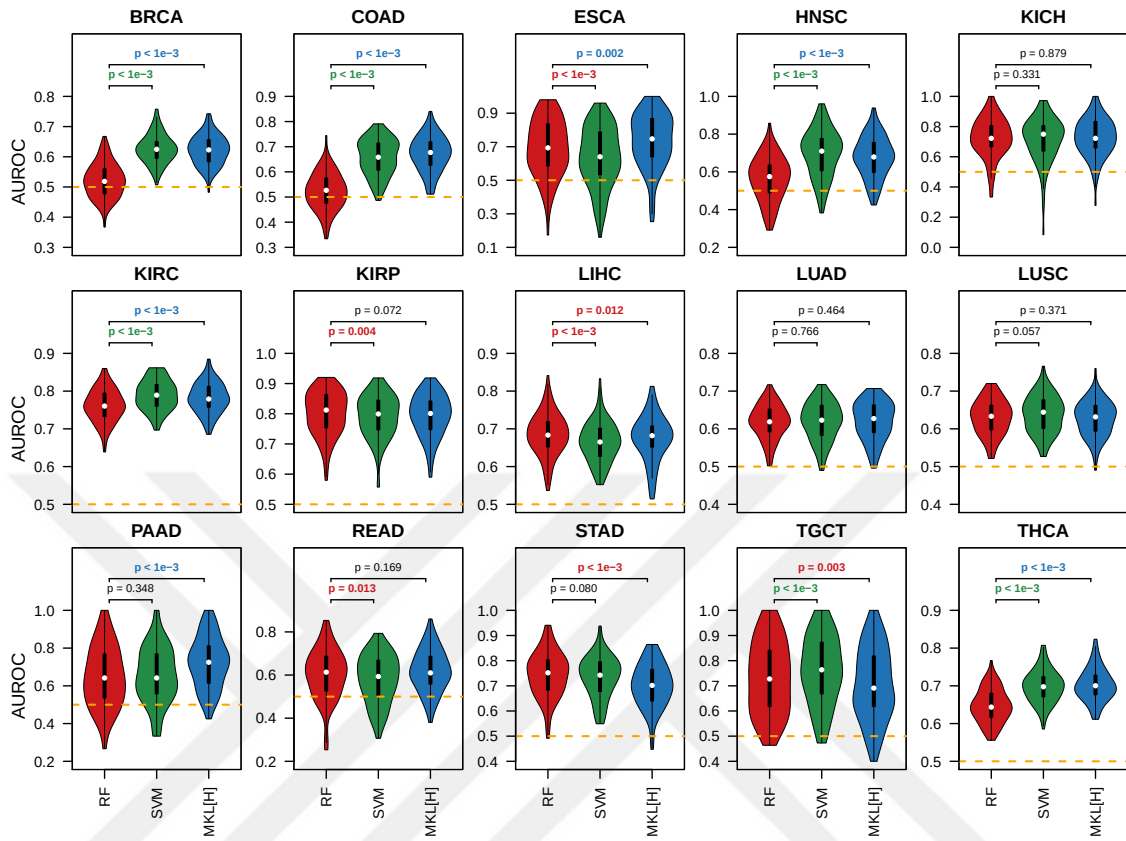


Figure 7.2: Predictive performances of RF, SVM and MKL [H] on 15 cancer datasets in E1 constructed from TCGA cohorts.

performance level (i.e., $\text{AUROC} = 0.5$). We see that the median performances of all three algorithms are better than the baseline performance (i.e., 0.5 AUROC value shown as dashed lines) on all datasets, which indicates that gene expression profiles carry meaningful information about pathological stages.

When we compare the performances of RF and SVM, we see that SVM obtained significantly better results on six out of 15 datasets (i.e., BRCA, COAD, HNSC, KIRC, TGCT and THCA), whereas RF was significantly better on four of them (i.e., ESCA, KIRP, LIHC and READ). Although RF is also a nonlinear model, the non linearity brought by the Gaussian kernel makes SVM a better algorithm for this highly complex classification problem. SVM improved the predictive performance by 10.44% on BRCA, 13.56% on COAD, 13.36% on HNSC and 5.12% on THCA, whereas the largest performance drop was 5.51% on ESCA.

When we compare the performances of **RF** and **MKL[H]**, we see that **MKL[H]** obtained significantly better results on seven out of 15 datasets (i.e., **BRCA**, **COAD**, **ESCA**, **HNSC**, **KIRC**, **PAAD** and **THCA**), whereas **RF** was significantly better on three of them (i.e., **LIHC**, **STAD** and **TGCT**). We see that principled combination of gene set information in the form of kernel functions increased the predictive performance even though **MKL[H]** used a small portion of the gene expression profiles. To be more specific, **RF** and **SVM** were using 19,814 gene expression features, whereas **MKL[H]** was using only 4,357 (i.e., less than one-fourth) gene expression features for the genes included in the **Hallmark** gene sets. **MKL[H]** improved the predictive performance by 10.00% on **BRCA**, 14.89% on **COAD**, 4.30% on **ESCA**, 11.43% on **HNSC**, 7.59% on **PAAD** and 5.50% on **THCA**, whereas the largest performance drop was 4.35% on **STAD**.

Note that **SVM** also outperformed **RF** in this set of experiments, but **MKL[H]** used significantly fewer gene expression features (i.e., even less than 4,357 input features) by discarding uninformative gene sets from the machine learning model and allowed us to identify informative ones for classification.

Figure A.1 shows the predictive performances of **RF**, **SVM** and **MKL[H]** for the second set of experiments **E2** (i.e., **Stage I** or **II** versus **Stage III** or **IV**). We see that the ordering of the algorithms with respect to their predictive performances stays the same (i.e., $\text{RF} < \text{SVM} < \text{MKL[H]}$).

7.2.2 Biological Mechanisms

To illustrate the biological relevance of our **MKL[H]** algorithm, we analysed its ability to identify relevant gene sets based on the kernel weights inferred during training. For each dataset and gene set pair, we counted the number of replications in which the corresponding kernel weight is nonzero (i.e., the number of replications where $\eta_m \neq 0$ was satisfied). Figure 7.3 shows the selection frequencies of 50 gene sets in the **Hallmark** collection for 15 cancer datasets in the first set of experiments **E1**. In this figure, rows and columns are clustered using hierarchical clustering with Euclidean distance and complete linkage functions. Column sums of selection frequencies are reported to identify datasets that use higher number of gene sets on the average.

Row sums of selection frequencies are reported to identify frequently selected gene sets across different datasets.

By looking at the column sums of the selection frequencies, we see that discriminating early- and late-stage cancers from each other is much more difficult in some disease types. For example, in BRCA, KIRC, LUSC and THCA datasets, MKL[H] used more than 15 out of 50 gene sets on the average. However, in some disease types such as ESCA and PAAD, MKL[H] used very few gene sets (less than six out of 50 gene set on the average) and even improved the predictive performance significantly compared to RF and SVM algorithms.

When we look at the row sums of the selection frequencies, we see that some gene sets were selected heavily across different datasets. For example, ANGIOGENESIS, KRAS_SIGNALING_DN, MYOGENESIS and SPERMATOGENESIS gene sets were used in more than six out of 15 datasets on the average, which were reported to be related to the cancer formation in early stages (Bergers and Benjamin 2003). Similarly, four metabolism-related gene sets, namely, BILE_ACID_METABOLISM, HEME_METABOLISM, PANCREAS_BETA_CELLS and XENOBIOTIC_METABOLISM were used more than five out of 15 datasets on the average. For diseases associated with the tissues that are known to be directly related to metabolism such as KIRC (kidney), LIHC (liver), PAAD (pancreas) and THCA (thyroid gland), MKL[H] picked these metabolism-related gene sets with very high frequencies. Gene sets that have a quite important role in epithelial cells, namely, APICAL_SURFACE and HYPOXIA, were selected for more than five out of 15 datasets on the average. These two gene sets were picked with very high frequencies in BRCA (breast), COAD (colon), LUAD (lung) and STAD (stomach) whose tissues are known to contain many epithelial cells. We also see that MKL[H] did not pick gene sets that were expected to be irrelevant for discriminating early- and late-stage cancers from each other. For example, cell cycle related gene sets such as DNA_REPAIR, E2F_TARGETS, G2M_CHECKPOINT, MYC_TARGETS_V1, MYC_TARGETS_V2 and MTORC1_SIGNALING were selected with very low frequencies across 15 datasets.

Figure A.2 gives the selection frequencies of 50 gene sets in the Hallmark collection for 18 datasets in the second set of experiments E2, where we can make similar

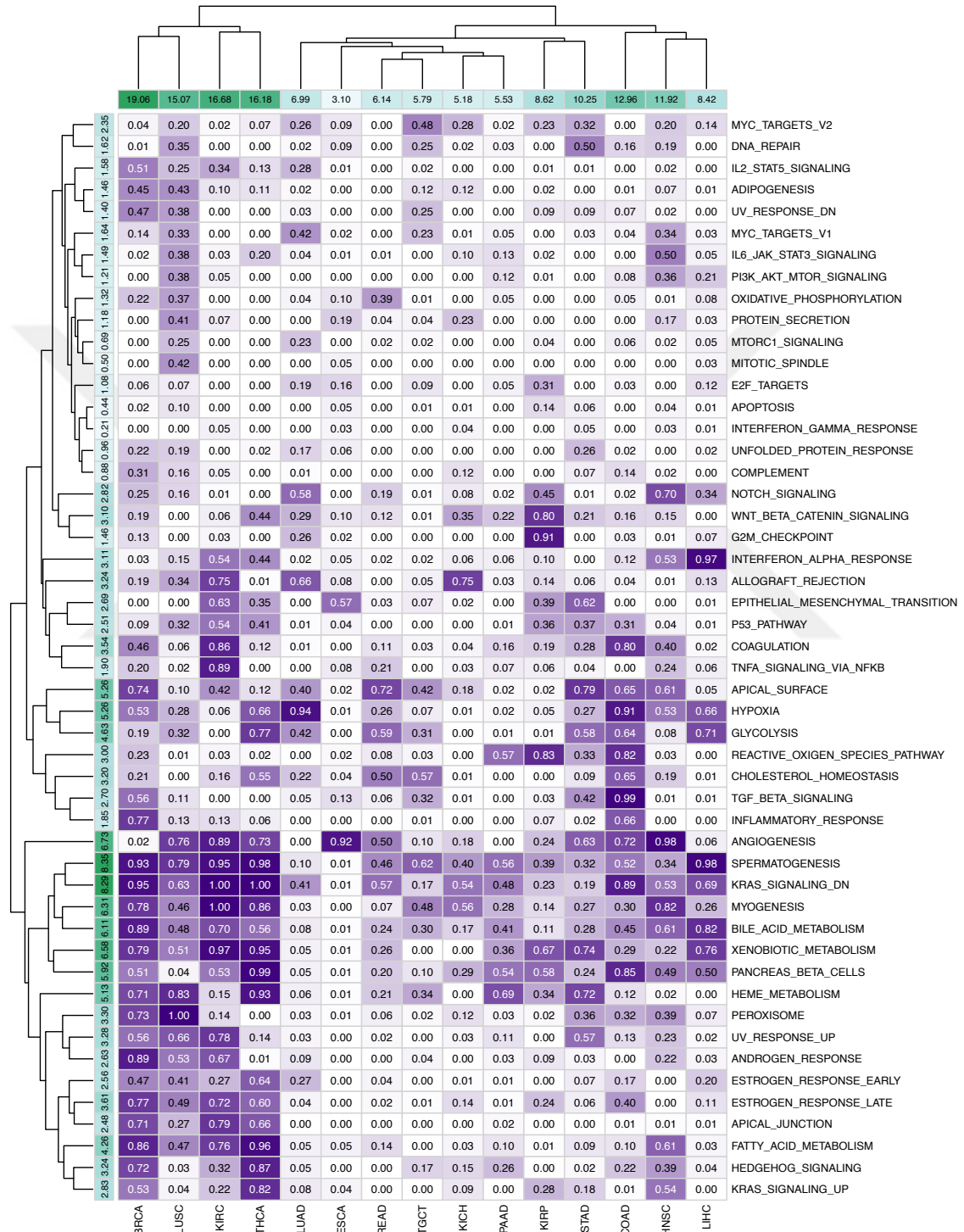


Figure 7.3: Selection frequencies of 50 gene sets in the Hallmark collection as identified by MKL[H] for 15 cancer datasets in E1.

observations about selected gene sets.



7.3 Sparse Multiple Kernel Learning

To test the predictive performance of SpMKL algorithm, we performed experiments on 27 datasets we constructed from the TCGA cohorts. As the baseline algorithm, we used random forests (RF) (Breiman 2001) and SVM, which are frequently used in phenotype prediction tasks of several bioinformatics applications. We also compared the performance of SpMKL against three variants of MKL on gene sets algorithm, namely SimpleMKL (Rakotomamonjy et al. 2008), group Lasso MKL and non-sparse ℓ_2 -MKL (Xu et al. 2010)

Additionally, for the first step of the SpMKL problem, we selected the number of kernels G from the set $\{5, 10, 15, 20\}$ and the penalty weight parameter ρ in the second step of SpMKL problem from the set $\{10^{-1}, 10^0, \dots, 10^{+4}\}$.

We performed 200 iterations to guarantee the convergence of the iterative algorithms for solving variants of MKL as well as penalized MKL problems. Moreover, we used the optimality gap threshold of $\zeta = 10^{-5}$ along with up to 200, 50 and 150 iterations respectively for the first, second and third phases of the three-phase algorithm for solving instances of the binary MKL problem. To produce the significance parameters $\{\theta_m\}_{m=1}^P$ for the kernel functions in the second step of SpMKL, we performed the first step of the SpMKL problem over 100 randomly generated replications. Each replication yielded a vector of kernel selection via binary variables $\mathbf{g}$. We then set θ_m for each kernel m as the average of g_m values over the 100 replications.

We implemented six classification methods in total. For SimpleMKL, we used R package RMKL version 1.0 (Wilson and Lir 2019), respectively. For SVM and other MKL algorithms, we used our own implementations in R with CPLEX 12.8 for solving the integer linear programs as well as the quadratic SVM subproblems.

7.3.1 Predictive Performance

For predictive performance comparison and to validate the robustness of our methods, we conducted three sets of experiments. In the first set of experiments (E1), we

constructed 15 datasets from TCGA cohorts for discriminating early- and late-stage cancers by considering **Stage I** as the early-stage and **Stages II,III,IV** as the late-stage cancers. In the second set of experiments (**E2**), we conducted another variant of **E1** by considering **Stages I-II** as the early-stage and **Stages III-IV** as the late-stage tumors. In the third set of experiments (**S1**), we constructed 20 datasets from TCGA cohorts for two-year survival analysis.

For each experiment, we compared 10 classification algorithms: random forest (RF), SVM using Gaussian kernel (SVM), four MKL algorithms with $P = 50$ kernels constructed based on the **Hallmark** gene sets, namely, SimpleMKL (SMKL[H]), group Lasso MKL (gLMKL[H]), ℓ_2 -MKL (L2MKL[H]) and SpMKL (SpMKL[H]), together with four MKL algorithms with $P = 196$ kernels based on the **PID** pathways (SMKL[P], gLMKL[P], L2MKL[P] and SpMKL[P])

Discriminating early- and late-stage cancers. Figure 7.4 illustrates the predictive performances of RF, SVM, SMK[L], gLMKL[H], L2MKL[H], SpMKL[H], SMK[P], gLMKL[P], L2MKL[P] and SpMKL[P] across the datasets created for experiment **E1**. In this figure, the violin plots illustrate the distribution of AUROC values across 100 replications of each algorithm. Moreover, SpMKL[H] and SpMKL[P] are compared against RF, SVM and the other three MKL methods using the two-sided paired t -test to check whether there are significant differences between the performances of algorithm pairs. The comparison between SpMKL and other MKL algorithms demonstrates the advantage of extreme sparsity. For p -value results, colors indicate that the respective method is statistically significantly better, and black means there is no difference.

In this experiment, the mean AUROC value averaged over 15 cohorts has increased from 69.78% in gLMKL[H] to 71.93% in SpMKL[H] (i.e., 2.15% improvement), and it has increased from 66.28% in gLMKL[P] to 71.42% in SpMKL[P] (i.e., 5.14% improvement). As presented in Figure 7.4, SpMKL[H] outperformed RF and SVM in 10 out of 15 cohorts, whereas RF was able to outperform SpMKL[H] in only one cohort (i.e., ESCA), and SVM did not outperform SpMKL[H] in any cohorts. Similarly, SpMKL[P] outperformed RF and SVM in 11 and 10 cohorts, respectively, while RF

did not outperform SpMKL[P] in any cohorts, and SVM outperformed SpMKL[P] in only one cohort (i.e., THCA). We also observe that neither SpMKL[H] nor SpMKL[P] were outperformed by any other MKL algorithms, which highlights the advantage of extreme sparsity.

Appendix B.2 presents the performance of these algorithms over the 18 datasets constructed for the second set of experiments E2 (i.e., Stage I or II versus Stage III or IV), which are in agreement with the results obtained in E1, further underscoring the superior predictive performance of SpMKL[H] and SpMKL[P].

Two-year survival analysis. In the two-year survival analysis, the mean AU-ROC value averaged over 20 cohorts has increased from 67.34% in gLMKL[H] to 70.61% in SpMKL[H] (i.e., 3.27% improvement), and it has increased from 67.12% in gLMKL[P] to 72.14% in SpMKL[P] (i.e., 5.02% improvement). As illustrated in Figure 7.5, SpMKL[H] outperformed RF and SVM in 13 and 15 out of 20 cohorts, respectively, whereas RF and SVM were able to outperform SpMKL[H] in only two cohorts (i.e., KIRP and UCEC). By the same token, SpMKL[P] outperformed RF and SVM in 15 cohorts, whereas RF outperform SpMKL[P] in 4 cohorts, and SVM did not outperform SpMKL[P] in any cohorts. As in the first set of experiments, neither SpMKL[H] nor SpMKL[P] were outperformed by any other MKL algorithms, which further highlights the advantage of extreme sparsity.

