Optimization Models for Survival Analysis to Identify Key Gene Sets in Cancer

by

Onur Dereli

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

Optimization Models for Survival Analysis to Identify Key Gene Sets in Cancer

Koç University

Graduate School of Sciences and Engineering

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

Onur Dereli

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:

_____

Prof. Ceyda Oğuz (Advisor)

_____

Assoc. Prof. Mehmet Gönen

_____

Assoc. Prof. Müjdat Zeybel

_____

Assoc. Prof. Nihal Ata Tutkun

_____

Assist. Prof. Öznur Taştan

Date: _____

To my beloved family...

ABSTRACT

Optimization Models for Survival Analysis to Identify Key Gene Sets in Cancer

Onur Dereli

Doctor of Philosophy in Industrial Engineering and Operations Management

August 5, 2020

Using genomic characterizations of tumours biopsied from cancer patients has a great importance in understanding the formation and progression mechanisms in cancer. Survival analysis is one of the research methods that is used to predict overall survival time of cancer patients and to understand the aforementioned progression mechanisms. High dimensional structure of the genomic characterizations with the limited number of training samples makes survival analysis a challenging task. To be able to identify the survival associated biological mechanisms, cancer-specific pathway/gene set collections can be integrated into machine learning models. Existing approaches usually follow a two-stage approach that either identify predictive genes using a feature selection method and map these selected genes to known pathways/gene sets, or train separate models for each pathway/gene set and try to pick informative ones considering each model's predictive performance. Following such a two-stage approach might result in inefficacy of mapping selected genes to a known biological pathway/gene set due to highly correlated structure between feature groups or including related or very similar pathways/gene sets into the final model due to analyzing each pathway/gene set separately.

In this thesis, rather than following such two-stage approaches, we propose machine learning models that can conjointly identify disease related biological mechanisms and perform survival prediction using only these identified biological mechanisms. Our algorithms obtain a sparse set of pathways/gene sets for the survival associated biological mechanisms by eliminating the uninformative ones from the model. We test our algorithms using 20 cancer datasets obtained from The Cancer Genome Atlas and two cancer-specific pathway/gene set collections as input data.

We first propose a survival analysis model that integrates pathway/gene set collection into the model using multiple kernel learning. Our algorithm with conjoint modelling approach obtains statistically significantly better or comparable predictive performances against survival random forest (RF) and survival support vector machine (SVM) using significantly fewer gene expression features.

Predictive performances of machine learning algorithms can be increased using multitask learning. For this purpose, we extend our multiple kernel learning-based algorithm towards multitask learning. Our multitask learning algorithm both models multiple cancer datasets simultaneously and integrates cancer related biological mechanisms into the machine learning model. The algorithm is able to identify common underlying biological mechanisms for cancer by obtaining better or comparable predictive results against survival RF, survival SVM, and our multiple kernel learning survival analysis algorithm.

We also extend our multitask learning algorithm towards task clustering to identify the groups of cancer types that share similar underlying biological mechanisms. To this aim, we propose a unified formulation for task clustering, survival analysis, and knowledge extraction. Our clustering algorithm identifies relevant cancer groups by obtaining statistically significantly better or comparable predictive performances against survival RF, survival SVM, our multiple kernel learning and multitask multiple kernel learning survival analysis algorithms. Numbers of gene expression features and gene sets used by our clustering algorithm are significantly fewer than those of benchmark algorithms.

These results show that our methods that identify survival associated biological mechanisms, obtain better or comparable predictive performances against survival analysis methods developed on genomic data without using the pathway/gene set information in the literature. In addition, we prove that survival prediction can be performed using fewer number of gene expression features compared to these benchmark algorithms. We also identify the cancer groups that share similar biological mechanisms without decreasing the predictive power.

ÖZETÇE

Kanser Hastalığında Önemli Gen Kümelerini Belirlemek İçin Geliştirilen En İyileme Modelleri

Onur Dereli

Endüstri Mühendisliği ve İşletme Yönetimi, Doktora

5 Ağustos 2020

Tümör biyopsilerinden elde edilen genomik karakterizasyonlar, kanserin oluşumu ve seyri hakkında bilgi edinmemize yardımcı olmaktadır. Sağkalım analizi, kanser hastalarının sağkalım sürelerini tahmin etmek ve hastalığın ilerleme mekanizmalarını anlamak için kullanılan araştırma yöntemlerinden birisidir. Eğitim örneklerinin sınırlı sayıda ve genomik verilerdeki öznitelikler arasındaki korelasyonun oldukça yüksek olması, sağkalım analizini zorlu bir hale getirmektedir. Genomik karakterizasyonların yanı sıra, kansere özgü biyolojik yolak bilgilerinin de sağkalım modellerinde girdi olarak kullanılması, sağkalım ile ilişkili biyolojik yolakların belirlenmesine olanak sağlar. Literatürde sunulan yöntemler, ya biyolojik yolak bilgilerini kullanmadan genomik veriler üzerinde tahmin modelleri geliştirip sağkalım ile ilişkili genleri belirler ve ardından yolak bilgilerini kullanarak seçilen gen bilgisini yorumlar, ya da her bir yolak bilgisi için ayrı tahmin modelleri geliştirip sonrasında bilgilendirici olanları seçmeye çalışır. Ancak, bu tür iki aşamalı bir yaklaşımın izlenmesi, genomik karakterizasyonlar arasındaki yüksek korelasyon nedeniyle, seçilen genlerin bilinen biyolojik yolaklar ile başarılı bir şekilde eşleştirilmesine engel olabilmektedir. Ayrıca, her bir yolak için ayrı tahmin modellerinin geliştirilmesi, birbirleriyle oldukça benzer ya da ilintili yolakların seçilmesine yol açabilmektedir.

Bu tezde, yukarıda bahsedildiği gibi iki aşamalı bir yöntem izlemek yerine, sağkalım analizi ve kanser ile ilgili biyolojik yolakların belirlenmesini aynı anda gerçekleştiren, ve sağkalım analizini yaparken yalnızca belirlenen biyolojik yolakları kullanan yeni yapay öğrenme yöntemleri önermekteyiz. Geliştirdiğimiz algoritmalar, sağkalım ile ilişkili olmayan biyolojik mekanizmaları modelden çıkararak, kanser hastalarının sağkalım süresi tahmini sırasında bilgilendirici olan biyolojik yolaklar için seyrek bir çözüm kümesi elde etmektedir. Algoritmalarımızı, Kanser Genom Atlası projesi kap-

samında oluşturulan 20 farklı kanser verisi ve kansere özgü biyolojik yolak bilgilerini içeren iki farklı veri tabanını kullanarak test etmekteyiz. İlk olarak, çoklu çekirdek öğrenimi yardımı ile biyolojik yolak bilgilerini modele ekleyen bir sağkalım analizi yöntemi önermekteyiz. Yolakların belirlenmesini ve sağkalım analizini birleşik olarak gerçekleştiren algoritmamız, sağkalım analizi için geliştirilen Rassal Orman (RO) ve Destek Vektör Makinesi(DVM) algoritmalarına kıyasla daha başarılı ya da benzer tahmin performanslarını, çok daha az sayıda öznitelik kullanarak elde etmiştir.

Çoklu görev öğrenimi yöntemlerinin yapay öğrenme algoritmalarının tahmin performansını arttırdığı bilinmektedir. Bu tezde, hem tahmin performansını arttırmak, hem de farklı kanser türlerini aynı anda modelleyerek, altta yatan ortak ya da benzer biyolojik sebepleri belirlemek adına, geliştirmiş olduğumuz çoklu çekirdek öğrenme tabanlı sağkalım analizi yöntemini çoklu görev öğrenimi ile birleştirmekteyiz. Çoklu görev öğrenme tabanlı algoritmamız; RO, DVM ve geliştirdiğimiz çoklu çekirdek öğrenme tabanlı algoritmamıza kıyasla, daha başarılı ya da benzer tahmin performansları elde ederek, farklı kanser türleri için altta yatan benzer biyolojik mekanizmaları belirlemektedir.

Bu tezde ek olarak, altta yatan benzer mekanizmalara sahip kanser öbeklerini belirlemek adına, geliştirdiğimiz çoklu görev çoklu çekirdek öğrenme tabanlı sağkalım analizi algoritmamızı öbekleme yöntemiyle birleştirmekteyiz. Bu amaç doğrultusunda, farklı kanser türlerinin kümelenmesi, sağkalım analizi ve bilgi çıkarımı adımları için birleşik bir matematiksel model önermekteyiz. Kümeleme tabanlı algoritmamız; RO, DVM, çoklu çekirdek öğrenme ve çoklu görev çoklu çekirdek öğrenme tabanlı sağkalım analizi algoritmalarımıza kıyasla daha başarılı ya da benzer tahmin performansları elde ederek, birbirleriyle benzer altta yatan sebeplere sahip kanser türlerini içeren öbekleri belirlemektedir. Kümele algoritmamız tarafından kullanılan öznitelik sayısı; RO, DVM ve çoklu görev çoklu çekirdek öğrenme algoritmamıza kıyasla çok daha az sayıdadır.

Elde ettiğimiz bu sonuçlarla, bu tezde sunmuş olduğumuz hayatta kalma ile ilişkili biyolojik mekanizmaları belirleyen yöntemlerimizin, literatürdeki yolak bilgilerini kullanmadan genomik veriler üzerinde geliştirilen sağkalım yöntemlerinden daha başarılı tahmin performansları elde ettiğini göstermekteyiz. Ayrıca, literatürdeki yöntemlere kıyasla, daha az sayıda öznitelik kullanarak tahmin işlemlerinin gerçekleştirilebileceğini ispatlamaktayız. Ek olarak, tahmin performansını düşürmeden, benzer altta yatan sebeplere sahip olan farklı kanser türlerini de öbekleyebilmekteyiz.

ACKNOWLEDGMENTS

First, I would like to express my deepest gratitude to my advisor Prof. Dr. Ceyda Oğuz for her endless support and encouragement throughout my education. Her guidance will always assist me in my future academic life. I am also very grateful to have the possibility to work with Assoc. Prof. Mehmet Gönen, I would not have been able to complete this research without him. His guidance, knowledge, and motivation will always be a pathfinder for me. I would also like to thank my thesis committee members, Assoc. Prof. Müjdat Zeybel, Assoc. Prof. Nihal Ata Tutkun, and Assist. Prof. Öznur Taştan, for their valuable time and useful feedbacks.

I would like to thank all the people that I have worked with at the Graduate School of Sciences and Engineering, especially to the former and current members of the Machine Intelligence and Data Analysis in Science Laboratory (MidasLab). I would particularly like to thank Çiğdem Ak, Veli Oğuzalp Bakır, and Zeynep Sümer for their valuable support and friendship. I would also like to thank my lifelong friends Alihan, Aslıgül, Burcu, Cengiz, Egehan, Elif, Osman, Ekinsu, Onur, Özgür, and Sezgin, for their support and encouragement.

I would like to thank Banu Ulusoy for making my life more meaningful. It is not possible to express how grateful I am for her unconditional love and support. Thank you for always being there for me. Lastly, I would like to express my sincere gratitude to my mother Filiz, my father Etem, and my elder brother Uğur, for their endless love and support. They always encouraged me to be the person who I am. I dedicate this thesis to them.

This thesis has been supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under Grant EEEAG 117E181 and the Ph.D. scholarship (2211) from TÜBİTAK.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

$\top$	Transpose
$1(\cdot)$	1 if the term inside the parenthesis is true, 0 otherwise
b	Bias term
C	Regularization parameter
D	Dimensionality of the feature space
$k(\cdot, \cdot)$	Kernel function
$\mathbf{K}$	Kernel matrix
K	Number of classes
$\mathcal{L}$	Lagrangian function
N	Number of training samples
P	Number of pathways/gene sets
$\mathbb{R}$	Real numbers
$\mathbb{R}_+$	Non-negative real numbers
T	Number of tasks
$\boldsymbol{w}$	Weight coefficients
$\boldsymbol{x}$	Data vector
y	Observed output value
z	Cluster assignment variable
$\mathbf{Z}$	Cluster assignment matrix
α	Support vector coefficient
β	Lagrange variable
δ_i	1 if sample i is censored, 0 otherwise
ϵ	Tube width parameter
$\boldsymbol{\eta}$	Kernel weights
ξ	Slack variable

`MKL`	Multiple Kernel Learning
`MTL`	Multitask Learning
`MTMKL`	Multitask Multiple Kernel Learning
`RF`	Random Forest
`SVM`	Support Vector Machine
`SVR`	Support Vector Regression
TCGA	The Cancer Genome Atlas

Chapter 1

INTRODUCTION

Advances in understanding genomic information have provided researchers many opportunities, such as gaining insight for the diseases and developing improved diagnostic and therapeutic methods. Genomic characterization is a method to learn about the genes and their interactions with each other and with the environment. It has been used in many machine learning studies about understanding the formation and progression of the diseases better. Predicting clinical outcomes of patients by using their genomic characterizations is one of the major approaches in answering research questions about the disease progression.

Survival analysis is one of these approaches performed to gain insight about the diseases and can be defined as the set of statistical techniques to analyze the time until the occurrence of an event. It has been widely used in many areas like finance, engineering, and bioinformatics. Predicting how long a cancer patient will survive after the diagnosis (i.e. overall survival time) or will stay disease-free after the treatment (i.e. disease-free survival time) are some of the research questions that are tried to be answered with survival analysis. In such studies, patients are observed over a specified time period to collect the survival data. Some patients do not experience the event (i.e. death) during the observation period. Such observations are accepted as incomplete and called as censored data. Censoring can occur as follows: i) patient survives until the end of observation period, ii) patient leaves the study, or iii) patient experiences another event affecting the follow-up. This phenomenon is named as right censoring, meaning that time to last follow-up for the right censored observation is only known, and this information is a lower bound for the real survival time of the right censored observation. Censoring can also occur

if we know when the event is experienced but we do not know when the condition is started. In this case, the observation is left censored. Another censoring type is the interval censoring, which we do not know the time when the condition is started and the event is experienced [Clark et al., 2003]. In this study, we will consider right censored data only. Table 1.1 shows a sample data for observed survival time of cancer patients. Figure 1.1 illustrates the time to death and time to last follow-up of the same sample data.