7.3.2 Biological Mechanisms

Here, we discuss the relationship between cancers and pathways highlighted by the discrimination models for the progression of cancer. Figures 7.6 and 7.7 along with Supplementary Figures B.1, B.4, B.5 and B.6 present the associations between cancer types and pathways identified by SpMKL. Many of the relations indicated in our results have been reported in the literature, showing the validity of our results. Moreover, the relations identified here that have not been studied extensively before can be of use to researchers to find new mechanisms of cancers. As an example, the ANGIOGENESIS pathway can be considered. Tumors, in order to survive and spread, need the ability to form new vessels. The ANGIOGENESIS pathway

takes part in this process. The relevance of this pathway to the head and neck squamous cell carcinoma (HNSC) has been reported in the literature (Carla et al. 2012). Similarly, the HYPOXIA pathway plays a role in the discrimination models for eight cancers specially the head and neck squamous cell carcinoma (HNSC), breast invasive carcinoma (BRCA) and thyroid carcinoma (THCA) (Bredell et al. 2016, Rausch et al. 2017, Mahkamova et al. 2018). In addition, a number of pathways that have been linked to early development of cancer (e.g., KRAS_SIGNALING_DN, MYOGENESIS and SPERMATOGENESIS) contribute heavily to discriminating early- and late-stage cancers (Bergers and Benjamin 2003). Similarly, pathways related to metabolism have been used in the discrimination models for many cancers. These pathways include BILE_ACID_METABOLISM, HEME_METABOLISM, PANCREAS_BETA_CELLS and XENOBIOTIC_METABOLISM. Also, our method is able to identify relations that are limited to a pathway and few cancers. For example, the relation between the pathway REACTIVE_OXYGEN_SPECIES_PATHWAY and the kidney cancer has been identified in the literature (Xiao and Meierhofer 2019). Our results for the survival analysis problem follow similar patterns. For instance, pathways such as KRAS_SIGNALING_DN, ANGIOGENESIS and SPERMATOGENESIS play a role in the prediction models for many cancers. Some of the specific pathway-cohort relationships identified by our methods, for example, the relationship between the COAGULATION pathway and low-grade gliomas (LGG) and the relationship between the ovarian cancer (OV) and the pathway HYPOXIA, have been reported individually in the literature (Ornstein et al. 2002, Liao et al. 2014).

Our results show that using a smaller subset of the pathways has explanatory power about the progression of cancers similar to or greater than that of using all pathways. Some pathways (e.g., MITOTIC_SPINDLE, INTERFERON_GAMMA_RESPONSE, COMPLEMENT) are not critical for discriminating early- and late-stage cancers or two-year survival analysis. Enforcing sparsity on the discrimination models makes identifying such pathways easier.

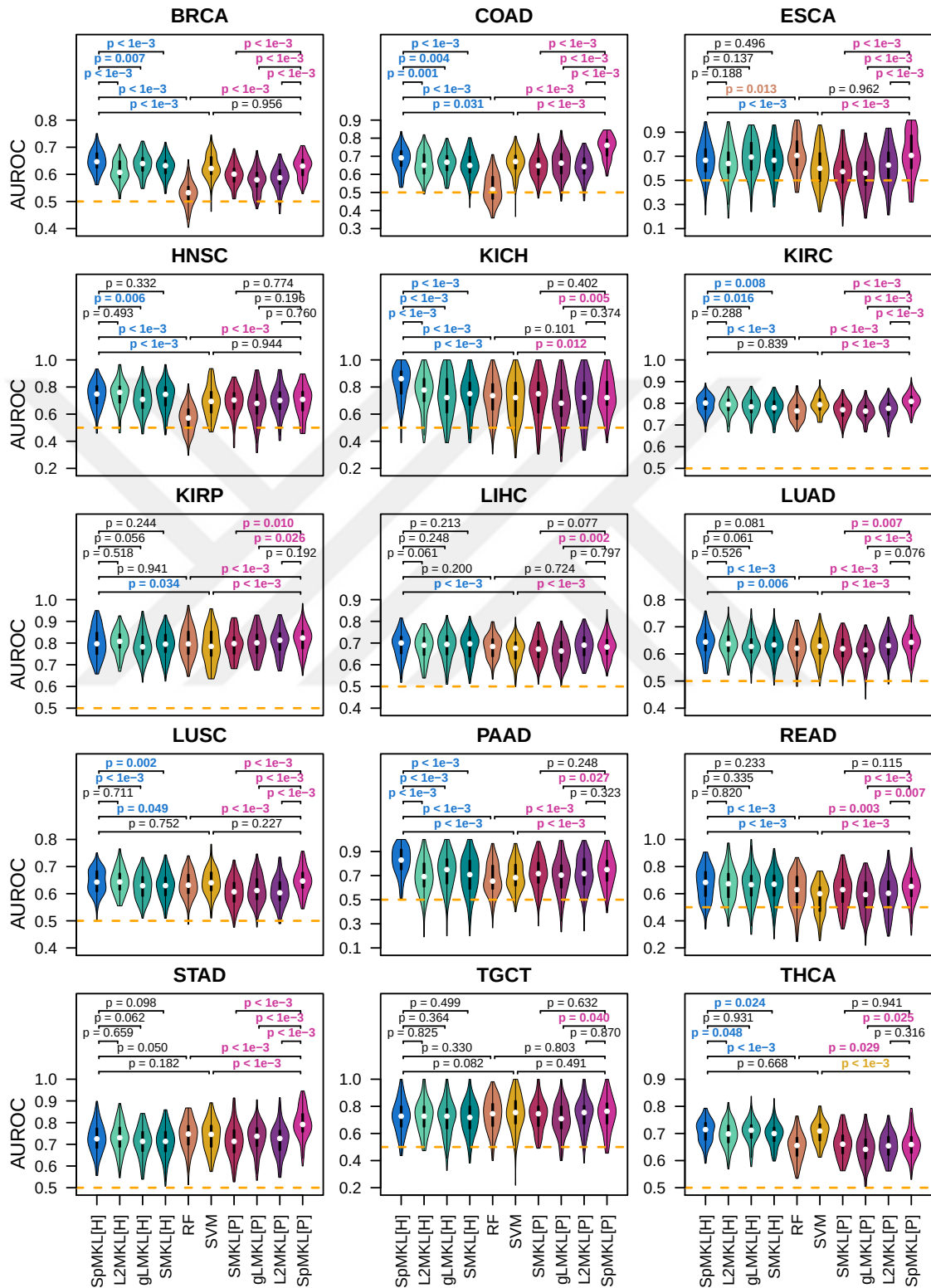


Figure 7.4: Predictive performances of SpMKL[P], SpMKL[H] and other classification algorithms on 15 cancer datasets in E1 constructed from TCGA cohorts.



Figure 7.5: Predictive performances of SpMKL[P], SpMKL[H] and other classification algorithms on 20 cancer datasets in S1 constructed from TCGA cohorts.

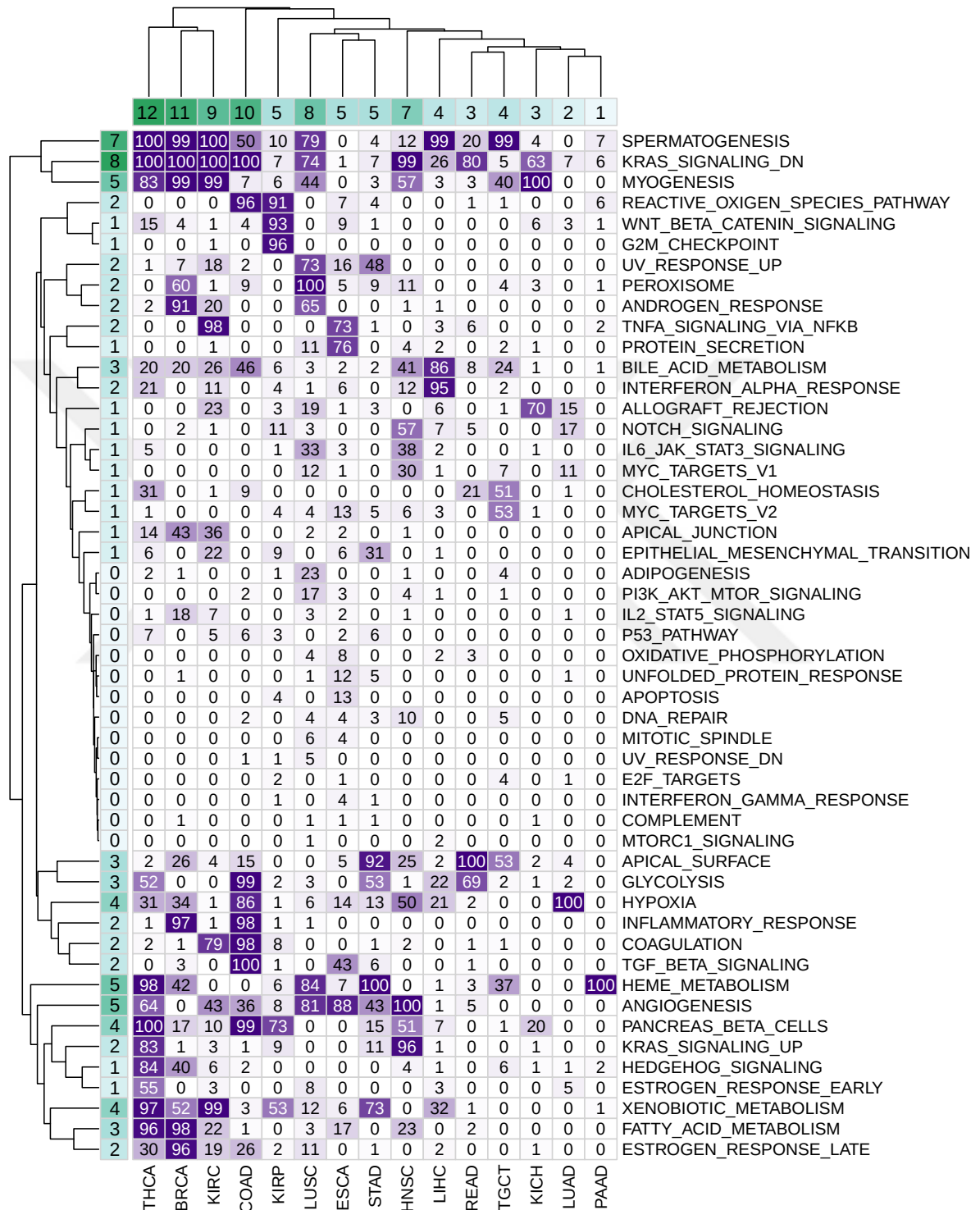


Figure 7.6: Selection frequencies of 50 gene sets in the **Hallmark** collection as identified by **SpMKL[H]** for 15 cancer datasets in **E1**.

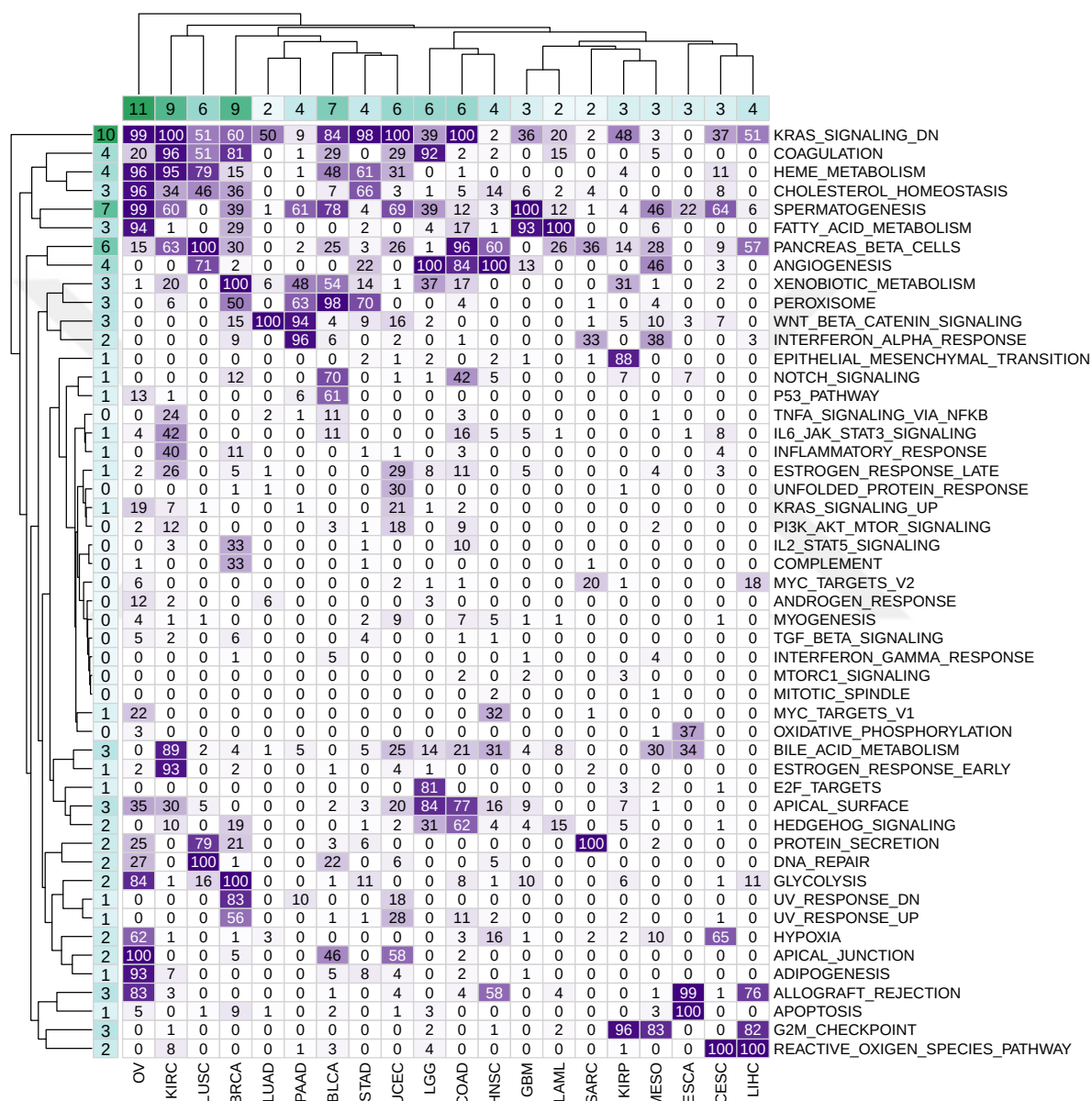


Figure 7.7: Selection frequencies of 50 gene sets in the Hallmark collection as identified by SpMKL[H] for 20 cancer datasets in S1.

7.3.3 Sparsity Analysis

To analyze the sparsity in the kernel weights of SpMKL and other MKL methods, we used the Gini score. The Gini score measures the inequality among values of a variable; in that, higher the score, more dispersed is the data. Therefore, a larger Gini score of kernel weights indicates a more sparse solution. The Gini scores of the kernel weights over 100 replications of each dataset in experiments E1, S1, and E2 are presented in Figures 7.8, 7.9 and B.3, respectively. In these figures, the scores are compared using two-sided paired t -tests, and p -values are color coded to indicate which method is more sparse. As illustrated in these figures, the kernel weights obtained by SpMKL[H] and SpMKL[P] are significantly more sparse than the ones obtained by other MKL algorithm across all experiments. Tables 7.1 7.2 and B.3 also present the number of genes used by each algorithm across E1, S1 and E2, respectively. According to these results, SpMKL[H] and SpMKL[P] use markedly fewer genes in building their classification models, which may be useful in better identifying the most relevant genes that play a critical role in cancer progression.

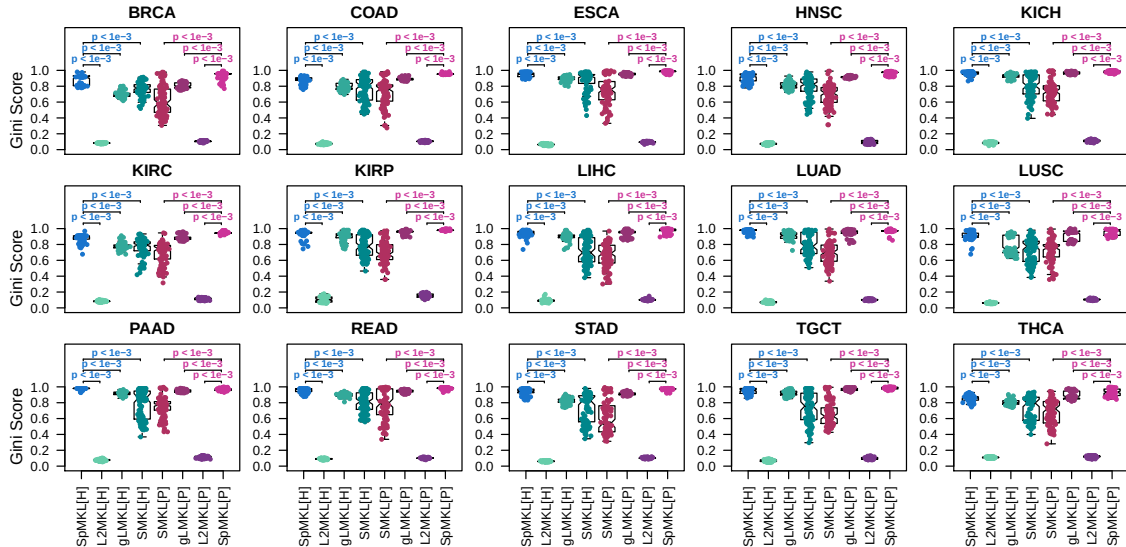


Figure 7.8: Sparsity comparison of SpMKL[P] and SpMKL[H] with other MKL algorithms on 15 cancer datasets in E1.

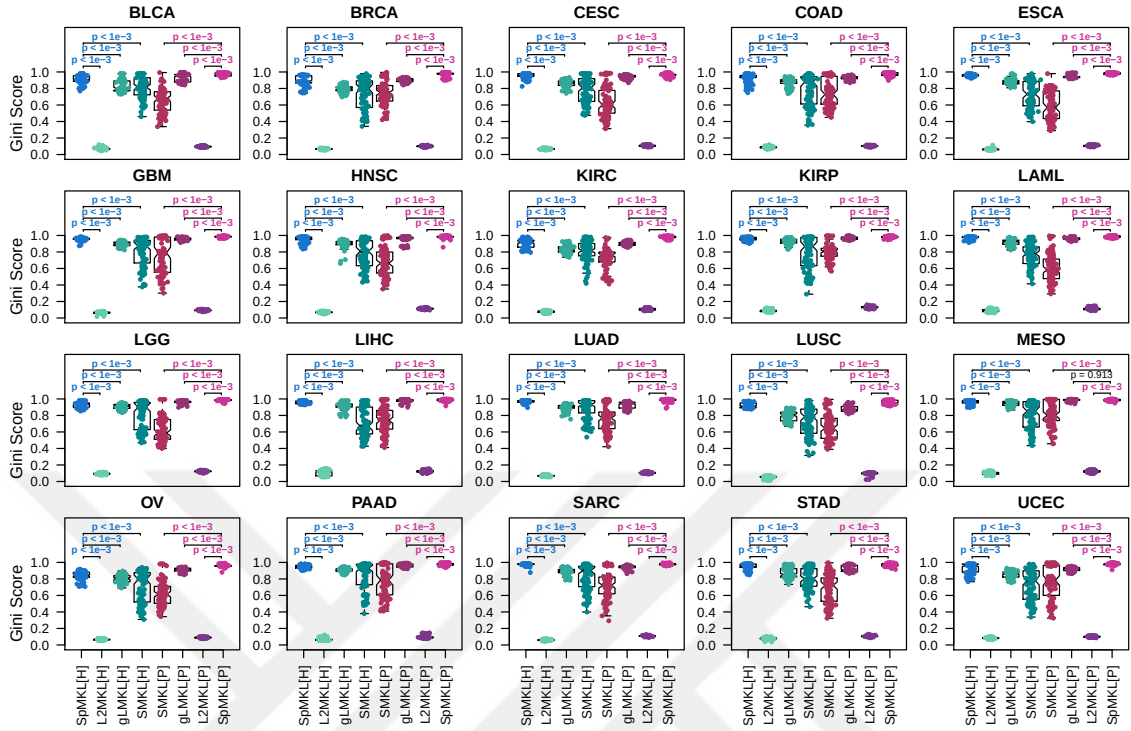


Figure 7.9: Sparsity comparison of SpMKL[P] and SpMKL[H] with other MKL algorithm on 20 cancer datasets in S1.