Table 1.1: Sample data showing the vital status and the observed survival time of cancer patients (i.e. Time to death for `Dead` patients, Time to last follow-up for `Alive` patients. Vital status of a patient represents whether the data is right censored (`Alive`) or not (`Dead`).

Vital status	**Time to death**	**Time to last follow-up**
Dead	520	NA
Alive	NA	1067
⋮	⋮	⋮
Dead	364	NA
Alive	NA	678

Traditional statistical methods cannot be used in survival analysis due to not considering the censored data during survival prediction. Censored data should be taken into account during survival analysis, and the predictions for the right censored data should be made higher than or equal to the observed survival time to get more realistic results. Cox's proportional hazards model is one of the common methods proposed for survival analysis with censored data [Cox, 1972]. In Cox's model, the probability of experiencing the event at time t given that the event was not experienced until that time (i.e. the hazard function) is tried to be predicted. In the general formulation of Cox's model, the hazard function is proportional to a baseline

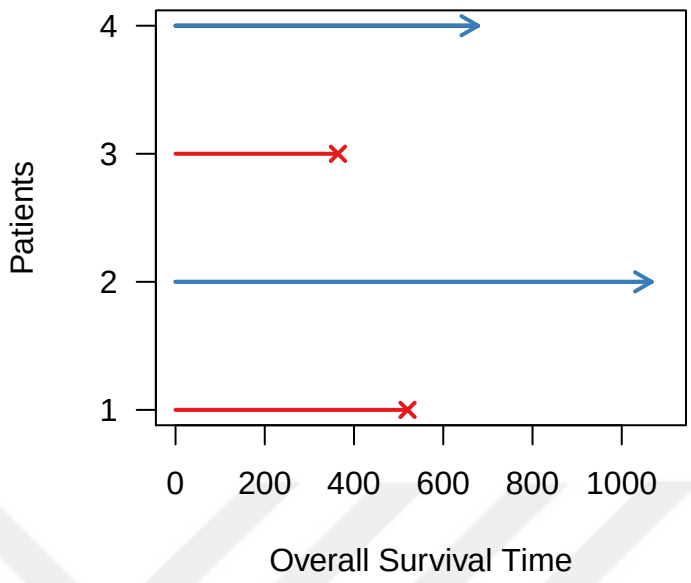

Figure 1.1: Illustration of the sample data. The blue lines represent the time to last follow-up for the right censored data, while the red lines represent the time to death.

hazard function, and the baseline hazard function is multiplied with an exponential function of the linear combination of all covariates. The main assumption of the Cox's model is that the baseline hazard is an unspecified function of time which forces all samples to be proportional to each other. These limiting assumptions may not be valid for all the time and may cause overfitting on the training data. Several extensions of Cox's model were proposed to overcome these drawbacks [Bakker et al., 2004; Cox and Oakes, 1984]. In these studies, it was proved that the idea of using only the relevant covariates for the survival prediction gives more robust predictive performances when low-dimensional input data is used. Following this idea, several machine learning algorithms were developed for survival prediction that eliminates the limiting affects of Cox's model and adds the non-linear effects of input data in a more robust manner [Evers and Messow, 2008; Ishwaran et al., 2008; Khan and Zubek, 2008; Shivaswamy et al., 2007; Van Belle et al., 2011a,b]

Survival Support Vector Machine (survival SVM) is the extension of the standard Support Vector Machine algorithm (SVM, [Cortes and Vapnik, 1995]) towards the

regression of censored data [Khan and Zubek, 2008; Shivaswamy et al., 2007]. It is a convex optimization problem which can handle non-linear problems with the help of kernel functions and has the ability to obtain a sparse (i.e. contains mostly zero values) global optimal solution with a fast training speed. The superiority of survival SVM approach against the standard survival analysis methods was shown in the literature [Van Belle et al., 2011a,b]. Since our methods are based on the survival SVM method, the details for support vector machines can be seen in Section 2.1.

Another approach proposed for regression of right-censored survival data is random survival forests (survival RF) [Ishwaran et al., 2008]. Random forest is an ensemble tree method that combines multiple weak decision trees [Breiman, 2001]. Survival RF extends the Breiman's random forest towards survival analysis by adding randomization to the tree growing and the splitting steps. Although splitting a node is based on the survival time and the censoring status, the success of survival RF algorithm highly depends on the censoring rate.

Several RF, SVM, and deep learning-based machine learning algorithms were later proposed for survival analysis of cancer patients using their genomic profiles and/or clinical information [Kiaee et al., 2016; Li et al., 2016; Mogensen and Gerds, 2013; Wang et al., 2017, 2016; Yousefi et al., 2017]. However, high-dimensional highly correlated structure of the genomic data creates some challenges. Generally, the number of training samples is significantly less than the number of covariates. Due to the high-dimensional and highly-correlated structure and the limited number of training samples, using genomic characterizations in machine learning studies is a challenging task. Most of the above mentioned studies usually cannot handle the high-dimensional structure of the genomic data and proved to be successful when using low-dimensional clinical variables as the input data [Yuan et al., 2014]. This phenomenon increases the importance of developing machine learning algorithms that can identify the informative parts of genomic characterizations with a limited number of training samples and use only these parts in the prediction step.

Predicting survival time of patients using their genomic characterizations is a crucial methodology to guide the diagnostic and therapeutic methods in cancer.

However, survival analysis by itself is insufficient in understanding the progression mechanisms of cancer. The biological mechanisms that affect the progression mechanisms, which determine the severity and the course of the disease, can be identified by using biological pathways/gene sets in survival analysis. A pathway/gene set is basically the series of actions among group of genes that has a specific role in human body, such as initiating the assembly of new molecules, activating or deactivating genes, and transmitting signals between cells. Dysfunctionalities in pathways/gene sets may cause serious diseases, such as cancer. If we map parts of genomic characterizations to these pathways/gene sets and select only a subset of them to perform survival prediction for the cancer patients, the selected pathways/gene sets can be used to understand the underlying mechanisms of cancer. Identification of informative biological mechanisms in cancer may lead to understand the root cause of cancer and to develop more effective strategies in diagnosis and treatment methods. Many machine learning algorithms were proposed to identify the pathways/gene sets associated with the survival time of cancer patients [Pang et al., 2010, 2012, 2011; Zhang et al., 2017]. Most of these algorithms usually follow a two-stage approach. We can divide the methods followed in these two-stage approaches into two. One of the approaches starts with selecting the survival related pathways/gene sets using a feature selection method. The feature selection method is followed by a learning model which uses genomic characterizations including only the genes in selected pathways/gene sets. In the second approach, a learning model is trained for each pathway/gene set separately, and then survival related pathways/gene sets are selected according to the predictive performance of the corresponding learning model. Both approaches have some drawbacks. The former might pick biologically unrelated genes due to highly correlated feature groups in genomic characterizations, and informative pathways/gene sets might not be identified. The latter might select pathways/gene sets that are very similar or related due to the analysis of each pathway/gene set separately. This is why it is important to develop machine learning algorithms that can integrate whole pathways/gene set collection into the model and identify a subset of pathways/gene sets without reducing the predictive

performance. There is a need for such single-stage approach in the literature.

There are several cancer-related studies, such as drug sensitivity prediction [Costello et al., 2014] and gene essentiality prediction [Gönen et al., 2017], that show the success of kernel-based algorithms in handling high-dimensional genomic data. The kernel functions basically define a similarity measure between pairs of samples. Using such kernel methods in machine learning algorithms that the genomic characterizations are used as input data is quite appropriate due to the dimensionality reduction feature of kernels. There are several machine learning algorithms that combine multiple kernel functions instead of using a single kernel, which is known as multiple kernel learning (MKL) [Gönen and Alpaydın, 2011]. To the best of our knowledge, there is only one study that uses MKL in survival analysis [Sinnott and Cai, 2018]. The algorithm proposed by Sinnott and Cai [2018] can be categorized into aforementioned two-stage approaches since the algorithm first selects kernels by considering each of them independently and then develops the prediction model using the combined kernels.

In this thesis, we focus on developing kernel-based machine learning algorithms that can pick the predictive pathways/gene sets from a given collection and train a predictive model on the subset of genomic features mapped to these selected pathways/gene sets. Our study has three main contributions. Unlike the existing approaches, our algorithms perform these knowledge extraction and survival analysis steps conjointly using multiple kernel learning. In addition, the existing multitask learning studies for survival analysis are not able to identify the relative importance of pathways/gene sets. We fill this gap by extending our multiple kernel learning-based survival analysis model towards multitask learning. We also propose a method that can identify the common biological mechanisms between multiple cancer types by extending our multitask learning-based algorithm towards clustering.

The outline of this thesis is as follows: In Chapter 2 we first review the support vector machine for binary classification problems and give the formulation and its derivations for survival support vector regression model. Then, we introduce our novel multiple kernel learning-based survival support vector regression model. In

Chapter 3, we give the details of our multitask multiple kernel learning algorithm for survival analysis, which is the extension of the model proposed in Chapter 2. Chapter 4 introduces the clustering algorithm for survival analysis of multiple cancer cohorts. Experimental results obtained with our proposed algorithms in Chapter 2–4 and their comparisons with the existing approaches in the literature are given in Chapter 5. We conclude our work and briefly discuss the possible future work in Chapter 6.

Chapter 2

MULTIPLE KERNEL LEARNING-BASED SURVIVAL ANALYSIS

We used support vector machine (SVM) as a base method to develop our algorithms. Therefore, we first give the basics of SVMs, kernel functions, and multiple kernel learning in Sections 2.1, 2.2, and 2.3, respectively. The details of our multiple kernel learning-based survival analysis algorithm can be seen in Section 2.4.

2.1 Support Vector Machines

SVM is a supervised machine learning algorithm that was first developed for binary classification problems [Cortes and Vapnik, 1995]. SVM models identify the optimum separating hyperplane that maximizes the margin between the data points belonging to different classes, by the help of a mapping function, so-called kernel trick. SVM models can be constructed as a convex quadratic optimization problem which enables to obtain global optimal solutions for the model parameters required to determine the discriminant function. One of the major advantages of SVM approach is that only a subset of model parameters, namely support vectors, is used to define the discriminant function. The time needed to solve the optimization problem is proportional to the number of support vectors obtained. SVMs can obtain a sparse solution set that significantly decreases the computational time. Another advantage of SVM approach comes from the ability of integrating kernel functions into the model. Integration of kernels into SVM models is an effective method used for capturing non-linear relations between input features. Instead of developing a non-linear discriminant function in the original space, we can map our model with the help of a kernel function to a higher dimensional space, where the discriminant

function can be written in a linear form. The strength of such kernel methods is related to the fact that the kernel functions significantly decrease the number of model parameters that are needed to be optimized and make them proportional to the number of samples instead of the input features [Alpaydin, 2020; Schölkopf and Smola, 2002].

SVMs can also be used for regression problems. However, the standard formulation for support vector regression (SVR) does not consider the censored targets. The success of SVM approach about handling non-linear problems using a sparse set of solution motivated researchers to extend SVR model towards survival analysis of censored targets (namely survival SVM) [Khan and Zubek, 2008; Shivaswamy et al., 2007].

The mathematical details of survival SVM for right censored targets can be constructed as follows. Let us define a sample of dataset as $\{(\boldsymbol{x}_i, \delta_i, y_i)\}_{i=1}^{N}$, where N is the number of training samples, $\boldsymbol{x}_i$ is the feature vector of sample i, $\delta_i \in \{0, 1\}$ is the binary indicator variable that represents whether the overall survival time of sample i is censored or not (i.e. if censored, δ_i is 1, 0 otherwise), and $y_i \in \mathbb{R}$ is the overall survival time of sample i (i.e. time to last follow-up if censored or time to death if uncensored). The estimation function for the overall survival time of a sample can be formulated as

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b.$$

The parameters of the estimation function f can be obtained by solving the following optimization problem:

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

where $\boldsymbol{w}$ is the set of weights assigned to features, C is the non-negative regularization parameter, $\boldsymbol{\xi}^+$ and $\boldsymbol{\xi}^-$ are the sets of slack variables, D is the number of input features, ϵ is the non-negative tube width parameter, and b is the bias parameter.

The aim of the optimization problem 2.1 is to find the estimation function f using a subset of all training samples with a small estimation error. The model complexity is regularized with the term $\boldsymbol{w}^\top \boldsymbol{w}$. Minimizing $\boldsymbol{w}^\top \boldsymbol{w}$ corresponds to maximizing the margin, and the model complexity decreases as the margin increases. There is a trade-off between the model complexity and the regression error. A smaller regression error corresponds to higher model complexity, where the number of training samples used to find the estimation function f increases as the model complexity increases. The sparsity of the regression model is achieved by not penalizing the errors below the tube width parameter ϵ that is chosen a priori. The errors made above the tube width parameter are defined with the non-negative slack variables (i.e. ξ_i^+ and ξ_i^-). The balance between the training error and the model complexity is controlled with the regularization parameter C. The optimization problem 2.1 considers both censored and uncensored targets. If the predicted survival time of a censored sample is greater than its observed survival time, the error made for that censored sample is not penalized by multiplying the error term ξ_i^- with $(1 - \delta_i)$, that is, $(1 - \delta_i)$ corresponds to zero for censored observations. By doing so, the observed survival time of a censored sample is considered as a lower bound for the corresponding prediction. Due to the nature of the optimization problem 2.1, both slack variables cannot be greater than zero. If one of the slack variables of a sample is greater than zero, the other slack variable of the same sample is forced to be zero since the regression model is a minimization problem, and the lower bound for that variable becomes negative. This fact ensures that no penalty occurs for censored samples in the objective function if their predicted survival times are higher than their observed values.