Table 7.1: Average number of genes used by SpMKL and other classification models over 100 replications across 15 cancer datasets we used in E1.

Cohort	RF & SVM	SMKL [H]	gLMKL [H]	L2MKL [H]	SpMKL [H]	SMKL [P]	gLMKL [P]	L2MKL [P]	SpMKL [P]
BRCA	19,814	2,724	3,340	7,324	1,683	1,431	1,577	8,099	800
COAD	19,814	2,293	2,014	7,324	1,296	1,419	1,048	8,099	541
ESCA	19,814	1,194	1,081	7,324	522	1,158	512	8,099	139
HNSC	19,814	1,991	1,772	7,324	944	1,664	776	8,099	330
KIRC	19,814	2,075	853	7,324	498	1,349	298	8,099	137
KIRP	19,814	2,627	2,602	7,324	1,428	1,631	1,054	8,099	698
KIRP	19,814	2,185	1,153	7,324	570	918	435	8,099	194
LIHC	19,814	2,960	1,214	7,324	588	1,391	549	8,099	264
LUAD	19,814	2,169	1,122	7,324	295	1,350	465	8,099	301
LUSC	19,814	2,929	2,400	7,324	1,133	1,446	762	8,099	448
PAAD	19,814	2,155	1,058	7,324	234	1,386	511	8,099	357
READ	19,814	1,886	1,058	7,324	438	1,005	564	8,099	204
STAD	19,814	2,506	1,882	7,324	783	1,151	884	8,099	380
TGCT	19,814	2,567	851	7,324	473	1,315	291	8,099	112
THCA	19,814	2,759	2,344	7,324	1,784	1,540	1,114	8,099	780
Mean	19,814	2,335	1,650	7,324	845	1,344	723	8,099	379

Table 7.2: Average number of genes used by SpMKL and other classification models over 100 replications across 20 cancer datasets we used in S1.

Cohort	RF & SVM	SMKL [H]	gLMKL [H]	L2MKL [H]	SpMKL [H]	SMKL [P]	gLMKL [P]	L2MKL [P]	SpMKL [P]
BLCA	19,814	1,955	1,944	7,324	982	1,608	658	8,099	395
BRCA	19,814	2,658	2,322	7,324	1,312	1,402	1,032	8,099	390
CESC	19,814	2,230	1,541	7,324	410	1,079	518	8,099	325
COAD	19,814	2,137	1,210	7,324	608	1,374	755	8,099	395
ESCA	19,814	2,867	1,410	7,324	505	1,180	588	8,099	238
GBM	19,814	2,037	1,279	7,324	421	745	439	8,099	177
HNSC	19,814	2,152	999	7,324	377	1,177	423	8,099	169
KIRC	19,814	2,072	1,981	7,324	1,290	1,617	1000	8,099	265
KIRP	19,814	2,209	956	7,324	588	1,366	381	8,099	251
LAML	19,814	2,657	901	7,324	276	1,108	418	8,099	190
LGG	19,814	1,859	878	7,324	640	1,199	396	8,099	180
LIHC	19,814	3,027	964	7,324	533	834	255	8,099	177
LUAD	19,814	1,232	1,232	7,324	177	1,231	637	8,099	148
LUSC	19,814	2,610	1,996	7,324	698	1,463	1,052	8,099	559
MESO	19,814	2,029	726	7,324	416	459	168	8,099	170
OV	19,814	2,997	2,299	7,324	1,867	1,579	850	8,099	477
PAAD	19,814	1,607	1,081	7,324	435	994	380	8,099	210
SARC	19,814	1,778	1,264	7,324	178	1,330	535	8,099	264
STAD	19,814	1,931	1,447	7,324	559	987	657	8,099	331
UCEC	19,814	2,837	1,760	7,324	853	935	852	8,099	300
Mean	19,814	2,244	1,409	7,324	656	1,183	600	8,099	281

7.4 Multitask Multiple Kernel Learning

For MTMKL algorithm, we selected the regularization parameter C and the penalty parameters λ_1 and λ_2 using the four-fold inner cross-validation, where the hyperparameter C was chosen from the set $\{10^{-4}, 10^{-3}, \dots, 10^{+5}\}$, λ_1 was chosen from $\{10^{-1}, 10^0, \dots, 10^{+4}\}$, and λ_2 was set as $\gamma\lambda_1$ for different values of γ chosen from $\{5^{-1}, 5^0, \dots, 5^{+2}\}$. Appendix C.2 presents an analysis of the sensitivity of the MTMKL algorithm to these parameters. For the co-clustering step, we selected the number of clusters K from $\{3, 4, 5\}$ and the number of heterogeneous co-clusters B at each iteration from $\{2, 3, 4, 5\}$. In Algorithm 3, we used the parameters $t_{max} = 400$ and $\zeta = 10^{-6}$ to ensure convergence of the algorithm. As discussed earlier in Section 7.1, for SVM and MKL algorithms, the parameter C was picked using the same cross-validation scheme.

7.4.1 Predictive Performance

On the 15 datasets constructed in the experiment E1 analyzed in this work, we compared the performance of four algorithms: random forests (RF), support vector machine (SVM), single-task multiple kernel learning with hallmark gene sets (MKL[H]) and multitask multiple kernel learning with hallmark gene sets (MTMKL[H]). The predictive accuracies of these algorithms are shown in Figure 7.10. In this figure, the box-and-whisker plots compare the AUROC values of the algorithms over 100 replications. SVM, MKL[H] and MTKML[H] are compared against RF, and MTKML[H] is compared against MKL[H], using a two-tailed paired t -test to check whether there is a significant difference between their performances. The dashed lines show the baseline performance level (i.e., AUROC = 0.5). All methods performed better than the baseline (i.e., AUROC = 0.5), indicating that the genomic information indeed has explanatory power for the stage.

First, we compare the performance of MTKML[H] with RF, which is used as the baseline algorithm in comparisons. Our results show that MTKML[H] performs statistically significantly better than RF in 12 out of 15 cohorts, whereas RF was able to outperform MTKML[H] in only one cohort (i.e., STAD). As discussed by Rahimi and Gönen (2018), SVM and MKL[H] statistically significantly outperform RF in 8 and 7 out of 15 cohorts, respectively, while our new SVM-based method (i.e., MTKML[H]) increases this number to 12 cohorts. These results clearly show that using kernel functions improves the performance by allowing for building more complex and non-linear prediction models. Supplementary results given in Appendix C.3 provide the performance comparison of these algorithms across the 18 datasets created for the second set of experiment E2 (i.e., discriminating Stages I–II from Stages III–IV), which further underscore the benefits of the MTKML algorithm.

To highlight the advantage of multitask learning, we note that MKL[H] (the closest method to MTKML[H]) is outperformed by MTKML[H] in all cohorts. Moreover, the mean AUROC value averaged over 15 cohorts has increased from 69.47% in MKL[H] to 72.53% in MTKML[H] (i.e., more than 3% improvement). This shows that the similarity between different cancer cohorts is meaningful and the similarity matrices

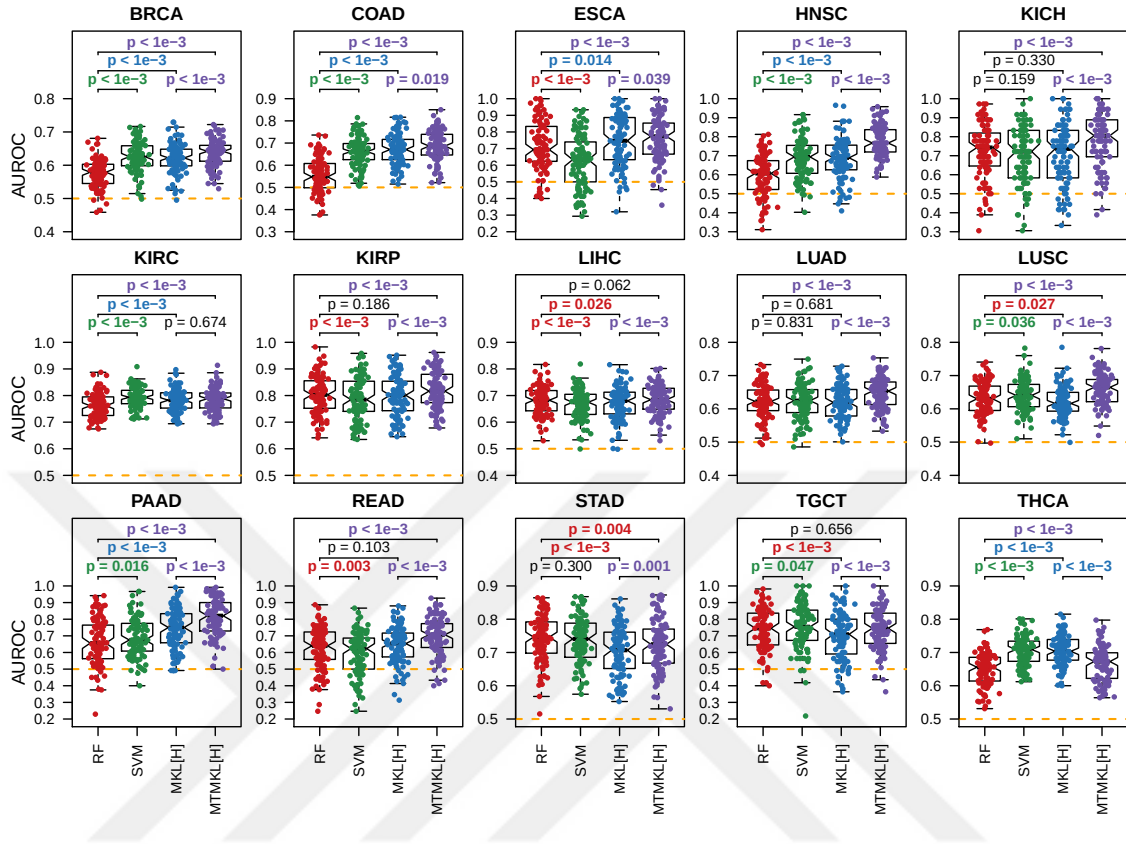


Figure 7.10: Predictive performances of RF, SVM, MKL [H], and MTMKL [H] on 15 cancer datasets constructed from TCGA in E1.

we derived in this work have explanatory power in discriminating early- and late-stage cancers using their genomic data.

There are two main factors that contribute to the overall improved predictive accuracy of MTMKL [H]. First, by looking at the selection frequencies of 50 gene sets in the Hallmark collection for 15 datasets (the leftmost column in Figure 7.11) and comparing the results with the ones given in Figure 7.3, we observe that significantly fewer gene sets for these cohorts are selected by MTMKL [H] compared to the MKL [H] model. This, in turn, results in a more sparse classifier, which is less likely to over-fit to the training data.

On the other hand, comparing the predictive performance of MTMKL [H] with MKL [H] (Figure 7.10), we observe that MTMKL [H] is able to improve the accuracy for smaller size cohorts (e.g., ESCA, KICH, KIRP, PAAD, READ and TGCT) without

compromising the accuracy of other cohorts. This is expected, as the purpose of multitask learning is to help improve the accuracy of the tasks with limited data using the tasks with abundant data.

7.4.2 Biological Mechanisms

The relationships between cohorts and pathways identified in our results are in agreement with many studies that investigate the effect of few pathways on certain cancer types. Figure 7.11 illustrates the relationships identified by our algorithm. For example, the relationships between the pathways `EPITHELIAL_MESENCHYMAL_TRANSITION`, `PI3K_AKT_MTOR_SIGNALING` and `ESCA` cohort were explored by Zhang et al. (2014). Irregularities in these pathways can increase the mobility of cells that are supposed to form walls around organs, consequently increasing the risk of metastasis. Since esophagus is populated with such cells, it is more vulnerable to the adverse effect of dysfunction of these pathways. Feng et al. (2018) described the relationships between `ANGIOGENESIS` and `STAD` cohort, they also identify other pathways such as `EPITHELIAL_MESENCHYMAL_TRANSITION` and `REACTIVE_OXYGEN_SPECIES_PATHWAY` as contributing factors. The contribution of these pathways to `STAD` cohort is further supported by our results. Also, gene sets that were expected to be irrelevant for identifying progression of cancer into late-stage were not selected by MTMKL algorithm. This is also in line with the results obtained by the models from the other chapters. Such pathways include cell cycle related gene sets such as `DNA_REPAIR`, `E2F_TARGETS`, `MYC_TARGETS_V1` and `MTORC1_SIGNALING`.

We also derived a similarity measure between cancer cohorts based on co-clustering. This similarity measure is illustrated in Figure 5.2. These results are also in line with the previous studies on the subject. Similarities between `COAD` and `READ` cohorts were discussed by Ylipää et al. (2013) and Khirade et al. (2015). Yang et al. (2013) described a case-study of `THCA` prompted by `BRCA` metastasis and suggests that other cases of `THCA` have to be tested for ruling out their tissue of origin being the chest area. Later, Yu et al. (2017) performed a retrospective study and concluded that in fact these two cohorts occur together more frequently than expected

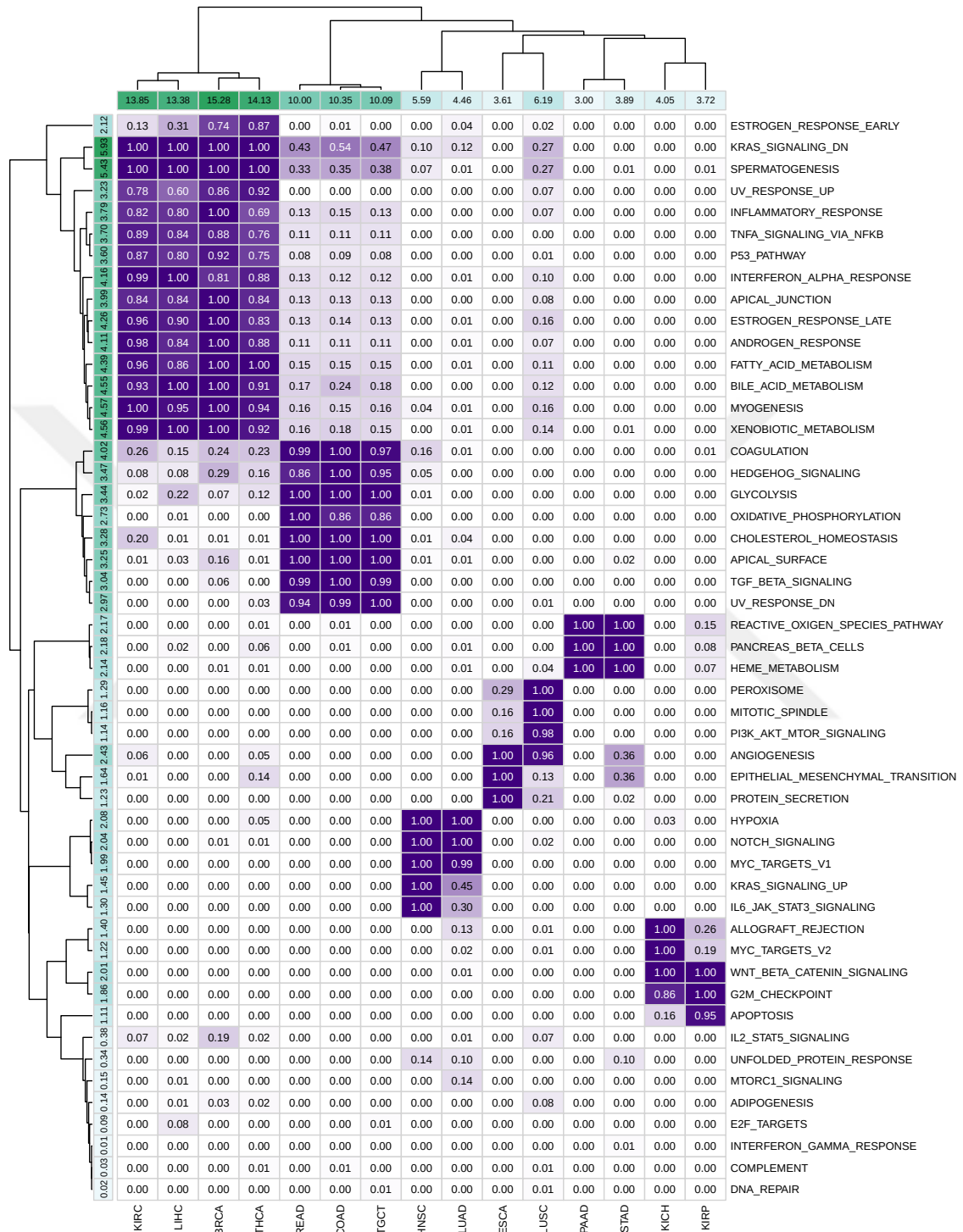


Figure 7.11: Selection frequencies of 50 gene sets in the Hallmark collection identified by MTMKL [H] for the 15 cancer datasets in experiment E1.

and the underlying factor might be mutations in estrogen receptor genes. The relation between THCA and BRCA was also reported by Nielsen et al. (2016). Reznik

and Sander (2015) reported similarities between BRCA and KIRC cohorts based on metabolism related genes. Finally, association of ESCA and LUSC was documented by Fekete et al. (1994).

7.4.3 Computational Complexity

Next, we analyze the time complexity of the models. Figure C.1 presents the cohort-based time comparison of the three single-task models (i.e., RF, SVM and MKL[H]) as well as the comparison of overall time of all four models over all cohorts. We observe that despite the high computation time of MTMKL[H] in the cross-validation phase, computation time of this model is significantly lower than that of all three single-task models in the test phase (to be explicit, MTMKL[H] is on the average 6 times faster than MKL[H], 4.4 times faster than SVM, and 4.1 times faster than RF). The high cross-validation time of MTMKL[H] is due to the large number of hyper-parameter evaluations in the cross-validation phase. In particular, MTMKL[H] requires evaluating $10 \times 6 \times 4$ choices of $(C, \lambda_1, \lambda_2)$, whereas SVM and MKL[H] require evaluating 10 choices of C and RF requires evaluating 5 levels of parameter `ntree`. The lower test time, on the other hand, underscores the efficiency of the proposed cutting-plane method presented in Algorithm 3 for solving the MTMKL problem. The comparative advantage of the cutting-plane algorithm is due to its ability to generate bounds on the objective value that help certify the optimality of a solution as soon as the bounds become sufficiently close.

7.5 Efficient Multitask Multiple Kernel Learning

To compare the time efficiency of the algorithms, the algorithms were terminated after 12 hours of running time or reaching an optimality gap of $\zeta = 0.01\%$. For our final set of experiments on the 15 TCGA cohorts, we used 4000 iterations or $\zeta = 0.01\%$ as the termination criteria of Algorithm 9.

7.5.1 Computational Complexity

In this section, we compare the performance of different versions of the proposed algorithms under various parameter settings. In total, we chose seven versions of the algorithms: one version for Algorithm 5 (VL), two versions for Algorithm 6 based on whether $\bar{J}_s$ values are computed using Proposition 3 (OLJ1) or Proposition 4 (OLJ2), and four versions for Algorithm 9 (BDF, BDF-P0, BDFS, BDFS-P0) which are combinations of selection of cumulative/strong formulations and normal/Pareto-optimal cuts. To compare these algorithms, we chose the number of tasks from $T \in \{7, 11\}$, the number of clusters from $K \in \{2, 4\}$ and the regularization parameter from $C \in \{10^{-2}, 10^{-1}, \dots, 10^6\}$.

We start with comparing two versions of Algorithm 6 (i.e., OLJ1 and OLJ2) as illustrated in Figure 7.12. In this figure, each point represents the median time/gap of 50 replications. The red horizontal lines show the time limit and dashed segments indicate that at least 25% of the replications of the corresponding instances have reached the time limit of 12 hours. We observe from these results that the time required for computing $\bar{J}_s$ using Proposition 4 is not compensated by the improved convergence of Algorithm 6, thus OLJ1 turns out to be the more time-efficient version of this algorithm.

Performance of different versions of Algorithm 9 (i.e., BDF, BDF-P0, BDFS, and BDFS-P0) are summarized in Figure 7.13. In this figure, each point represents the median time/gap of 50 replications. The red horizontal lines show the time limit and dashed segments indicate that at least 25% of the replications of the corresponding instances have reached the time limit of 12 hours. These graphs suggest the positive effect of the proposed acceleration techniques on the performance of Algorithm 9. More specifically, the strong formulation of FMP (SFMP) substantially improves the performance of Algorithm 9. Additionally, employing Pareto-optimal cuts further improves the convergence of Algorithm 9. Hence, in the following, we use BDFS-P0 as the best version of Algorithm 9.

We next compare the performance of Algorithm 5 (VL), Algorithm 6 (OLJ1) and Algorithm 9 (BDFS-P0) under a wider range of T and K . To showcase the effect of

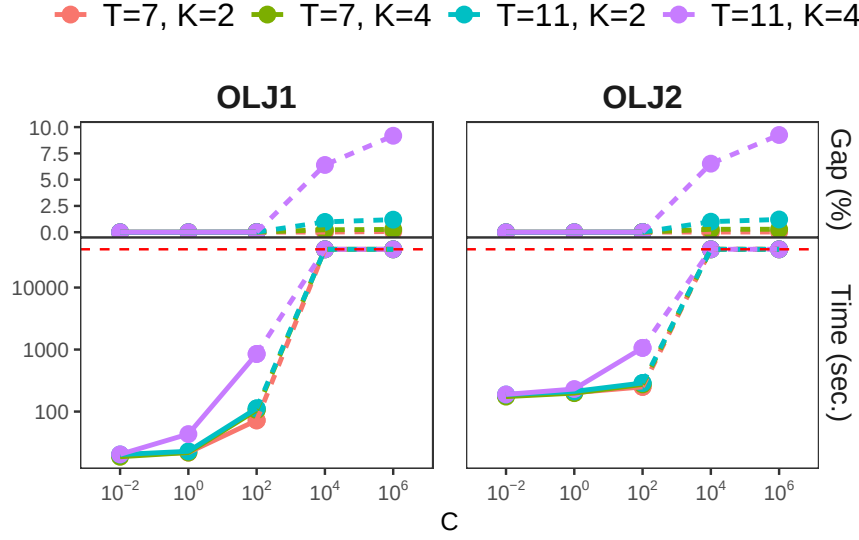


Figure 7.12: Performance of two versions of Algorithm 6 (OLJ1 and OLJ2) under different values of T , K , and C .

parameter C , in Figure 7.14, we present the performance of the selected versions of each algorithm under $C \in \{10^{+1}, 10^{+3}, 10^{+5}\}$. As before, each point represents the median time/gap of 50 replications. The red horizontal lines show the time limit and dashed segments indicate that at least 25% of the replications have reached the time limit of 12 hours. Recall that as T , K , and C grow, the problem becomes increasingly more difficult. When C is small, overall, the performance of three algorithms are close, and all algorithms are able to optimally solve the problem instances within the given optimality threshold (note that the largest gap is only 0.05%). As instances become more challenging, simple linearization models VL and OLJ1 are unable to close the gap within the allotted time limit of 12 hours, and BDForest (BDFS-P0) substantially becomes the superior of the three algorithms. Consequently, we employ BDFS-P0 for our final computational experiments on the 15 cancer cohorts and different number of clusters.

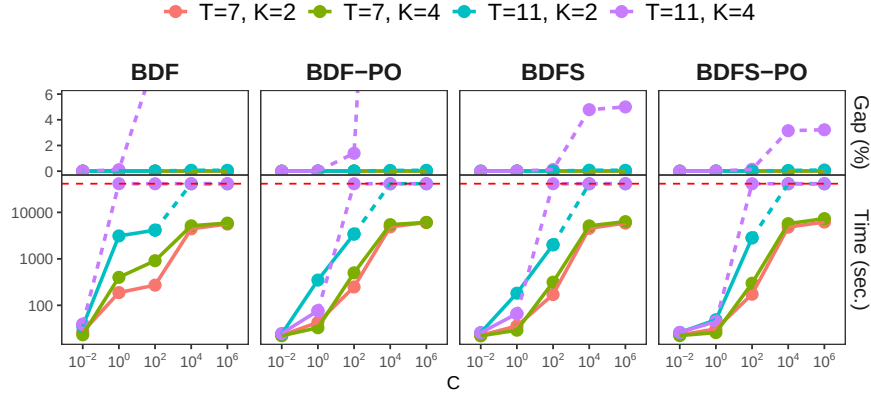


Figure 7.13: Four versions of Algorithm 9 (BDForest) across different values of T and K .

7.5.2 Results on the TCGA Datasets

In our final set of experiments, we solved 100 replications of the MTMKLC problem on the 15 datasets constructed for E1 using the BDForest algorithm. To produce interpretable results, in each replication, we selected the number of clusters K from the set $\{3, \dots, 7\}$ and the value of parameter C from the set $\{10^0, 10^1, \dots, 10^6\}$ using a four-fold inner cross-validation scheme.

Figure 7.15 illustrates the similarities between the cohorts based on these experiments, which are calculated as the average grouping frequency of each pair of cohorts (tasks) over 100 replications. In this figure, the frequencies are scaled between 0 and 1 for the sake of interpretability. These results indicate the existence of two large groups of cohorts, where one of the groups breaks down into smaller groups such as a group consisting of COAD and READ, a group consisting of KIRP and THCA and a group consisting of KIRC, BRCA, KICH and LIHC. Some of these groups correspond to similarities reported in the literature. For example, the similarity between the cohorts COAD and READ is reported by Ylipää et al. (2013). Yang et al. (2013) hypothesizes that BRCA can be prompted by THCA. Xia et al. (2019) reports similarities between BRCA, HNSC, KIRP and THCA in terms of biochemical pathways in line with our results that put these cohorts in the same large group. Also similarities between BRCA and KIRC have been reported in the literature (Reznik and Sander 2015).

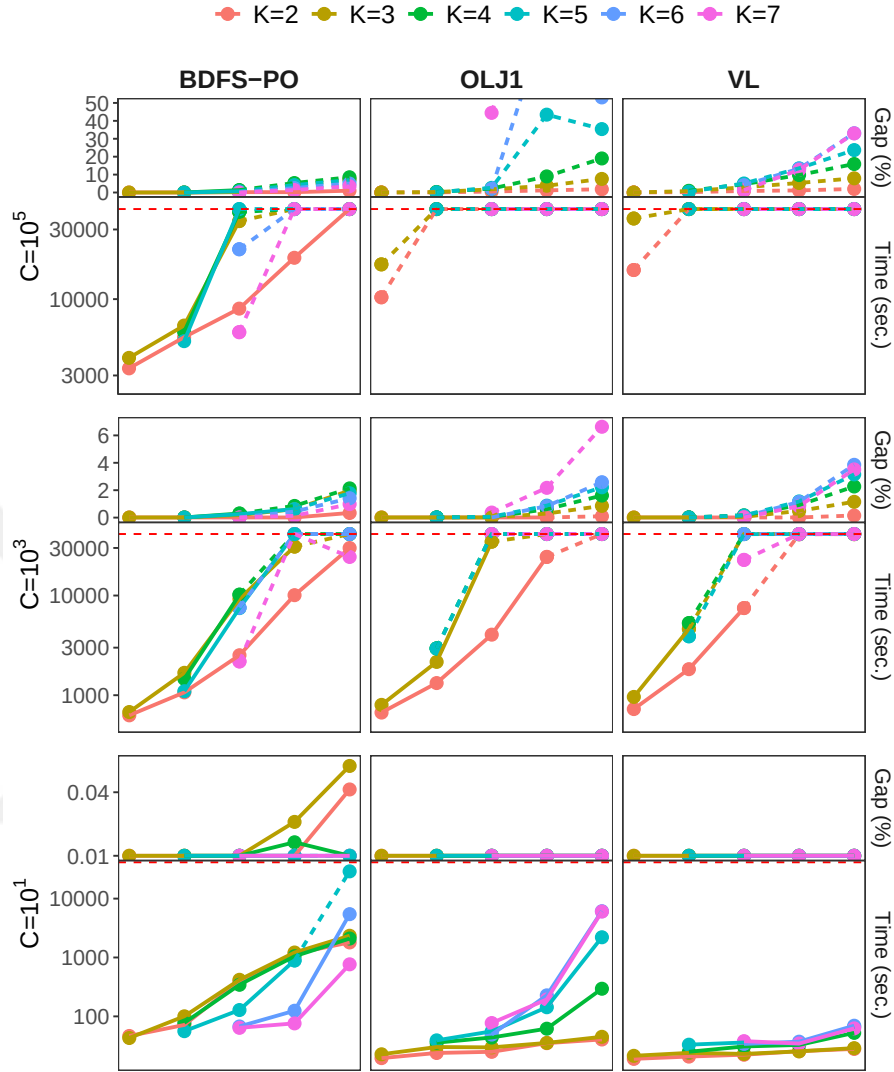


Figure 7.14: Performance of Algorithms 5 (VL), 6 (OLJ1) and 9 (BDFS-P0) across different number of tasks and clusters.

For predictive performance comparison on the TCGA datasets, we employed the two-stage MTMKL scheme proposed in Chapter 5, which requires a matrix representing the similarity between each pair of cohorts and a matrix representing association between cohorts and gene sets. For the cohort-cohort similarity matrix, we used the matrix presented in Figure 7.15. To obtain a cohort-pathway association matrix, we performed co-clustering on the kernel profiles obtained by 100 replications of the MTMKLC problem to obtain 100 assignment matrices. We then

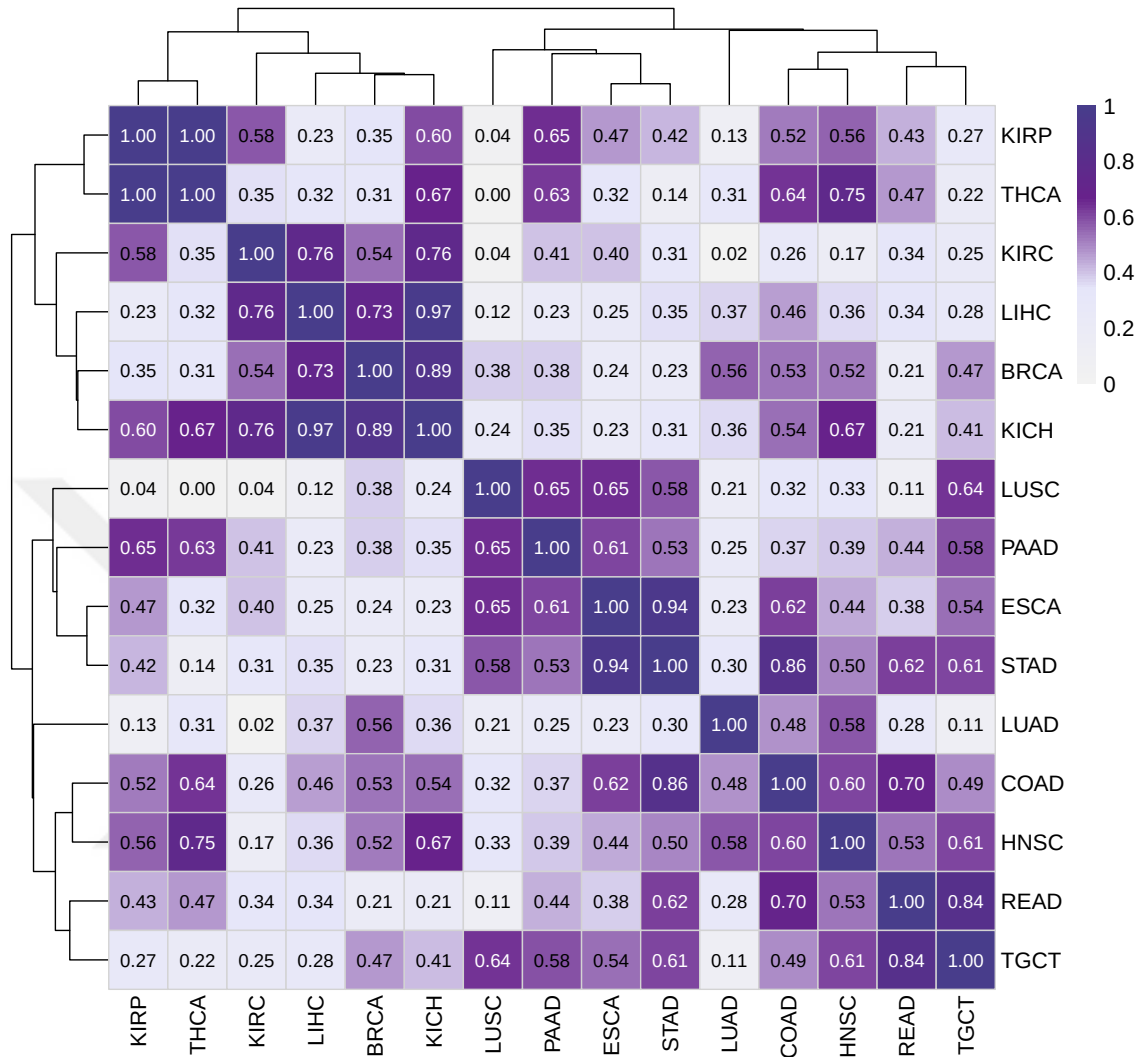


Figure 7.15: Grouping frequencies between 15 TCGA cohorts over 100 replications of the MTMKLC problem.

computed the cohort-pathway association matrix by taking the average of these assignment matrices. Finally, we passed these two matrices to the second stage MTMKL presented in Chapter 5 to obtain the AUROC values.

We compared the performance of five algorithms: random forests (RF), support vector machine with Gaussian kernel (SVM), single-task multiple kernel learning with **Hallmark** gene sets (MKL[H]) as presented in Chapter 3, multitask multiple kernel learning with **Hallmark** gene sets (MTMKL[H]) as presented in Chapter 5, and the results obtained in this chapter using the two-stage scheme described earlier (BDF[H]).

The predictive accuracies of these algorithms are shown in Figure 7.16. In this figure, the box-and-whisker plots compare the AUROC values of the algorithms over 100 replications. SVM, MKL[H], MTMKL[H], and BDF[H] are compared against RF, and BDF[H] is compared against SVM, MKL[H], and MTMKL[H], using the two-tailed paired t -test to check whether there is a significant difference between their performances. The dashed lines show the baseline performance level (i.e., AUROC = 0.5). p -values are color coded to indicate which method performs better in the t -test. All methods performed better than the baseline (i.e., AUROC = 0.5), indicating once more that the genomic information indeed has explanatory power for progression of cancer. BDF[H] predominantly outperformed RF, SVM, and MKL[H], and maintained comparable performance compared to MTMKL[H], which highlights the added value of multitask learning in improved accuracies.

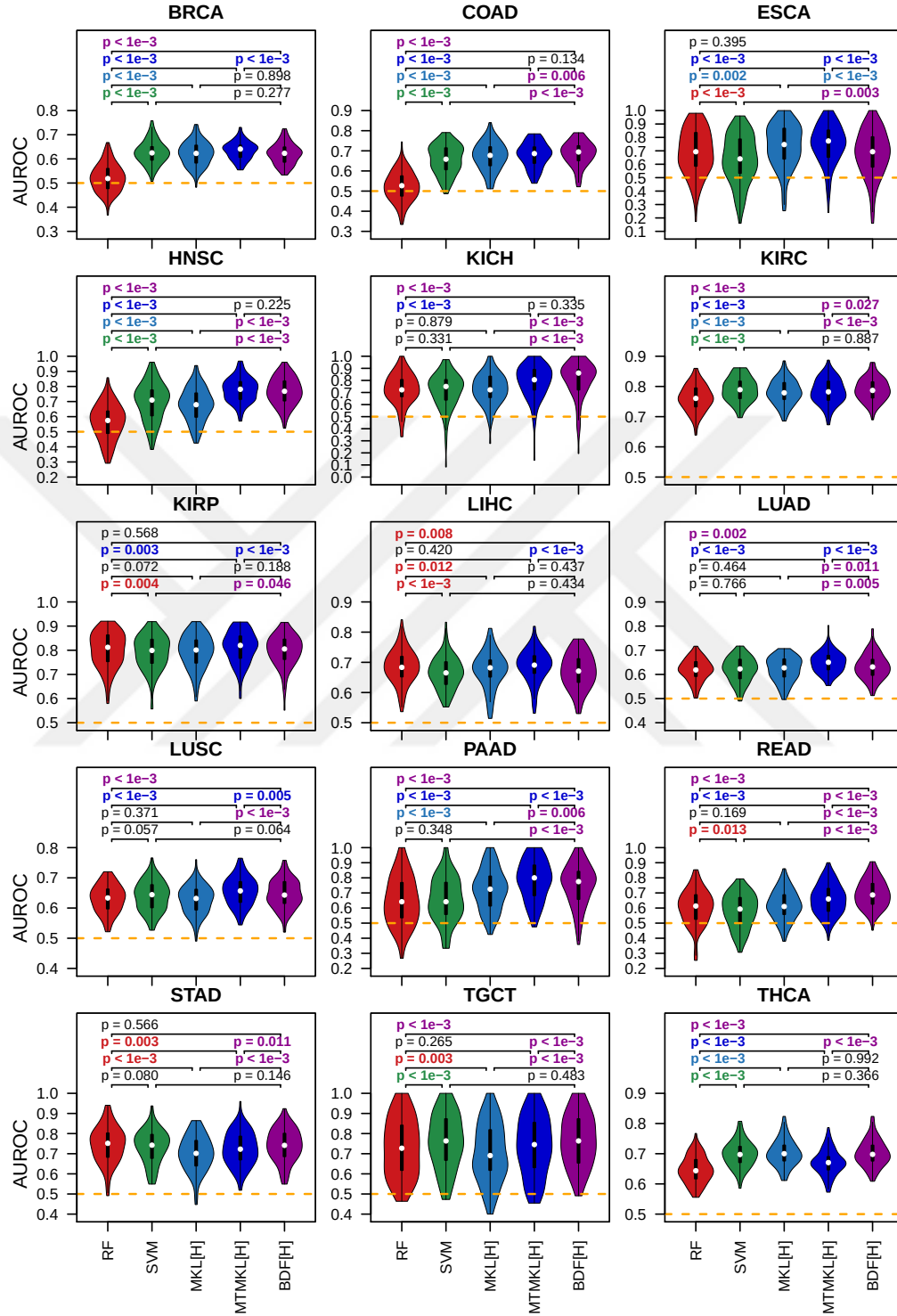


Figure 7.16: Predictive performances of MTMKL [H], BDF [H], and other classification methods on 15 datasets constructed from TCGA in E1.