We solve the dual formulation of the given problem to be able to integrate kernel functions into the model and to decrease the number of decision variables. The dual formulation of an optimization problem is obtained by first constructing its

Lagrangian function. The Lagrangian function of an optimization problem can be developed by introducing a non-negative dual set of variables for each corresponding constraint set. The formulation of the Lagrangian function for the primal optimization problem 2.1 is as follows:

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

where $\boldsymbol{\alpha}^+,\ \boldsymbol{\alpha}^-,\ \boldsymbol{\beta}^+,\ \boldsymbol{\beta}^-$ are the corresponding dual sets of variables for each constraint set given in the optimization problem 2.1, and each dual variable has to be non-negative, i.e., $\boldsymbol{\alpha}_i^+,\ \boldsymbol{\alpha}_i^-,\ \boldsymbol{\beta}_i^+,\ \boldsymbol{\beta}_i^- \geq 0$. The obtained Lagrangian function is a convex quadratic optimization problem. Therefore, $\mathcal{L}$ can be solved using Karush-Kuhn-Tucker (KKT) optimality conditions, where the derivatives with respect to the primal decision variables (i.e. $\boldsymbol{w},\ b,\ \xi_i^+,\ \xi_i^-$) must be equal to zero;

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = \sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-)\boldsymbol{x}_i$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-) = 0$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i^+} = 0 \Rightarrow C = \alpha_i^+ + \beta_i^+ \qquad \forall i$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i^-} = 0 \Rightarrow C(1-\delta_i) = \alpha_i^- + \beta_i^- \qquad \forall i.$$

We can obtain the dual optimization problem when we plug these partial derivatives

back into the Lagrangian function;

$$\begin{aligned}
\text{minimize} \quad & -\sum_{i=1}^{N} y_i(\alpha_i^+ - \alpha_i^-) + \epsilon \sum_{i=1}^{N}(\alpha_i^+ + \alpha_i^-) \\
& + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)\boldsymbol{x}_i^\top \boldsymbol{x}_j \\
\text{with respect to} \quad & \boldsymbol{\alpha}^+ \in \mathbb{R}^N, \quad \boldsymbol{\alpha}^- \in \mathbb{R}^N \\
\text{subject to} \quad & \sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-) = 0 \\
& C \geq \alpha_i^+ \geq 0 \qquad \forall i \\
& C(1-\delta_i) \geq \alpha_i^- \geq 0 \qquad \forall i,
\end{aligned} \tag{2.2}$$

where the number of decision variables is $2N$ instead of $(D + 2N + 1)$ which is the number of decision variables of the primal optimization problem 2.1. The dual formulation 2.2 can be kernelized by replacing the term $\boldsymbol{x}_i^\top \boldsymbol{x}_j$ with a kernel function $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, so that we can capture non-linearity with survival SVM formulation. Since we can describe the primal variable $\boldsymbol{w}$ as the linear combination of the dual variables, the estimation function f can be reformulated as

$$f(\boldsymbol{x}) = \sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-)\boldsymbol{x}_i^\top \boldsymbol{x} + b,$$

where we can replace the term $\boldsymbol{x}_i^\top \boldsymbol{x}$ with the kernel function $k(\boldsymbol{x}_i, \boldsymbol{x})$. We can obtain the bias parameter b using KKT optimality conditions which state that at optimality, multiplication of each constraint with the corresponding dual variable must be equal to zero;

$$\begin{aligned}
\alpha_i^+(\epsilon + \xi_i^+ - y_i + \boldsymbol{w}^\top \boldsymbol{x}_i + b) = 0 \qquad & \forall i \\
\alpha_i^-(\epsilon + \xi_i^- + y_i - \boldsymbol{w}^\top \boldsymbol{x}_i - b) = 0 \qquad & \forall i \\
(C - \alpha_i^+)\xi_i^+ = 0 \qquad & \forall i \\
(C(1-\delta_i) - \alpha_i^-)\xi_i^- = 0 \qquad & \forall i.
\end{aligned}$$

We can conclude from these equalities that: i) Only the samples with with an error value greater than zero, i.e., $\xi_i^+(\xi_i^-) \geq 0$, have a corresponding $\alpha_i^+(\alpha_i^-) = C$, ii)

both α_i^+ and α_i^- cannot be simultaneously non-zero, iii) for $\alpha_i^+(\alpha_i^-) \in (0, C)$ the corresponding $\xi_i^+(\xi_i^-) = 0$, so that we can obtain b as follows:

$$b = y_i - \boldsymbol{w}^\top \boldsymbol{x}_i - \epsilon \qquad \forall \alpha_i^+ \in (0, C)$$
$$b = y_i - \boldsymbol{w}^\top \boldsymbol{x}_i + \epsilon \qquad \forall \alpha_i^- \in (0, C).$$

The optimization problem 2.2 yields a sparse set of solution with respect to $\boldsymbol{\alpha}^+ + \boldsymbol{\alpha}^-$, where all the samples that have an error less than the tube width parameter ϵ the corresponding $\boldsymbol{\alpha}^+ + \boldsymbol{\alpha}^-$ variables are equal to zero. This can be explained with the above mentioned KKT optimality conditions.

2.2 Kernel Functions

In machine learning problems, non-linearity between the model parameters can be captured by mapping the problem to a higher dimensional space, and a linear model, which corresponds to a non-linear model in the original space, can be used in this new space to solve the original problem. The transformation of the problem requires an explicit mapping function, which is hard to calculate and increases the dimensionality of the problem enormously. Instead of using such mapping functions, it is much easier to use kernel functions directly in the original space. This method is called as kernelization.

Let us consider the optimization problem 2.2. We can obtain the non-linearity for this problem by replacing the term $\boldsymbol{x}_i^\top \boldsymbol{x}_j$ with $\Phi(\boldsymbol{x}_i)^\top \Phi(\boldsymbol{x}_j)$, where $\Phi()$ is the mapping function. Instead, we can use the kernel function $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ to obtain the non-linearity.

There are several kernel functions, such as linear, polynomial, and Gaussian kernels. We use the following Gaussian kernel function in our algorithms due to its success in cancer-related problems [Costello et al., 2014; Gönen et al., 2017]:

$$k_{\mathcal{G}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\big(-(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top (\boldsymbol{x}_i - \boldsymbol{x}_j)/(2\sigma^2)\big), \tag{2.3}$$

where σ is the kernel width parameter.

2.3 Multiple Kernel Learning

The success of the kernel-based machine learning algorithms is highly affected by the kernel function selected. Therefore, which kernel function to use in these algorithms has a vital importance. In kernel-based methods, using a cross-validation technique on the training data, the best kernel function that gives the best predictive performance among several kernels can be selected and it can be used to perform predictions on the test data. However, the selected kernel function might not be enough to capture the non-linearity of the given problem. Instead of using a single kernel function, several methods have been proposed that uses the combination of multiple kernel functions. There is no single kernel function that gives the best similarity between all pairs of samples. Accordingly, it is better to define multiple kernels and pick the best one or find the best combination among them.