Chapter 8

CONCLUSIONS

This thesis investigates the problem of generating classification models for cancer biology, with a focus on interpretability. To this end, the first step is integration of pathways into the classification models. Compared to genes, pathways more directly reflect the biological processes taking place in an organism. Furthermore, the number of pathways is much smaller compared to the number of genes. The integration of pathways into the classification models is achieved using kernel functions. For each pathway, a kernel function is built and the classification algorithm chooses the optimal combination of these kernel functions for explaining the difference between the two classes. This approach generates models that are significantly more interpretable and more accurate compared to the models generated without integration of pathways.

Despite the success of the models generated by the integration of pathways, not all the models generated for all the cancers have the same level of clarity. Some cancers seem to depend on a couple of pathways while others use several pathways. This is attributed to the inherent heterogeneity of cancers, namely that the mechanisms or identities of various cancers might correspond to factors more complicated than just their tissue of origin. However, another contributor to this disparity between the classification models for various cancers is the mechanism with which sparsity is enforced on the models. Traditionally, sparsity is forced on the models by indirect means, i.e., through regularization norms. However, once the desired sparsity exceeds a threshold, the computational efficiency of these norms is lost. In addition, this approach does not guarantee a predetermined level of sparsity and the sparsity of the solution can be assessed only when the training is complete. To remedy these issues, we developed a sparse MKL algorithm that is able to deliver a predefined

level of sparsity. Given that the problem is essentially selecting a given number of kernel functions that can best explain the data, and as such involves optimization over a large discrete set, we employed a cutting-plane algorithm to facilitate solving the problem efficiently. Through empirical experiments, we show how this approach delivers more accurate and more sparse solutions.

Next, we extended the models mentioned so far to a multitask setting. The similarities of various cancers has been widely known and studied. However, a commonly used measure of similarity between the cancers is not available, specifically since such a measure must be application-specific since the mechanisms of the generation of a primary tumor are not necessarily identical with the mechanisms that cause its progression to late-stages. For these reasons, we developed a method that is able to derive a data-driven measure of similarity and learn the classification models simultaneously. More specifically, the method generates clusters of cohorts and pathways in a co-clustering setting and penalizes the classification models that utilize associations between members of different clusters. This approach increases the accuracy and the interpretability of the classification models. Since the co-clustering problem is combinatorial in nature, the resulting problem is hard to solve and we have employed a heuristic method for solving it efficiently.

Given that exact solution methods often give more robust results and pave the way for solving more complicated problems using the exact methods as building blocks, we developed an exact solution method for the problem of multitask multiple kernel classification by introducing the forest formulation, in which clusters of tasks are represented as disjoint tree structures in a graph, and devising a tailored accelerated Benders decomposition algorithm for solving instances of this problem. We show that the tree structure of the clusters helps eliminate redundancies in the solution space. The problem of assigning the tasks to various groups while optimizing the multiple kernel classification models is highly non-linear, degenerate and involves a large number of integer variables. The non-linearity of the problem is eliminated by reformulating the problem as a semi-infinite integer optimization problem. The inherent bi-level structure of the multitask multiple kernel learning

problem allows us to handle the choice of assigning tasks to clusters in the Benders master problem, and delegate the decision on kernel weights to subproblems. With the help of acceleration techniques and generating strong Pareto-optimal Benders cuts, we developed the BDForest algorithm, and demonstrated its computational efficiency in solving large-scale instances of the multitask multiple kernel problem through empirical experiments.

This work can be extended in several directions. The recent advances in sequencing technology have paved the way for new streams of research related to genomic data. The new data sources generated by these methods are closely related to expression data. Examples of these datasets include epigenomics data, single cell transcriptomics and spatial information alongside genomic data. The data from these fields can be readily integrated into the classification models for cancer using the multiple kernel framework.

Another direction for extending this work can be building more accurate models using more accurate representations of the pathways. In this work, we have viewed pathways as gene sets. Pathways can be represented more accurately as graphs where the genes constitute the nodes and their interactions constitute the graph edges. This representation can be more accurate compared to the set representation but still it might not fully reflect all the available data about the pathways. Lastly, the data that has been gathered to identify individual pathways can be integrated into the classification models directly. The maximal accuracy of pathway representation provided by this approach requires more advanced mathematical formulations and solution algorithms allowing for learning the pathways as complex functions of the transcriptomic data while learning classification models based on these same pathways.

Lastly, the large scale optimization methods developed here, can be extended by developing an efficient optimization method for clustering tasks into soft clusters. In this setting, a task can potentially belong to several clusters with various degrees of membership. This approach can provide a better penalty structure for multi-task learning with many tasks as the similarity pattern between various tasks might

be too complex for the similar/dissimilar dichotomy. This variation of the multi-task multiple kernel problem can potentially be solved using the column generation method where each cluster of tasks corresponds to a new column.



BIBLIOGRAPHY

- Andreev, K. and Racke, H. (2006). Balanced graph partitioning. *Theory Comput. Syst.*, 39(6):929–939.
- Bai, Z. et al. (2014). Sparse extreme learning machine for classification. *IEEE Trans. Cybern.*, 44(10):1858–1870.
- Benders, J. (1962). Partitioning procedures for solving mixed-variables programming problems ‘. *Numerische Mathematik*, 4(1):238–252.
- Bergers, G. and Benjamin, L. E. (2003). Tumorigenesis and the angiogenic switch. *Nat. Rev. Cancer*, 3(6):401–410.
- Bertsimas, D. et al. (2017a). From predictive methods to missing data imputation: an optimization approach. *J. Mach. Learn. Res.*, 18(1):7133–7171.
- Bertsimas, D. and King, A. (2015). OR forum - an algorithmic approach to linear regression. *Oper. Res.*, 64(1):2–16.
- Bertsimas, D., Pauphilet, J., and Van Parys, B. (2017b). Sparse classification and phase transitions: A discrete optimization perspective. *arXiv preprint arXiv:1710.01352*.
- Bhalla, S. et al. (2017). Gene expression-based biomarkers for discriminating early and late stage of clear cell renal cancer. *Sci. Rep.*, 7:44997.
- Bredell, M. G. et al. (2016). Current relevance of hypoxia in head and neck cancer. *Oncotarget*, 7(31):50781.
- Breiman, L. (2001). Random forests. *Mach. Learn.*, 45(1):5–32.
- Broët, P. et al. (2006). Identifying gene expression changes in breast cancer that distinguish early and late relapse among uncured patients. *Bioinformatics*, 22(12):1477–1485.

- Carla, C. et al. (2012). Angiogenesis in head and neck cancer: a review of the literature. *J. Oncol.*, 2012.
- Caruana, R. (1997). Multitask learning. *Mach. Learn.*, 28(1):41–75.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Mach. Learn.*, 20(3):273–297.
- Díaz-Uriarte, R. and De Andres, S. A. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinform.*, 7(1):3.
- Ein-Dor, L. et al. (2005). Outcome signature genes in breast cancer: Is there a unique set? *Bioinformatics*, 21(2):171–178.
- Ein-Dor, L. et al. (2006). Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proc. Natl. Acad. Sci. USA*, 103(15):5923–5928.
- Fekete, F. et al. (1994). Associated primary esophageal and lung carcinoma: a study of 39 patients. *Ann. Thorac. Surg.*, 58(3):837–842.
- Feng, Y. et al. (2018). Association between angiogenesis and cytotoxic signatures in the tumor microenvironment of gastric cancer. *OncoTargets Ther.*, 11:2725–2733.
- Geoffrion, A. M. (1972). Generalized Benders decomposition. *J. Optim. Theory Appl.*, 10(4):237–260.
- Gönen, M. (2012). Predicting drug–target interactions from chemical and genomic kernels using bayesian matrix factorization. *Bioinformatics*, 28(18):2304–2310.
- Gönen, M. and Alpaydm, E. (2011). Multiple kernel learning algorithms. *J. Mach. Learn. Res.*, 12(Jul):2211–2268.
- Gönen, M. and Margolin, A. A. (2014). Localized data fusion for kernel k-means clustering with application to cancer biology. In *Advances in Neural Information Processing Systems*, pages 1305–1313.
- Gress, D. M. et al. (2017). Principles of cancer staging. *AJCC Cancer Staging Manual*, 8:3–30.

- Gu, X. et al. (2014). Multitask coupled logistic regression and its fast implementation for large multitask datasets. *IEEE Trans. Cybern.*, 45(9):1953–1966.
- Hanahan, D. and Weinberg, R. A. (2000). The hallmarks of cancer. *Cell*, 100(1):57–70.
- Ishwaran, H. and Kogalur, U. B. (2017). *Random Forests for Survival, Regression, and Classification (RF-SRC)*, R package version 2.5.1.
- Jacob, L., Vert, J.-p., and Bach, F. R. (2009). Clustered multi-task learning: A convex formulation. In *Advances in Neural Information Processing Systems*, pages 745–752.
- Jagga, Z. and Gupta, D. (2014). Classification models for clear cell renal carcinoma stage progression, based on tumor RNA-seq expression trained supervised machine learning algorithms. In *BMC Proc.*, volume 8, page S2. BioMed Central.
- Ji, G. et al. (2010). PLS-based gene selection and identification of tumor-specific genes. *IEEE Trans. Syst. Man Cybern. Part C-Appl. Rev.*, 41(6):830–841.
- Jin, J. (2009). Impossibility of successful classification when useful features are rare and weak. *Proc. Nat. Acad. Sci.*, 106(22):8859–8864.
- Khirade, M. F. et al. (2015). Derivation of a fifteen gene prognostic panel for six cancers. *Sci. Rep.*, 5:13248.
- Kloft, M. et al. (2009). Efficient and accurate ℓ_p -norm multiple kernel learning. In *Advances in Neural Information Processing Systems*, pages 997–1005.
- Li, M. and Zhou, Z.-H. (2007). Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Trans. Syst. Man Cybern. Part A-Syst. Hum.*, 37(6):1088–1098.
- Liao, J. et al. (2014). Ovarian cancer spheroid cells with stem cell-like properties contribute to tumor generation, metastasis and chemotherapy resistance through hypoxia-resistant metabolism. *PloS One*, 9(1):e84941.
- Liberzon, A. et al. (2015). The Molecular Signatures Database (MSigDB) hallmark gene set collection. *Cell Syst.*, 1(6):417–425.

- Magnanti, T. L. and Wong, R. T. (1981). Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Oper. Res.*, 29(3):464–484.
- Mahkamova, K. et al. (2018). Hypoxia increases thyroid cancer stem cell-enriched side population. *World J. Surg.*, 42(2):350–357.
- Manica, M. et al. (2019). PIMKL: Pathway-induced multiple kernel learning. *NPJ Syst. Biol. Appl.*, 5(1):1–8.
- MOSEK ApS (2017). *MOSEK Optimization Suite Release 8.1.0.34*.
- Muller, K.-R. et al. (2001). An introduction to kernel-based learning algorithms. *IEEE Trans. Neural Netw. Learn. Syst.*, 12(2):181–201.
- Nam, H. et al. (2009). Combining tissue transcriptomics and urine metabolomics for breast cancer biomarker identification. *Bioinformatics*, 25(23):3151–3157.
- Nascimento, A. C. et al. (2016). A multiple kernel learning algorithm for drug-target interaction prediction. *BMC Bioinform.*, 17(1):46.
- Nielsen, S. M. et al. (2016). The breast–thyroid cancer link: A systematic review and meta-analysis. *Cancer Epidemiol. Biomark. Prev.*, 25(2):231–238.
- Ornstein, D. L. et al. (2002). The coagulation system as a target for the treatment of human gliomas. In *Semin. Thromb. Hemost.*, volume 28, pages 19–28.
- Pan, S. et al. (2016). Task sensitive feature exploration and learning for multitask graph classification. *IEEE Trans. Cybern.*, 47(3):744–758.
- Pang, H. et al. (2006). Pathway analysis using random forests classification and regression. *Bioinformatics*, 22(16):2028–2036.
- Rahimi, A. and Gönen, M. (2018). Discriminating early-and late-stage cancers using multiple kernel learning on gene sets. *Bioinformatics*, 34(13):i412–i421.
- Rahmaniani, R. et al. (2017). The Benders decomposition algorithm: A literature review. *Eur. J. Oper. Res.*, 259(3):801–817.
- Rakotomamonjy, A., Bach, F. R., Canu, S., and Grandvalet, Y. (2008). SimpleMKL. *J. Mach. Learn. Res.*, 9(Nov):2491–2521.

- Rausch, L. K. et al. (2017). The linkage between breast cancer, hypoxia, and adipose tissue. *Front. Oncol.*, 7:211.
- Ren, X. et al. (2012). ellipsoidFN: a tool for identifying a heterogeneous set of cancer biomarkers based on gene expressions. *Nucleic Acids Res.*, 41(4):e53–e53.
- Reznik, E. and Sander, C. (2015). Extensive decoupling of metabolic genes in cancer. *PLoS Comput. Biol.*, 11(5):e1004176.
- Rifai, N. et al. (2006). Protein biomarker discovery and validation: the long and uncertain path to clinical utility. *Nat. Biotechnol.*, 24(8):971–983.
- Safavian, S. R. and Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybern. Part C-Appl. Rev.*, 21(3):660–674.
- Schaefer, C. F. et al. (2009). PID: The pathway interaction database. *Nucleic Acids Res.*, 37(suppl_1):D674–D679.
- Sehhati, M. et al. (2015). Stable gene signature selection for prediction of breast cancer recurrence using joint mutual information. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 12(6):1440–1448.
- Seoane, J. A. et al. (2014). A pathway-based data integration framework for prediction of disease progression. *Bioinformatics*, 30(6):838–845.
- Statnikov, A., Wang, L., and Aliferis, C. F. (2008). A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC Bioinform.*, 9(1):319.
- Telonis, A. G. et al. (2017). Knowledge about the presence or absence of miRNA isoforms (isomiRs) can successfully discriminate amongst 32 TCGA cancer types. *Nucleic Acids Res.*, 45(6):2973–2985.
- Tian, X. et al. (2018). Eigenfunction-based multitask learning in a reproducing kernel Hilbert space. *IEEE Trans. Neural Netw. Learn. Syst.*, 30(6):1818–1830.
- Tian, Y. et al. (2013). Nonparallel support vector machines for pattern classification. *IEEE Trans. Cybern.*, 44(7):1067–1079.

- Tsai, M.-H. et al. (2014). A bio-inspired computing model for ovarian carcinoma classification and oncogene detection. *Bioinformatics*, 31(7):1102–1110.
- Vapnik, V. (1992). Principles of risk minimization for learning theory. In *Advances in Neural Information Processing Systems*, pages 831–838.
- Wang, B. et al. (2017). Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nat. Methods*, 14(4):414–416.
- Wang, Y. et al. (2018). Weighted general group lasso for gene selection in cancer classification. *IEEE Trans. Cybern.*, 49(8):2860–2873.
- Wilson, C. and Lir, K. (2019). *RMKL: Multiple kernel learning for classification or regression problems, R package version 1.0*.
- Xia, X. et al. (2019). Network analysis of aging acceleration reveals systematic properties of 11 types of cancers. *FEBS Open Bio*.
- Xiao, Y. and Meierhofer, D. (2019). Glutathione metabolism in renal cell carcinoma progression and implications for therapies. *Int. J. Mol. Sci.*, 20(15):3672.
- Xu, Q. and Yang, Q. (2011). A survey of transfer and multitask learning in bioinformatics. *J. Comput. Sci. Eng.*, 5(3):257–268.
- Xu, Z. et al. (2010). Simple and efficient multiple kernel learning by group Lasso. In *27th International Conference on Machine Learning*.
- Yan, P. et al. (2014). Label image constrained multiatlas selection. *IEEE Trans. Cybern.*, 45(6):1158–1168.
- Yang, M. et al. (2013). Thyroid gland metastasis arising from breast cancer: A case report. *Oncol. Lett.*, 5(6):1836–1838.
- Ylipää, A. et al. (2013). Characterization of aberrant pathways across human cancers. *BMC Syst. Biol.*, 7(1):S1.
- Yousefi, N. et al. (2015). Multi-task learning with group-specific feature space sharing. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 120–136. Springer.

Yu, F. et al. (2017). Association between breast cancer and thyroid cancer: a descriptive study. *Transl. Cancer Res.*, 6(2):393–401.

Yu, S. et al. (2010). ℓ_2 -norm multiple kernel learning and its application to biomedical data fusion. *BMC Bioinform.*, 11(1):309.

Zhang, H.-Y. et al. (2014). Transforming growth factor- β 1-induced epithelial-mesenchymal transition in human esophageal squamous cell carcinoma via the PTEN/PI3K signaling pathway. *Oncol. Rep.*, 32(5):2134–2142.

Zhang, X.-L. (2014). Convex discriminative multitask clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(1):28–40.