The multiplication of a valid kernel with a constant a, and multiplication or summation of valid kernels also yield valid kernels;

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = a k_1(\boldsymbol{x}_i, \boldsymbol{x}_j)$$
$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = k_1(\boldsymbol{x}_i, \boldsymbol{x}_j) + k_1(\boldsymbol{x}_i, \boldsymbol{x}_j)$$
$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = k_1(\boldsymbol{x}_i, \boldsymbol{x}_j) k_2(\boldsymbol{x}_i, \boldsymbol{x}_j).$$

Hence, we can replace a kernel function with the convex combination of multiple kernels, which is known as multiple kernel learning (MKL) [Gönen and Alpaydın, 2011], as follows:

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

where P is the number of feature representations, η_m is the weight of $k_m(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\boldsymbol{\eta} \geq 0$.

Different kernels can be defined using different notions of similarity or different feature representations. In either case, it is not known which one or which combination gives the best similarity. Therefore, for example in SVM method, kernel weights $\boldsymbol{\eta}$ must be optimized with the SVM parameters $\boldsymbol{\alpha}$.

2.4 Pathway/Gene Set-based Survival Analysis Using Multiple Kernel Learning

As mentioned in the introduction chapter, identification of key biological mechanisms in cancer has a vital importance in understanding the progression of the disease. For this purpose, we extend survival SVM formulation [Khan and Zubek, 2008; Shivaswamy et al., 2007] towards a one-step MKL formulation (named as Path2Surv). Our algorithm identifies informative pathways/gene sets on predicting overall survival time of cancer patients using their gene expression profiles of the given cancer cohort. We define multiple kernels on each pathway/gene set and integrate them into the model. Figure 2.1 gives the summary of our Path2Surv algorithm. Our Path2Surv algorithm first calculates multiple kernel functions on pathways/gene sets. Each kernel function defines a similarity measure between each pair of patients, using gene expression profiles containing only the genes included in the corresponding pathway/gene set. Our algorithm tries to find the optimal combination of these kernel functions by assigning zero to the most of the kernel weights, so that the uninformative pathways/gene sets are eliminated from the model. We calculate the weighted sum of these kernel functions to obtain a better similarity measure and use this combined kernel function to predict the survival times of cancer patients.

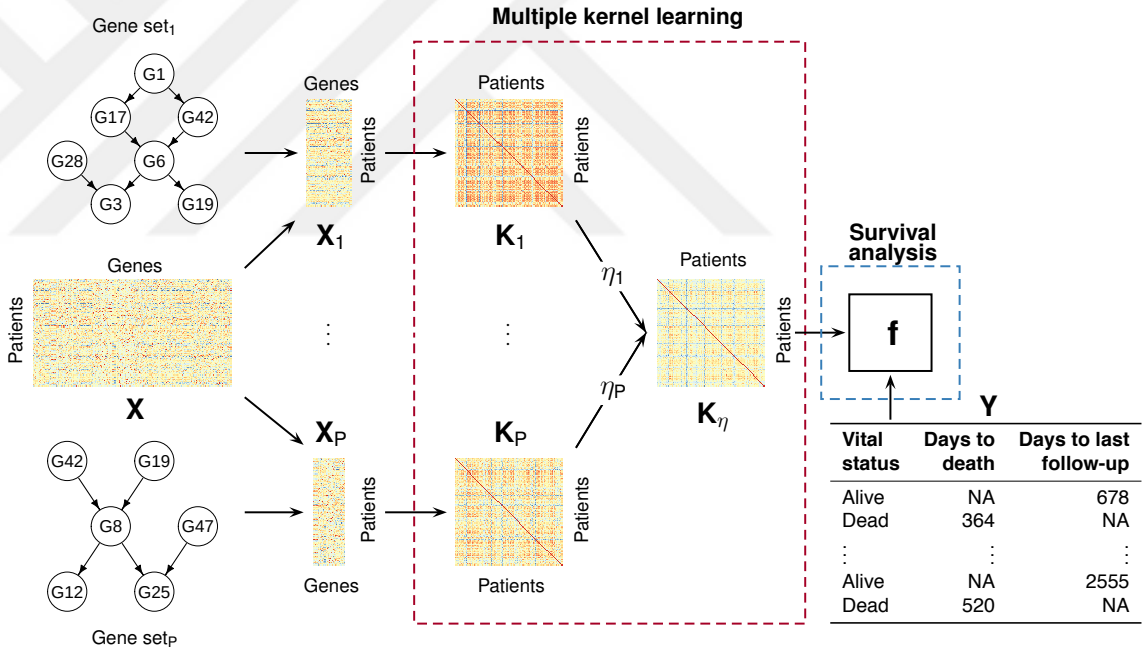

Figure 2.1: The overview of our proposed Path2Surv algorithm. Path2Surv algorithm takes the matrix of gene expression profiles of patients, denoted as $\mathbf{X}$, survival characteristics of cancer patients, denoted as $\mathbf{Y}$ (i.e. vital status, days to death, and days to last follow-up), and a pathway/gene set collection as its input. It then calculates distinct kernel matrices, denoted as $\mathbf{K}_1, \ldots, \mathbf{K}_P$, for input pathways/gene sets on gene expression slices, denoted as $\mathbf{X}_1, \ldots, \mathbf{X}_P$, taken from the matrix of gene expression profiles. These multiple kernel matrices are combined with a weighted sum to obtain a more informative kernel matrix, denoted as $\mathbf{K}_\eta$, between pairs of patients. The optimized kernel matrix is then used to learn a function, denoted as f, to predict overall survival times of out-of-sample cancer patients.

As in optimization model 2.1, we represent the training dataset as $\{(\boldsymbol{x}_i, \delta_i, y_i)\}_{i=1}^{N}$, where N is the number of patients, $\boldsymbol{x}_i$ is the gene expression profile of tumour biopsied from patient i, $\delta_i \in \{0, 1\}$ is the binary indicator variable that represents whether the overall survival time of patient i is censored or not (i.e. if censored, δ_i is 1, 0 otherwise), and $y_i \in \mathbb{R}$ is the overall survival time of patient i (i.e. time to last follow-up if censored or time to death if uncensored).

The optimization problem to obtain the parameters of the estimation function $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$ can be formulated as

$$
\begin{aligned}
\text{minimize} \quad & \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + C\sum_{i=1}^{N}(\xi_i^+ + (1-\delta_i)\xi_i^-) \\
\text{with respect to} \quad & \boldsymbol{w} \in \mathbb{R}^D, \ \boldsymbol{\xi}^+ \in \mathbb{R}^N, \ \boldsymbol{\xi}^- \in \mathbb{R}^N, \ b \in \mathbb{R} \\
\text{subject to} \quad & \epsilon + \xi_i^+ \geq y_i - \boldsymbol{w}^\top \boldsymbol{x}_i - b \qquad \forall i \\
& \epsilon + \xi_i^- \geq \boldsymbol{w}^\top \boldsymbol{x}_i + b - y_i \qquad \forall i \\
& \xi_i^+ \geq 0 \qquad \forall i \\
& \xi_i^- \geq 0 \qquad \forall i,
\end{aligned}
\tag{2.4}
$$

where $\boldsymbol{w}$ is the set of weights assigned to features, C is the non-negative regularization parameter, $\boldsymbol{\xi}^+$ and $\boldsymbol{\xi}^-$ are the sets of slack variables, D is the number of input features, that is, the number of genes in gene expression profiles, ϵ is the non-negative tube width parameter, and b is the bias parameter.

We develop the corresponding dual formulation of the optimization problem 2.4 following the same Lagrangian dual method applied for the survival SVM formulation. However, survival SVM algorithm uses a single kernel function in the dual formulation which highly affects the success of the algorithm. Instead, we use multiple kernel functions defined on pathways/gene sets which enables us to obtain more robust survival predictors with an ability of knowledge extraction. We assume that there are P different kernel functions corresponding to each pathway/gene set. We replace the single kernel function, $k(\cdot, \cdot)$, used in the optimization problem 2.2 with the weighted sum of these multiple kernel functions (i.e. $\{k_m(\cdot, \cdot)\}_{m=1}^{P}$) by using a convex combination rule. We use the following dual optimization model to obtain

the weights of multiple kernel functions:

$$\begin{aligned} \text{minimize} \quad & J(\boldsymbol{\eta}) \\ \text{with respect to} \quad & \boldsymbol{\eta} \in \mathbb{R}^P \\ \text{subject to} \quad & \sum_{m=1}^{P} \eta_m = 1 \\ & \eta_m \geq 0 \qquad \forall m, \end{aligned} \tag{2.5}$$

where $\boldsymbol{\eta}$ represents the kernel weights, and $J(\boldsymbol{\eta})$ is the following optimization problem modified with MKL:

$$\begin{aligned} \text{minimize} \quad & -\sum_{i=1}^{N} y_i(\alpha_i^+ - \alpha_i^-) + \epsilon \sum_{i=1}^{N}(\alpha_i^+ + \alpha_i^-) \\ & + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)k_{\boldsymbol{\eta}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \text{with respect to} \quad & \boldsymbol{\alpha}^+ \in \mathbb{R}^N, \;\; \boldsymbol{\alpha}^- \in \mathbb{R}^N \\ \text{subject to} \quad & \sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-) = 0 \\ & C \geq \alpha_i^+ \geq 0 \qquad \forall i \\ & C(1-\delta_i) \geq \alpha_i^- \geq 0 \qquad \forall i. \end{aligned} \tag{2.6}$$

In the inner optimization problem 2.6, we replace the term $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \boldsymbol{x}_j$ with $k_{\boldsymbol{\eta}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{m=1}^{P} \eta_m k_m(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The equality constraint in the optimization problem 2.5 is called as the unit simplex constraint which enforces ℓ_1-norm on the kernel weights and ensures to obtain a sparse solution set for the kernel weights.

The overall optimization problem 2.5 cannot be solved globally since it is not jointly convex with respect to decision variables $\boldsymbol{\eta}$ and $\{\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-\}$. Instead of solving the problem 2.5 directly, we follow an alternating optimization approach since the outer optimization problem is convex with respect to $\boldsymbol{\eta}$ and the inner optimization problem 2.6 is convex with respect to $\{\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-\}$. Our alternating optimization approach is inspired from the group Lasso MKL algorithm which is developed for binary classification problems [Xu et al., 2010].

2.4.1 Solution Methodology

Xu et al. [2010] developed a closed-form update equation for kernel weights by showing the connection between MKL and group Lasso. Here, we show the same relation for regression problems using a similar approach to obtain a closed-form update equation.

As derived by Xu et al. [2010], we first show that the following optimization problem is equivalent to the optimization problem 2.4:

$$\begin{aligned}
\text{minimize} \;\; & \frac{1}{2}\sum_{m=1}^{P}\eta_m \boldsymbol{w}_m^\top \boldsymbol{w}_m + C\sum_{i=1}^{N}(\xi_i^+ + (1-\delta_i)\xi_i^-) \\
\text{with respect to} \;\; & \{\boldsymbol{w}_m \in \mathbb{R}^{D_m}\}_{m=1}^{P},\;\; \boldsymbol{\xi}^+ \in \mathbb{R}^N,\;\; \boldsymbol{\xi}^- \in \mathbb{R}^N,\;\; b \in \mathbb{R} \\
\text{subject to} \;\; & \epsilon + \xi_i^+ \geq y_i - \sum_{m=1}^{P}\eta_m \boldsymbol{w}_m^\top \boldsymbol{x}_i - b \qquad \forall i \\
& \epsilon + \xi_i^- \geq \sum_{m=1}^{P}\eta_m \boldsymbol{w}_m^\top \boldsymbol{x}_i + b - y_i \qquad \forall i \\
& \xi_i^+ \geq 0 \qquad \forall i \\
& \xi_i^- \geq 0 \qquad \forall i,
\end{aligned} \tag{2.7}$$

where $\boldsymbol{w}_m$ is weighted by η_m and $\boldsymbol{\eta}$ lies on a simplex, i.e., $\{\boldsymbol{\eta} \in \mathbb{R}^P : \mathbf{1}^\top \boldsymbol{\eta} = 1, \boldsymbol{\eta} \geq 0\}$. We derive the dual formulation of the primal model 2.7 as follows:

$$\begin{aligned}
\mathcal{L} = \frac{1}{2}\sum_{m=1}^{P}\eta_m \boldsymbol{w}_m^\top \boldsymbol{w}_m + C\sum_{i=1}^{N}(\xi_i^+ + (1-\delta_i)\xi_i^-) - \sum_{i=1}^{N}\alpha_i^+(\epsilon + \xi_i^+ - y_i + \sum_{m=1}^{P}\eta_m \boldsymbol{w}_m^\top \boldsymbol{x}_i + b) \\
- \sum_{i=1}^{N}\beta_i^+\xi_i^+ - \sum_{i=1}^{N}\alpha_i^-(\epsilon + \xi_i^- - \sum_{m=1}^{P}\eta_m \boldsymbol{w}_m^\top \boldsymbol{x}_i - b + y_i) - \sum_{i=1}^{N}\beta_i^-\xi_i^-,
\end{aligned}$$

and the partial derivatives with respect to the primal decision variables are:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_m} = 0 \Rightarrow \boldsymbol{w}_m = \sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-)\boldsymbol{x}_i \qquad \forall m \\
\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-) = 0 \\
\frac{\partial \mathcal{L}}{\partial \xi_i^+} = 0 \Rightarrow C = \alpha_i^+ + \beta_i^+ \qquad \forall i \\
\frac{\partial \mathcal{L}}{\partial \xi_i^-} = 0 \Rightarrow C(1-\delta_i) = \alpha_i^- + \beta_i^- \qquad \forall i.
\end{aligned}$$

We can obtain the dual optimization problem when we plug these partial derivatives back into the Lagrangian function;

$$\begin{aligned}
\text{minimize} \quad & -\sum_{i=1}^{N} y_i(\alpha_i^+ - \alpha_i^-) + \epsilon \sum_{i=1}^{N}(\alpha_i^+ + \alpha_i^-) \\
& + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)\sum_{m=1}^{P}\eta_m k_m(\boldsymbol{x}_i^\top \boldsymbol{x}_j) \\
\text{with respect to} \quad & \boldsymbol{\alpha}^+ \in \mathbb{R}^N, \;\; \boldsymbol{\alpha}^- \in \mathbb{R}^N \\
\text{subject to} \quad & \sum_{i=1}^{N}(\alpha_i^+ - \alpha_i^-) = 0 \\
& C \geq \alpha_i^+ \geq 0 \qquad \forall i \\
& C(1-\delta_i) \geq \alpha_i^- \geq 0 \qquad \forall i.
\end{aligned} \tag{2.8}$$

We can observe that the dual formulation 2.8 is equivalent to the optimization problem 2.6. After showing the equivalence of these optimization models, we can obtain the closed-form update equation for the kernel weights as follows.