Appendix A

SUPPLEMENTARY MATERIALS FOR MULTIPLE KERNEL LEARNING ON GENE SETS

A.1 Supplementary Numerical Results For Early- and Late-Stage Discrimination (E2)

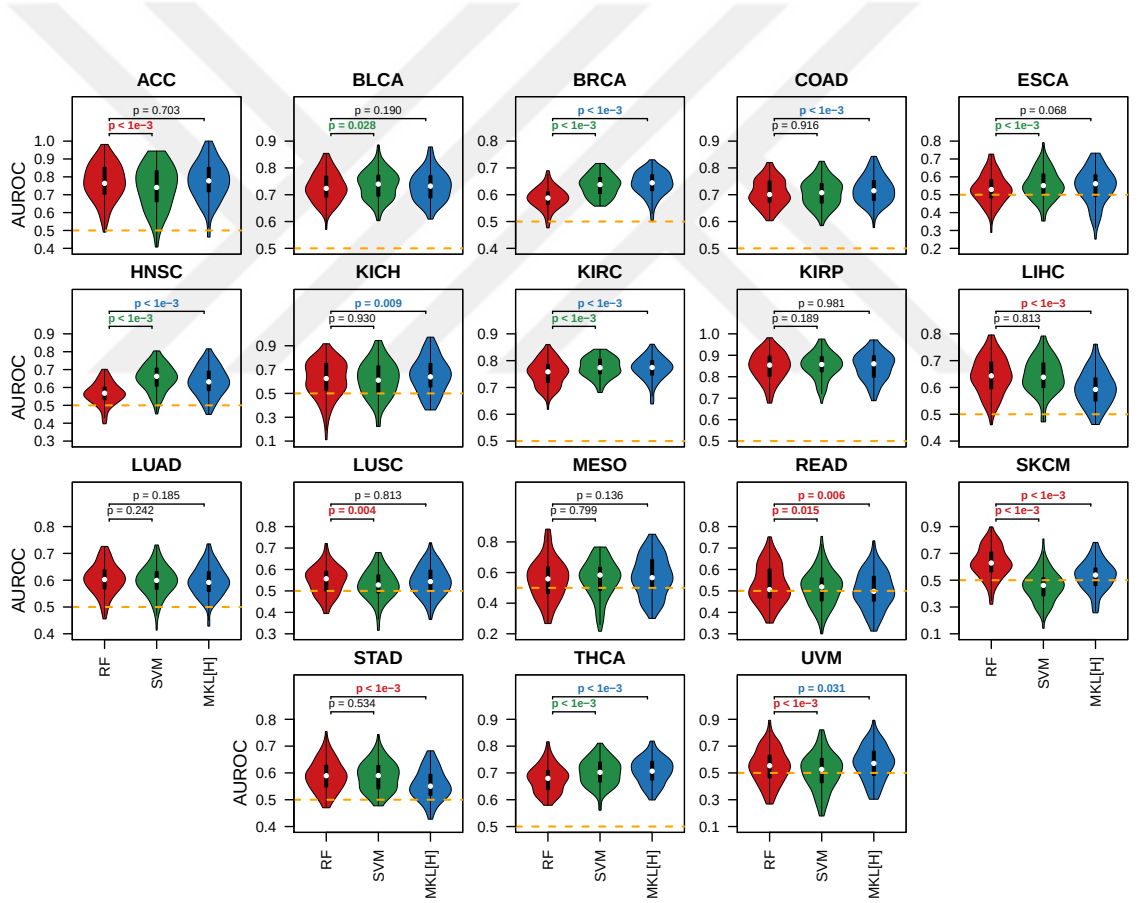


Figure A.1: Predictive performances of RF, SVM and MKL[H] on 18 cancer datasets in E2 constructed from TCGA cohorts.

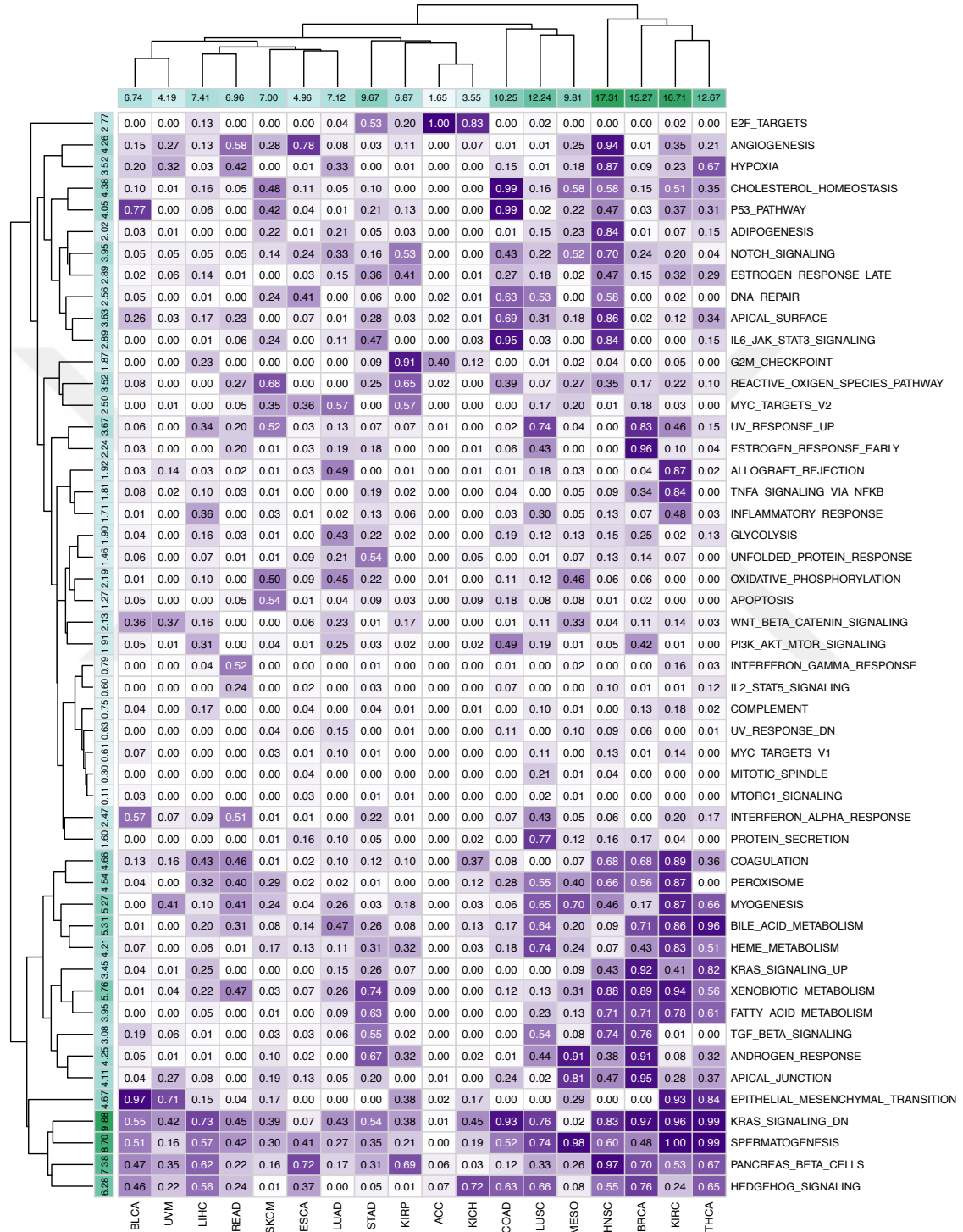


Figure A.2: Selection frequencies of 50 gene sets in the Hallmark collection identified by MKL[H] for 18 cancer datasets in E2.

Appendix B

SUPPLEMENTARY MATERIALS FOR SPARSE
MULTIPLE KERNEL LEARNING**B.1 Supplementary Numerical Results For Early- and Late-Stage Discrimination (E1)**

Table B.1: Mean AUROC values averaged over 100 replications of classification algorithms across 15 cancer datasets we used in E1.

Cohort	RF	SVM	SMKL [H]	gLMKL [H]	L2MKL [H]	SpMKL [H]	SMKL [P]	gLMKL [P]	L2MKL [P]	SpMKL [P]
BRCA	52.88%	62.76%	62.81%	63.73%	61.37%	64.85%	59.84%	57.76%	58.60%	62.84%
COAD	52.46%	66.38%	64.96%	66.31%	65.64%	68.31%	64.33%	65.74%	64.25%	74.83%
ESCA	71.09%	60.93%	65.82%	69.32%	64.47%	66.92%	58.19%	55.27%	62.36%	71.36%
HNSC	56.73%	69.20%	72.82%	70.95%	74.72%	73.86%	68.77%	66.85%	69.78%	69.22%
KICH	72.21%	70.67%	74.67%	73.22%	76.17%	82.67%	72.50%	67.89%	72.33%	74.39%
KIRC	76.51%	79.34%	78.17%	78.37%	78.94%	79.44%	76.70%	76.26%	77.06%	80.86%
KIRP	80.06%	78.54%	79.33%	78.88%	80.54%	80.11%	79.55%	79.88%	80.64%	81.75%
LIHC	68.23%	66.61%	68.32%	68.42%	67.96%	69.00%	67.12%	66.16%	68.56%	68.38%
LUAD	62.20%	62.55%	63.01%	62.99%	63.71%	64.08%	62.07%	60.92%	62.76%	64.05%
LUSC	63.20%	64.24%	62.95%	62.64%	64.61%	64.43%	60.60%	61.60%	60.88%	64.84%
PAAD	67.45%	70.03%	70.38%	73.36%	69.11%	82.09%	71.82%	69.32%	72.06%	74.08%
READ	61.57%	56.95%	65.39%	65.72%	66.88%	67.23%	61.94%	58.76%	59.89%	64.72%
STAD	74.33%	73.77%	71.20%	71.09%	73.07%	72.74%	71.52%	73.03%	71.94%	79.77%
TGCT	73.91%	75.04%	71.38%	71.07%	72.62%	72.33%	73.35%	70.51%	73.89%	74.18%
THCA	65.15%	70.49%	69.63%	70.74%	69.65%	70.70%	66.03%	64.34%	65.40%	66.09%
Mean	66.53%	68.50%	69.39%	69.79%	69.96%	71.92%	67.62%	66.28%	68.03%	71.42%

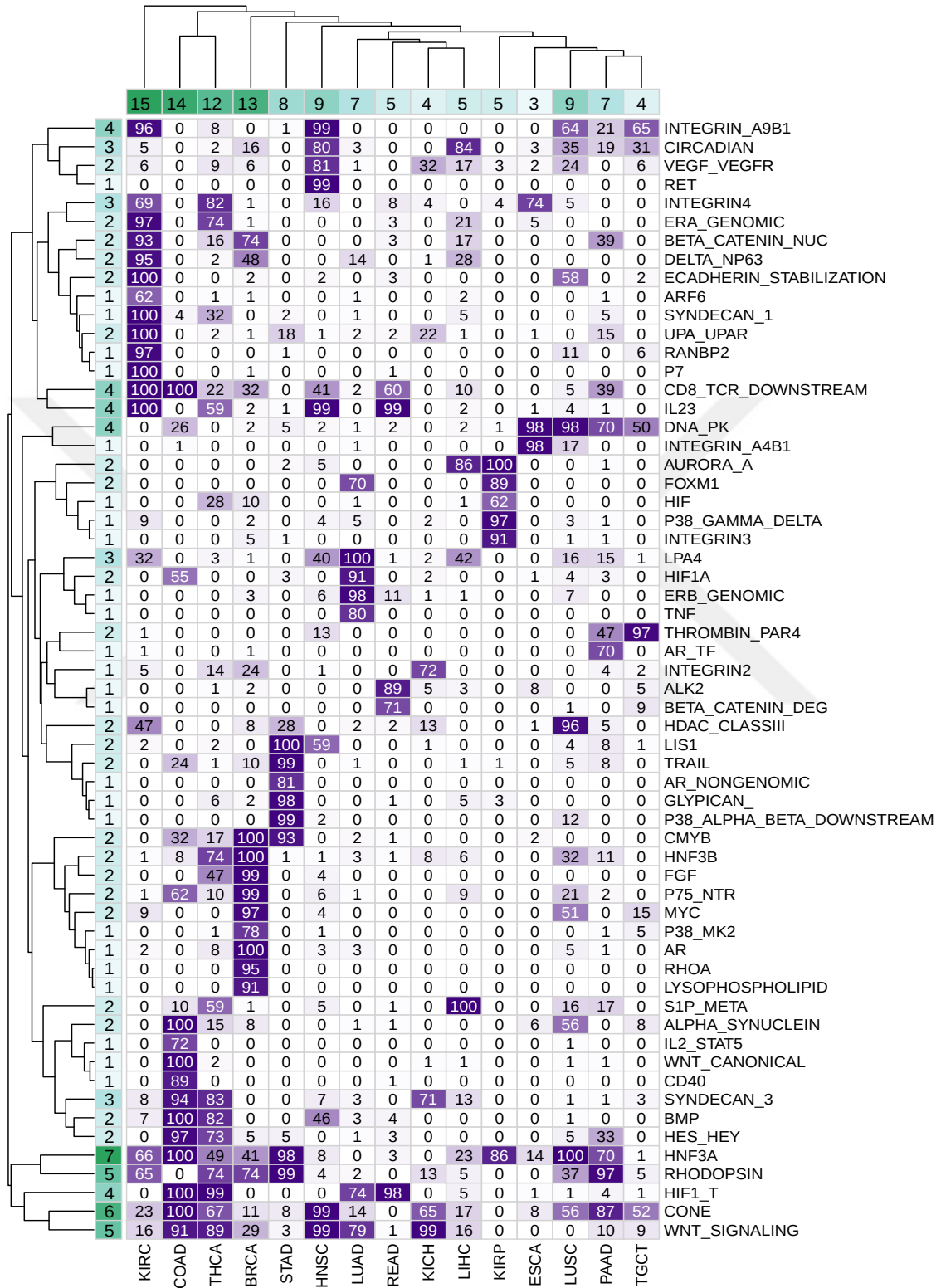


Figure B.1: Selection frequencies of 196 pathways in the PID collection as identified by SpMKL[P] for 15 cancer datasets in E1. For better visualization, only pathways with a minimum frequency of 60 are shown.

B.2 Supplementary Numerical Results For Early- and Late-Stage Discrimination (E2)

To test the robustness of our methods, in our second experiment (E2), we constructed 18 datasets from the TCGA cohorts by considering **Stages I-II** as the early-stage cancers, and **Stages III-IV** as the late-stages cancers. Details of these datasets are presented in Table 2.1.

Table B.2: Mean AUROC values averaged over 100 replications of classification algorithms across 18 cancer datasets we used in E2.

Cohort	RF	SVM	SMKL[H]	gLMKL[H]	L2MKL[H]	SpMKL[H]	SMKL[P]	gLMKL[P]	L2MKL[P]	SpMKL[P]
ACC	76.66%	72.43%	78.20%	78.87%	77.15%	79.31%	80.98%	75.67%	75.30%	79.67%
BLCA	73.49%	73.42%	72.05%	73.10%	72.46%	74.51%	72.10%	72.03%	72.48%	74.43%
BRCA	59.91%	62.62%	60.39%	60.54%	62.30%	62.45%	61.73%	61.78%	61.56%	64.72%
COAD	70.77%	70.48%	69.81%	70.02%	69.87%	72.26%	68.45%	70.14%	70.27%	71.43%
ESCA	55.33%	50.66%	56.44%	58.66%	52.40%	60.78%	53.74%	52.99%	54.47%	60.15%
HNSC	58.85%	65.00%	66.74%	65.76%	65.16%	66.95%	64.65%	65.17%	62.17%	69.47%
KICH	62.81%	61.47%	64.39%	66.97%	60.11%	72.31%	67.11%	62.00%	65.06%	74.17%
KIRC	76.29%	78.17%	76.63%	77.19%	78.60%	78.30%	75.60%	74.64%	75.61%	77.88%
KIRP	84.61%	84.06%	85.03%	84.34%	85.54%	83.66%	84.73%	83.93%	85.23%	84.58%
LIHC	64.50%	64.31%	61.78%	61.25%	63.35%	59.50%	64.71%	61.48%	65.35%	65.73%
LUAD	60.18%	58.51%	60.05%	60.13%	59.37%	60.27%	56.78%	56.03%	57.67%	61.18%
LUSC	53.93%	53.70%	54.74%	54.63%	54.64%	56.87%	53.68%	54.24%	52.29%	58.73%
MESO	54.84%	56.32%	56.97%	58.03%	53.90%	64.70%	62.03%	60.92%	60.58%	73.10%
READ	53.11%	52.67%	54.53%	51.77%	51.90%	53.05%	51.95%	51.25%	52.40%	65.38%
SKCM	63.51%	44.41%	56.03%	54.88%	58.33%	64.10%	58.95%	63.23%	59.85%	68.63%
STAD	59.01%	57.93%	56.18%	55.82%	56.72%	59.47%	56.54%	56.84%	57.88%	63.86%
THCA	68.35%	70.34%	71.13%	70.81%	69.81%	70.86%	67.22%	67.98%	68.38%	68.32%
UCEC	54.81%	49.75%	56.28%	56.02%	50.41%	63.14%	58.74%	63.93%	54.07%	70.21%
Mean	63.94%	62.57%	64.30%	64.38%	63.45%	66.80%	64.43%	64.12%	63.92%	69.54%

Table B.3: Average number of genes used by each classification model over 100 replications across 18 cancer datasets we used in E2.

Cohort	RF & SVM	SMKL [H]	gLMKL [H]	L2MKL [H]	SpMKL [H]	SMKL [P]	gLMKL [P]	L2MKL [P]	SpMKL [P]
ACC	19,814	1,486	300	7,324	206	79	138	8,099	72
BLCA	19,814	1,943	1,322	7,324	748	1,409	733	8,099	469
BRCA	19,814	2,668	3,044	7,324	1,774	1,652	1,373	8,099	693
COAD	19,814	2,279	1,271	7,324	595	1,003	618	8,099	465
ESCA	19,814	1,535	551	7,324	316	1,079	402	8,099	128
HNSC	19,814	2,475	2,396	7,324	1,624	1,505	1,024	8,099	589
KICH	19,814	1,943	571	7,324	276	922	307	8,099	92
KIRC	19,814	2,637	2,672	7,324	1,785	1,552	816	8,099	536
KIRP	19,814	2,432	1,101	7,324	679	1,034	405	8,099	282
LIHC	19,814	2,341	1,521	7,324	1,374	1,534	869	8,099	488
LUAD	19,814	2,531	2,331	7,324	1,115	1,635	1,033	8,099	254
LUSC	19,814	2,288	2,217	7,324	1,117	1,087	990	8,099	404
MESO	19,814	2,144	1,306	7,324	351	1,122	435	8,099	171
READ	19,814	2,020	1,210	7,324	708	1,151	473	8,099	192
SKCM	19,814	2,435	1,198	7,324	297	727	297	8,099	164
STAD	19,814	2,405	1,413	7,324	429	1,280	981	8,099	632
THCA	19,814	2,599	1,742	7,324	1,178	1,292	695	8,099	419
UVM	19,814	1,866	461	7,324	279	626	197	8,099	140
Mean	19,814	2,224	1,479	7,324	825	1,149	655	8,099	344

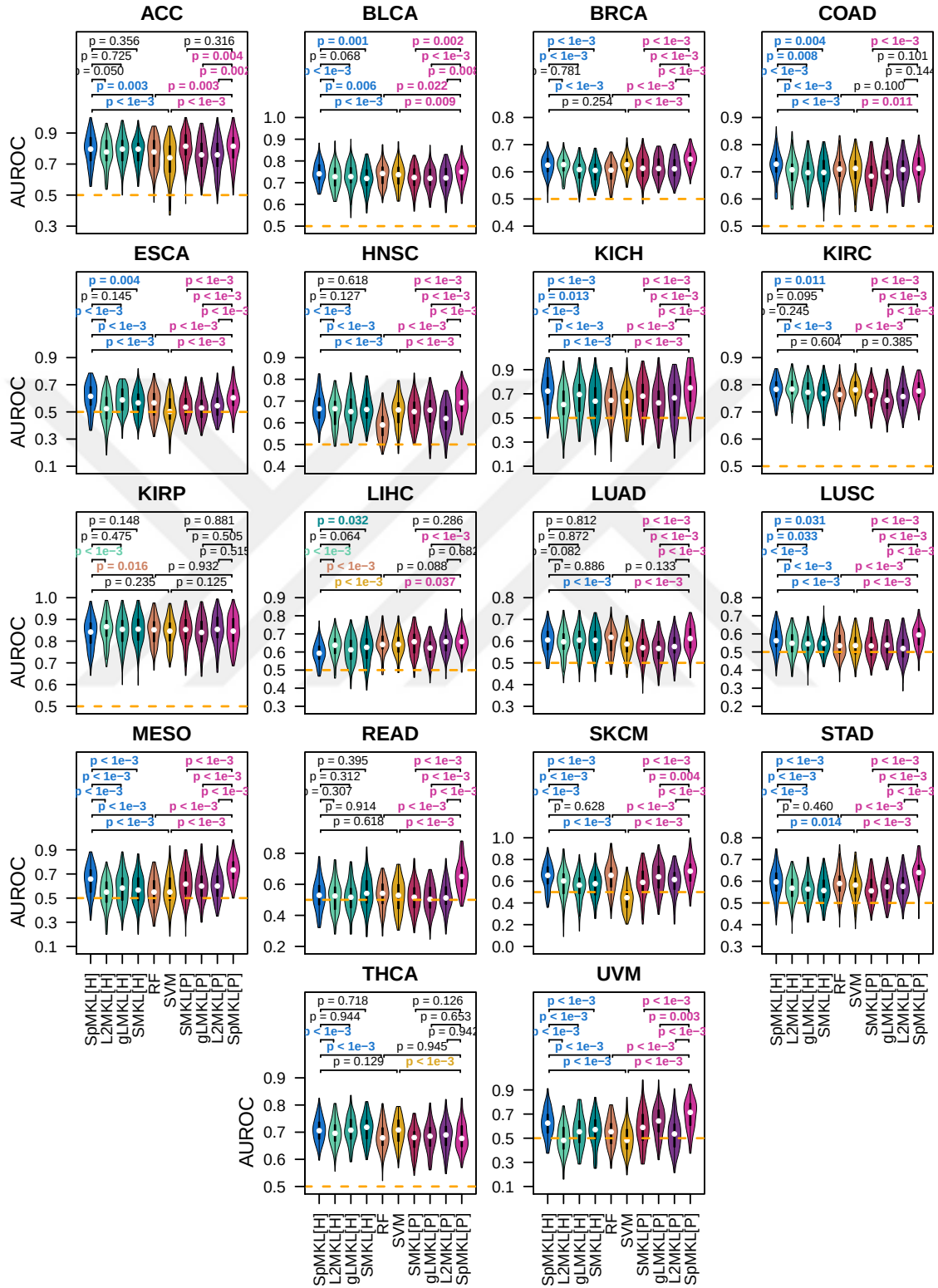


Figure B.2: Predictive performances of SpMKL[H], SpMKL[P], and other classification algorithms on 18 cancer datasets constructed for E2.

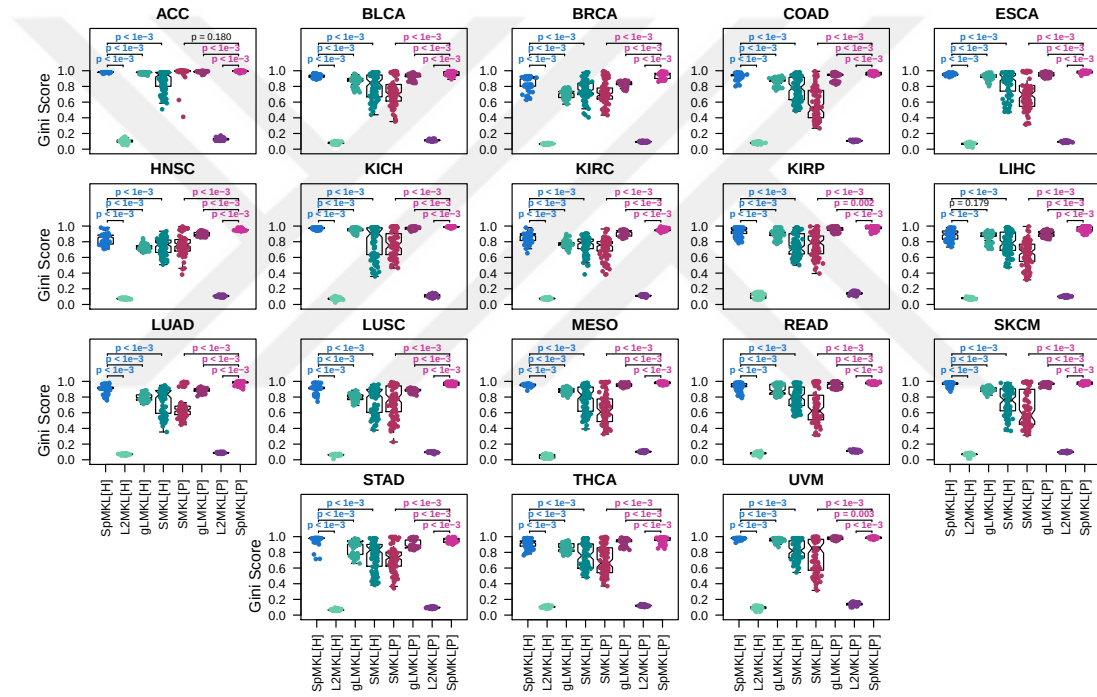


Figure B.3: Sparsity comparison of $\text{SpMKL}[\text{H}]$, $\text{SpMKL}[\text{P}]$, and other MKL algorithms on 18 cancer datasets constructed for E2.

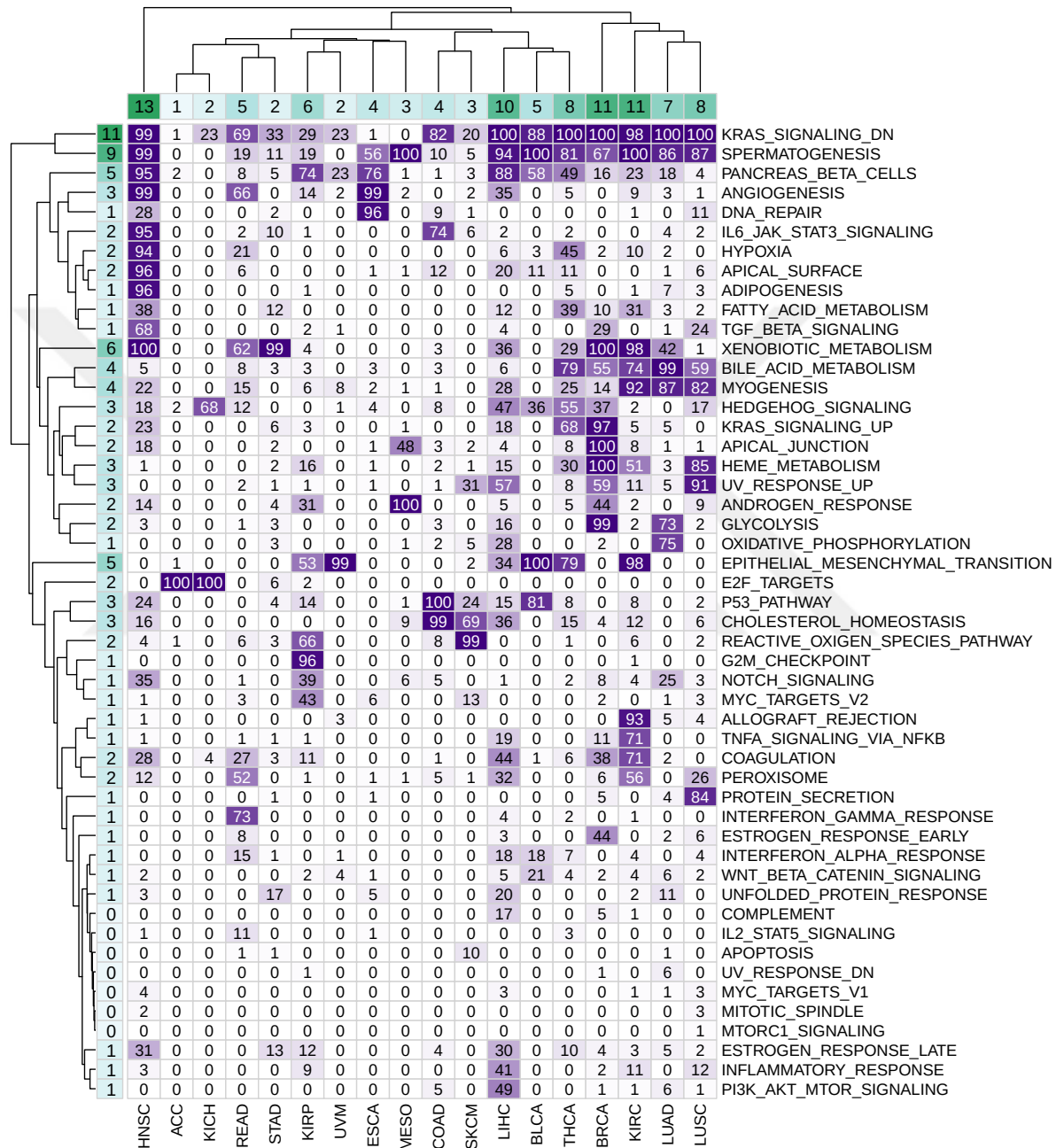


Figure B.4: Selection frequencies of 50 gene sets in the Hallmark collection as identified by SpMKL[H] for 18 cancer datasets in E2.

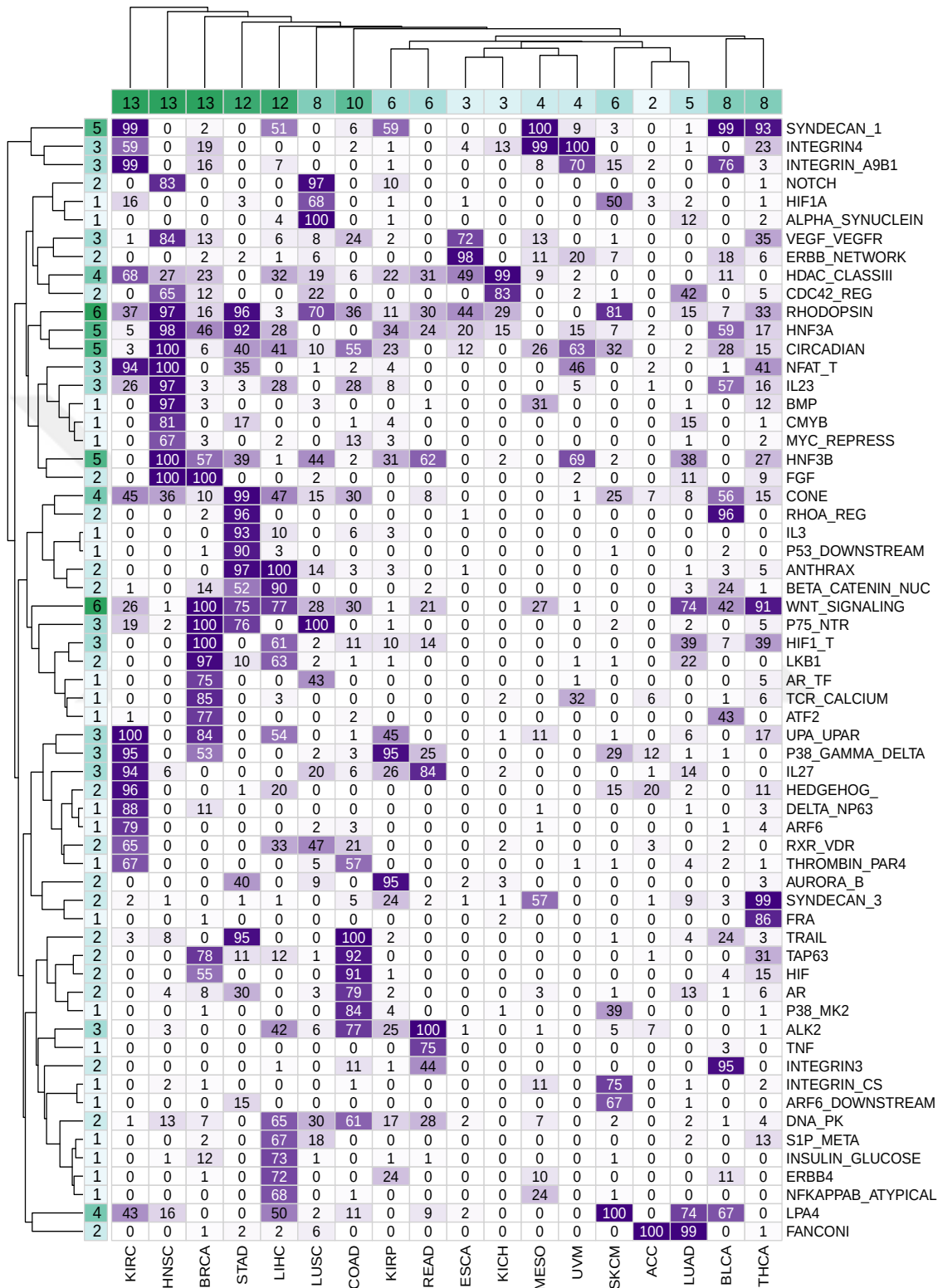


Figure B.5: Selection frequencies of 196 pathways in the PID collection as identified by SpMKL[P] for 18 cancer datasets in E2.

B.3 Supplementary Numerical Results for Two-Year Survival Analysis (S1)

Table B.4: Mean AUROC values averaged over 100 replications of classification algorithms across 20 cancer datasets we used in S1.

Cohort	RF	SVM	SMKL [H]	gLMKL [H]	L2MKL [H]	SpMKL [H]	SMKL [P]	gLMKL [P]	L2MKL [P]	SpMKL [P]
BLCA	64.45%	59.76%	59.21%	58.68%	59.08%	63.38%	58.44%	57.40%	58.99%	61.87%
BRCA	64.56%	68.59%	65.05%	64.92%	67.00%	67.31%	64.77%	64.25%	65.37%	70.80%
CESC	70.38%	74.44%	73.10%	74.14%	74.98%	77.51%	74.91%	73.46%	75.05%	77.72%
COAD	60.89%	63.20%	61.40%	62.08%	64.02%	64.17%	61.52%	61.74%	62.49%	64.95%
ESCA	60.24%	68.65%	74.65%	74.33%	75.89%	82.71%	75.44%	72.97%	74.70%	85.27%
GBM	60.12%	51.92%	56.65%	57.01%	55.06%	64.29%	56.17%	56.03%	56.06%	71.12%
HNSC	61.91%	59.44%	61.79%	62.62%	63.00%	62.11%	63.74%	64.18%	63.99%	65.66%
KIRC	68.05%	67.11%	68.37%	68.40%	70.44%	69.11%	66.78%	65.77%	67.28%	72.20%
KIRP	88.86%	88.35%	86.71%	85.90%	87.49%	87.03%	87.59%	86.66%	87.42%	87.34%
LAML	72.53%	68.79%	69.84%	68.50%	70.01%	74.21%	68.30%	70.37%	70.12%	75.30%
LGG	87.51%	89.17%	89.85%	90.42%	89.13%	91.11%	87.62%	87.60%	88.63%	88.66%
LIHC	70.57%	68.87%	70.64%	69.96%	70.97%	71.25%	69.52%	69.50%	70.18%	69.18%
LUAD	65.14%	65.20%	67.03%	67.45%	65.70%	69.54%	67.08%	67.46%	67.58%	70.54%
LUSC	53.12%	52.04%	52.07%	52.26%	51.41%	58.90%	54.17%	53.90%	54.16%	63.70%
MESO	80.27%	75.73%	79.72%	79.35%	80.02%	80.78%	80.10%	80.17%	80.25%	80.75%
OV	65.18%	66.34%	66.38%	65.34%	66.31%	66.80%	68.99%	67.19%	69.63%	72.21%
PAAD	60.43%	59.30%	56.71%	56.77%	56.54%	62.78%	53.57%	52.77%	54.47%	57.14%
SARC	70.73%	66.89%	70.87%	71.31%	70.67%	75.62%	70.14%	70.97%	70.57%	74.13%
STAD	59.07%	58.14%	56.55%	56.73%	55.71%	60.64%	57.66%	58.55%	58.07%	65.70%
UCEC	64.88%	66.75%	62.83%	60.66%	63.89%	62.97%	61.65%	61.44%	63.78%	68.50%
Mean	67.44%	66.94%	67.47%	67.34%	67.87%	70.61%	67.41%	67.12%	67.94%	72.14%

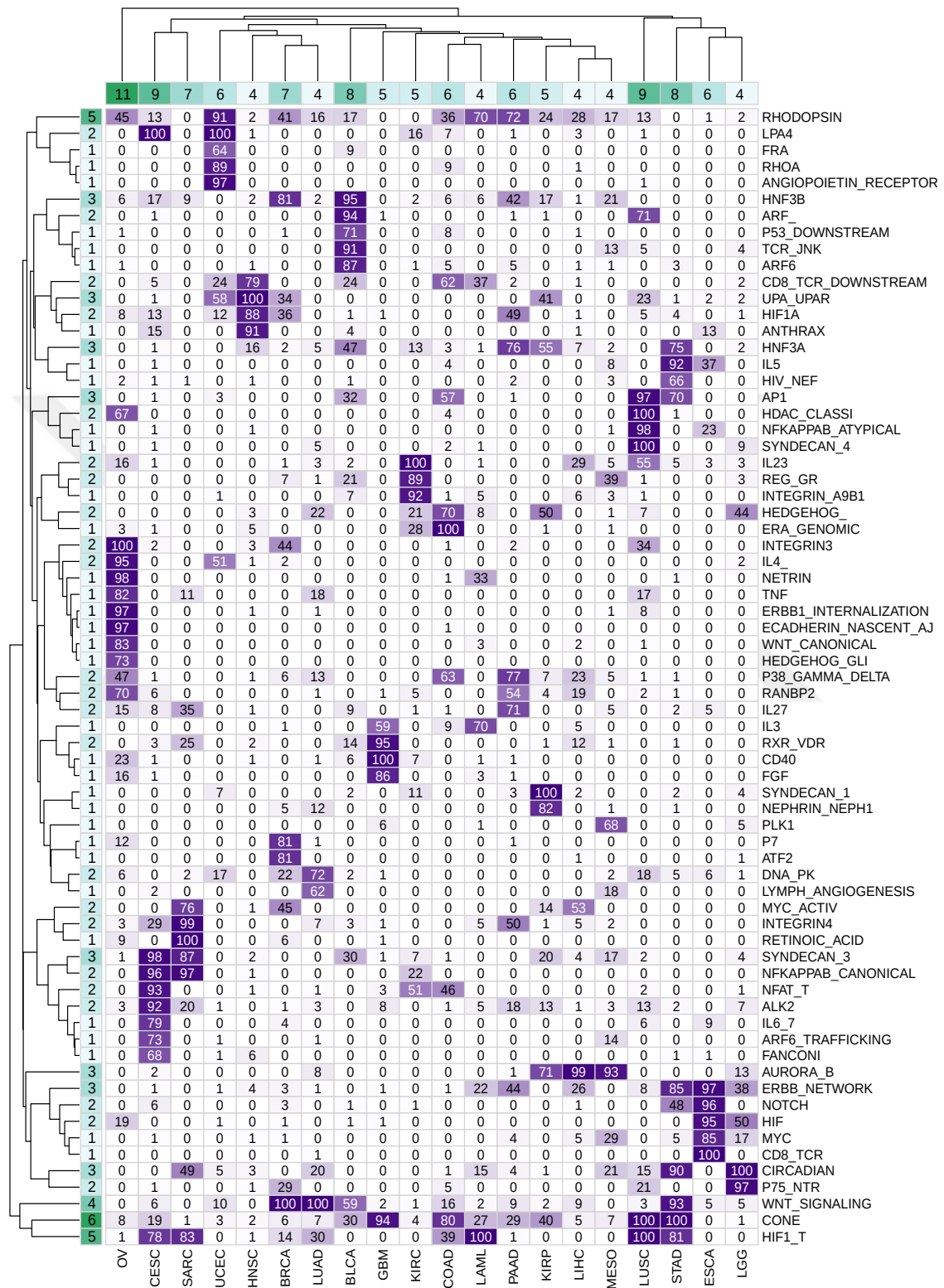


Figure B.6: Selection frequencies of 196 pathways in the PID collection as identified by SpMKL[P] for 20 cancer datasets in S1.

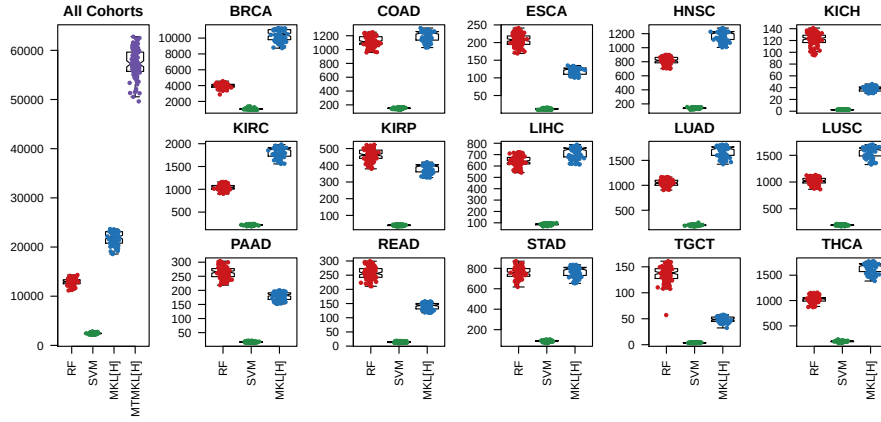
Appendix C

SUPPLEMENTARY MATERIALS FOR MULTITASK MULTIPLE KERNEL LEARNING

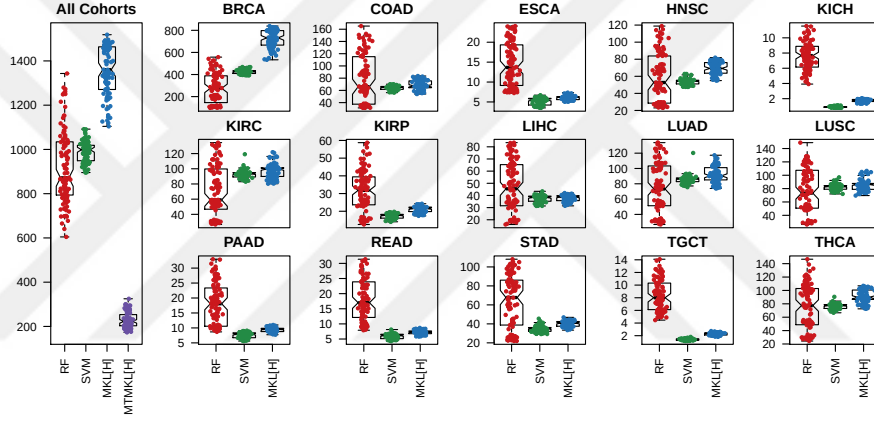
C.1 Computational Complexity Analysis

Figure C.1 presents the cohort-based time comparison of the three single-task models (i.e., RF, SVM and MKL[H]) as well as the comparison of the overall time of all four models over all cohorts. Despite the high computation time of MTMKL[H] in the cross-validation phase, we observe that computation time of this model is significantly lower than that of all three single-task models in the test phase. The high cross-validation time of MTMKL[H] is due to the large number of hyper-parameter evaluations in the cross-validation phase. In particular, MTMKL[H] requires evaluating $|C| \times |\lambda_1| \times |\lambda_2|$ choices of the hyper parameters, whereas SVM and MKL[H] require evaluating $|C|$ parameters and RF requires evaluating $|\text{ntree}|$ parameters. The lower test time, on the other hand, underscores the efficiency of the proposed cutting-plane algorithm for solving the MTMKL problem, as opposed to the iterative method proposed by Xu et al. (2010) for solving the MKL problem. This is because, in the cutting-plane algorithm, both upper and lower bounds on the objective function are available and as soon as these bounds are close enough, we can verify optimality of the solution. On the other hand, the only piece of information available during the iterative method for solving the MKL problem is the upper bound on the objective function, which may not be sufficient for asserting convergence.

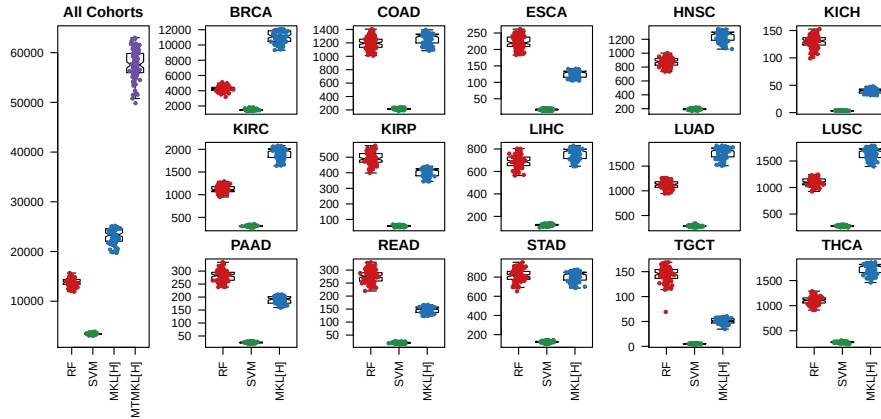
We would also like to remark that the presented times for MTMKL[H] in Figure C.1 correspond to the second stage MTMKL problem and do not consist of the first stage (i.e., the co-clustering stage) MTMKL problem. The sole purpose of performing the first stage is to obtain the similarity matrices $\bar{\phi}$ and $\bar{\theta}$. This stage involves performing 100 independent replications each with a four-fold cross-validation phase for selecting hyper parameters $(C, \lambda_1, \lambda_2)$ and a final test phase for obtaining the co-clustering matrices ϕ and θ . For each replication, these two phases together take around 208 hours on average. Nevertheless, because of parallelizable nature of the cross-validation phase, with sufficient number of processors, the first stage can be performed in two hours.



(a) Cross-validation time (in seconds)



(b) Test time (in seconds)



(c) Total time (in seconds)

Figure C.1: Cross-validation, test and total time of RF, SVM, MKL [H], and MTMKL [H] for 15 cancer datasets constructed from TCGA in E1 over 100 replications.

C.2 Sensitivity to Hyper-parameters

We next provide some numerical results regarding the effect of the model parameters on accuracy of the model. As illustrated in Figure C.2, the accuracy of the prediction models is mildly sensitive to the hyper-parameters. Hence, to build robust models, the hyper-parameters of the models (C, λ_1, λ_2) are set using a four-fold cross-validation scheme.

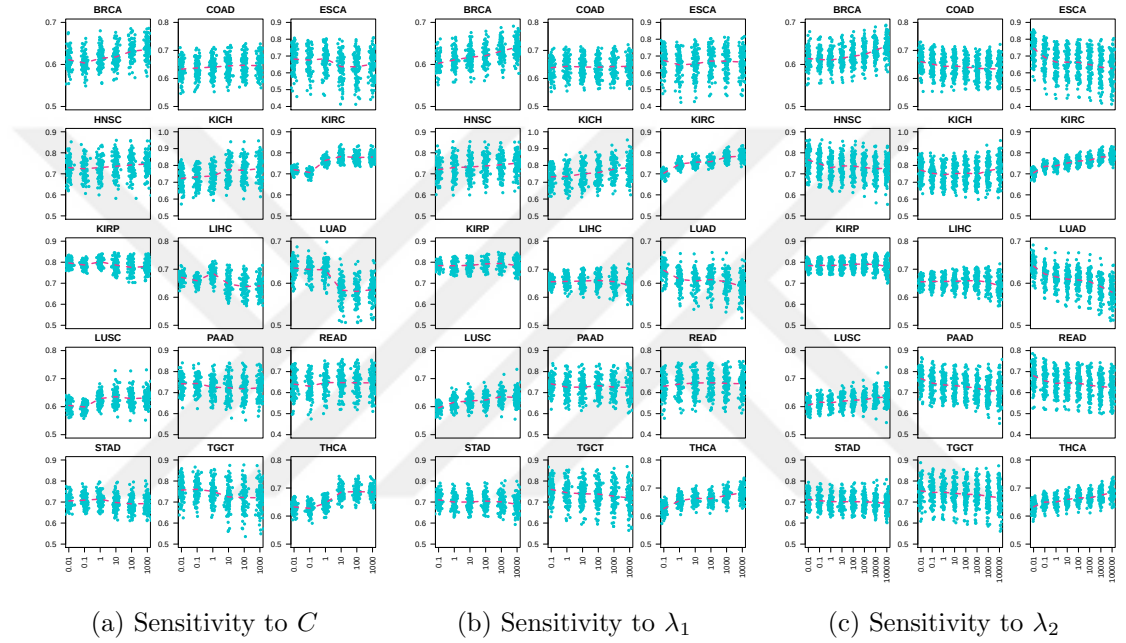


Figure C.2: Sensitivity of average cross-validation accuracy to different values of hyper parameters C , λ_1 and λ_2 for 15 cancer datasets in E1.

Additionally, the magnitude of each parameter influences the behavior of the model as follows:

- The parameter C controls sensitivity of the classifier to misclassification, thus for small values of C the solution is less affected by the underlying data of the problem (i.e., weight of kernel functions). As C grows, the soft-margin SVM converges to the hard-margin SVM.
- The parameter λ_1 is the weight of the linear penalty term in the objective function (i.e., $\lambda_1 \sum_{s=1}^T \sum_{m=1}^P (1 - \theta_{s,m}) \eta_{s,m}$). As this parameter grows, the solution is forced to follow the given pattern $\boldsymbol{\theta}$ (or $\bar{\boldsymbol{\theta}}$ in the MTMKL problem) more strictly. Note that in the co-clustering step, $\theta_{s,m} = 1$ when cohort s and kernel

m are in the same cluster and $\theta_{s,m} = 0$ otherwise. Hence, a large value of λ_1 enforces $\eta_{s,m}$ to be zero when $\theta_{s,m} = 0$. Consequently, as λ_1 increases, the “soft” clustering of cohorts and gene sets becomes a “hard” clustering. Similarly, in the MTMKL problem, a large value of λ_1 enforces $\eta_{s,m}$ to take a small value when $\bar{\theta}_{s,m}$ is small and vice versa.

- The parameter λ_2 is the weight of the quadratic penalty term in the objective function (i.e., $\sum_{s=1}^T \sum_{q=1}^T \bar{\phi}_{s,q} \|\boldsymbol{\eta}_s - \boldsymbol{\eta}_q\|^2$). The effect of the magnitude of this parameter is similar to that of λ_1 . As λ_2 grows, the solution is forced to follow the given pattern $\boldsymbol{\phi}$ (or $\bar{\boldsymbol{\phi}}$ in the MTMKL problem) more strictly. A large value of λ_2 enforces $\|\boldsymbol{\eta}_s - \boldsymbol{\eta}_q\|$ to be zero when $\phi_{s,q} = 1$. This implies that as λ_2 increases, the “soft” clustering of cohorts s and q becomes a “hard” clustering, which means when $\phi_{s,q} = 1$ cohorts s and q must attain the same kernel profiles. Similarly, in the MTMKL problem, a large value of λ_2 enforces cohorts s and q to attain more similar kernel profiles when $\bar{\phi}_{s,q}$ is large and vice versa.

C.3 Supplementary Numerical Results For Early- and Late-Stage Discrimination (E2)

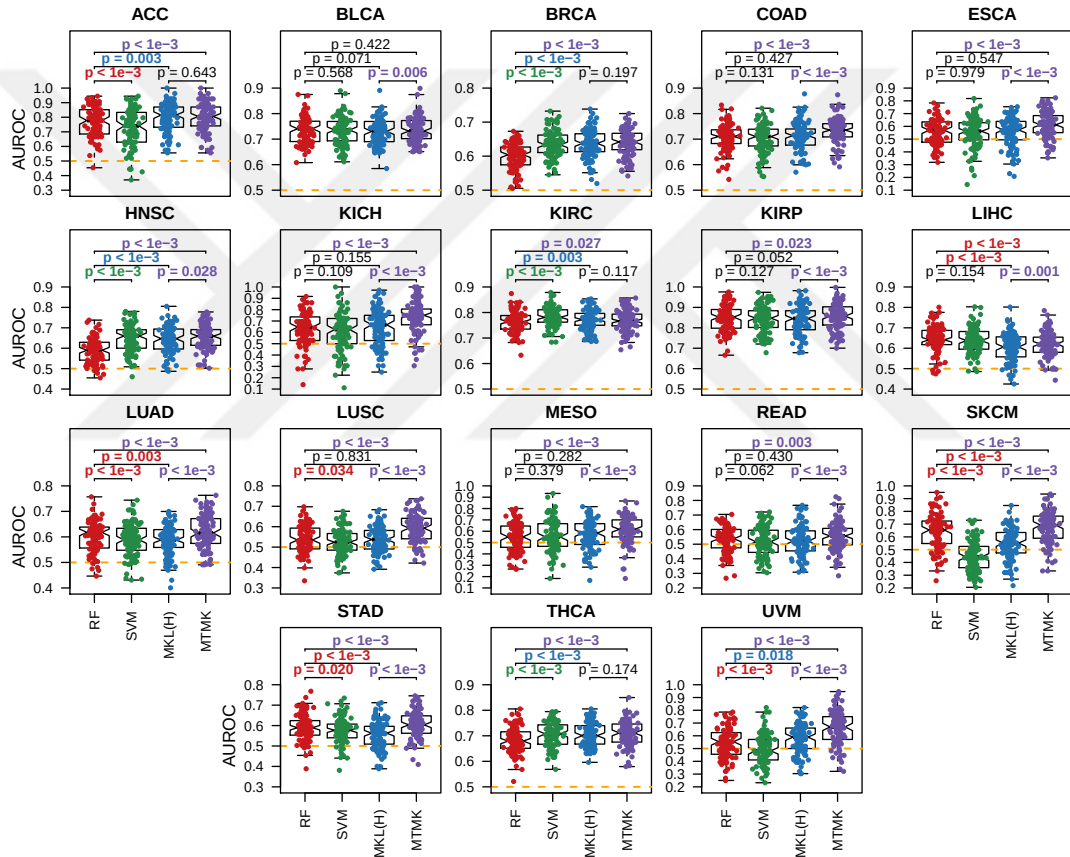


Figure C.3: Predictive performances of RF, SVM, MKL [H], and MTMKL [H] on 18 cancer datasets constructed from TCGA in E2.

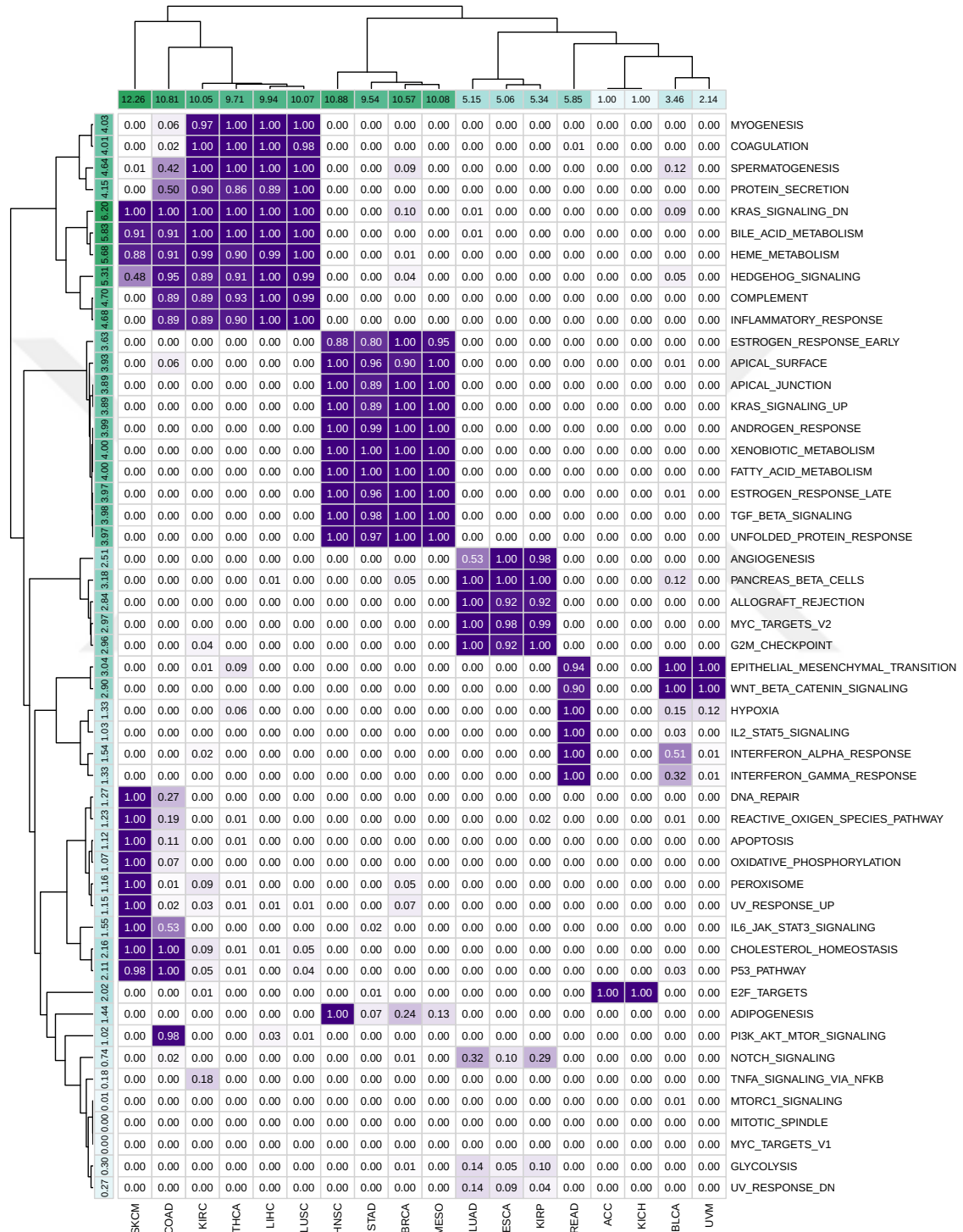


Figure C.4: Selection frequencies of 50 gene sets in the Hallmark collection as identified by MTMKL[H] for 18 cancer datasets in E2.