Let us define $\widetilde{\boldsymbol{w}}_m = \eta_m \boldsymbol{w}$ and rewrite the optimization problem 2.7 as:

$$\begin{aligned}
\text{minimize} \quad & \frac{1}{2}\sum_{m=1}^{P}\frac{1}{\eta_m}\widetilde{\boldsymbol{w}}_m^\top \widetilde{\boldsymbol{w}}_m + C\sum_{i=1}^{N}(\xi_i^+ + (1-\delta_i)\xi_i^-) \\
\text{with respect to} \quad & \{\widetilde{\boldsymbol{w}}_m \in \mathbb{R}^{D_m}\}_{m=1}^{P}, \;\; \boldsymbol{\xi}^+ \in \mathbb{R}^N, \;\; \boldsymbol{\xi}^- \in \mathbb{R}^N, \;\; b \in \mathbb{R} \\
\text{subject to} \quad & \epsilon + \xi_i^+ \geq y_i - \sum_{m=1}^{P}\widetilde{\boldsymbol{w}}_m^\top \boldsymbol{x}_i - b \qquad \forall i \\
& \epsilon + \xi_i^- \geq \sum_{m=1}^{P}\widetilde{\boldsymbol{w}}_m^\top \boldsymbol{x}_i + b - y_i \qquad \forall i \\
& \xi_i^+ \geq 0 \qquad \forall i \\
& \xi_i^- \geq 0 \qquad \forall i,
\end{aligned} \tag{2.9}$$

where $\{\boldsymbol{\eta} \in \mathbb{R}^P : \; \mathbf{1}^\top \boldsymbol{\eta} = 1, \; \boldsymbol{\eta} \geq 0\}$. If we minimize 2.9 with respect to $\boldsymbol{\eta}$ following the procedure given in Appendix A, we obtain the following update equation for the kernel weights:

$$\eta_m = \frac{\sqrt{\widetilde{\boldsymbol{w}}_m^\top \widetilde{\boldsymbol{w}}_m}}{\sum_{o=1}^{P}\sqrt{\widetilde{\boldsymbol{w}}_o^\top \widetilde{\boldsymbol{w}}_o}}, \tag{2.10}$$

and we can calculate $\widetilde{\boldsymbol{w}}_m^\top \widetilde{\boldsymbol{w}}_m$ as

$$\widetilde{\boldsymbol{w}}_m^\top \widetilde{\boldsymbol{w}}_m = \eta_m^2 \sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)k_m(\boldsymbol{x_i}, \boldsymbol{x_j}). \tag{2.11}$$

Xu et al. [2010] also stated that an alternating optimization that optimizes $\boldsymbol{\eta}$ with fixed $\widetilde{\boldsymbol{w}}_m$ and optimizes $\widetilde{\boldsymbol{w}}_m$ with fixed $\boldsymbol{\eta}$ converges to a global optimal solution, since the optimization problem 2.9 is convex with respect to $\boldsymbol{\eta}$ and the dual formulation 2.8 is convex with respect to $\{\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-\}$ variables.

Therefore, we follow a similar alternating optimization approach that we update the kernel weights using Equation 2.12. Our algorithm starts with initializing the kernel weights to uniform values, that is, $\eta_m^{(0)} = 1/P$ is the weight of kernel m at iteration (0). We solve the inner optimization problem given in 2.6 with fixed $\eta_m^{(s)}$ values to obtain the support vector variables $\{\boldsymbol{\alpha}^{+(s)}, \boldsymbol{\alpha}^{-(s)}\}$ at iteration s. At the next iteration $(s+1)$, kernel weights are updated using the following kernel update equation:

$$\eta_m^{(s+1)} = \frac{\eta_m^{(t)} \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i^{+(s)} - \alpha_i^{-(s)})(\alpha_j^{+(s)} - \alpha_j^{-(s)})k_m(\boldsymbol{x_i}, \boldsymbol{x_j})}}{\sum_{o=1}^{P}\eta_o^{(s)} \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i^{+(s)} - \alpha_i^{-(s)})(\alpha_j^{+(s)} - \alpha_j^{-(s)})k_o(\boldsymbol{x_i}, \boldsymbol{x_j})}}, \tag{2.12}$$

which can be obtained by plugging Equation 2.11 back into Equation 2.10. These alternating steps monotonically decrease the objective function and we repeat this procedure until the convergence. At the end, we obtain a sparse set of $\boldsymbol{\eta}$ values showing the kernel weights of the corresponding pathways/gene sets. The non-zero kernel weights show us which pathways/gene sets are included in the final model and informative during survival prediction of cancer patients.

Chapter 3

MULTITASK MULTIPLE KERNEL LEARNING ALGORITHM FOR SURVIVAL ANALYSIS

Multitask learning (MTL) is a machine learning algorithm that increases the predictive performances and the accuracy of the learning models by sharing information between related problems. The idea of modelling multiple cancer types simultaneously was shown to be successful in different cancer studies [Costello et al., 2014; Gönen et al., 2017]. The fact that different cancer types might have same or very similar underlying biological mechanisms has been shown in the studies that jointly model multiple cancer cohorts, known as pan-cancer studies [Anaya et al., 2016; Choi et al., 2014; Damrauer et al., 2014; Hoadley et al., 2018, 2014; Khirade et al., 2015; Lawrence et al., 2014; Pappa et al., 2015; The Cancer Genome Atlas Research Network et al., 2013; Wan et al., 2015; Yang et al., 2014]. Survival analysis studies that utilize the commonalities between multiple cancer types were also proposed [Li et al., 2016; Wang et al., 2017]. However, these existing methods are not eligible for the identification of informative pathways/gene sets on survival prediction.

In this thesis, we combine survival analysis, multiple kernel learning, and multitask learning in a unified formulation to identify survival-related pathways/gene sets (named as Path2MSurv, see Figure 3.1). We extend our MKL algorithm for survival analysis towards MTL based on the assumption that all cancer types included in this study have the same or very similar underlying biological mechanisms. Following this idea, we aim to identify common underlying biological mechanisms among these cancer types using the genomic characterizations of patients from multiple cohorts.

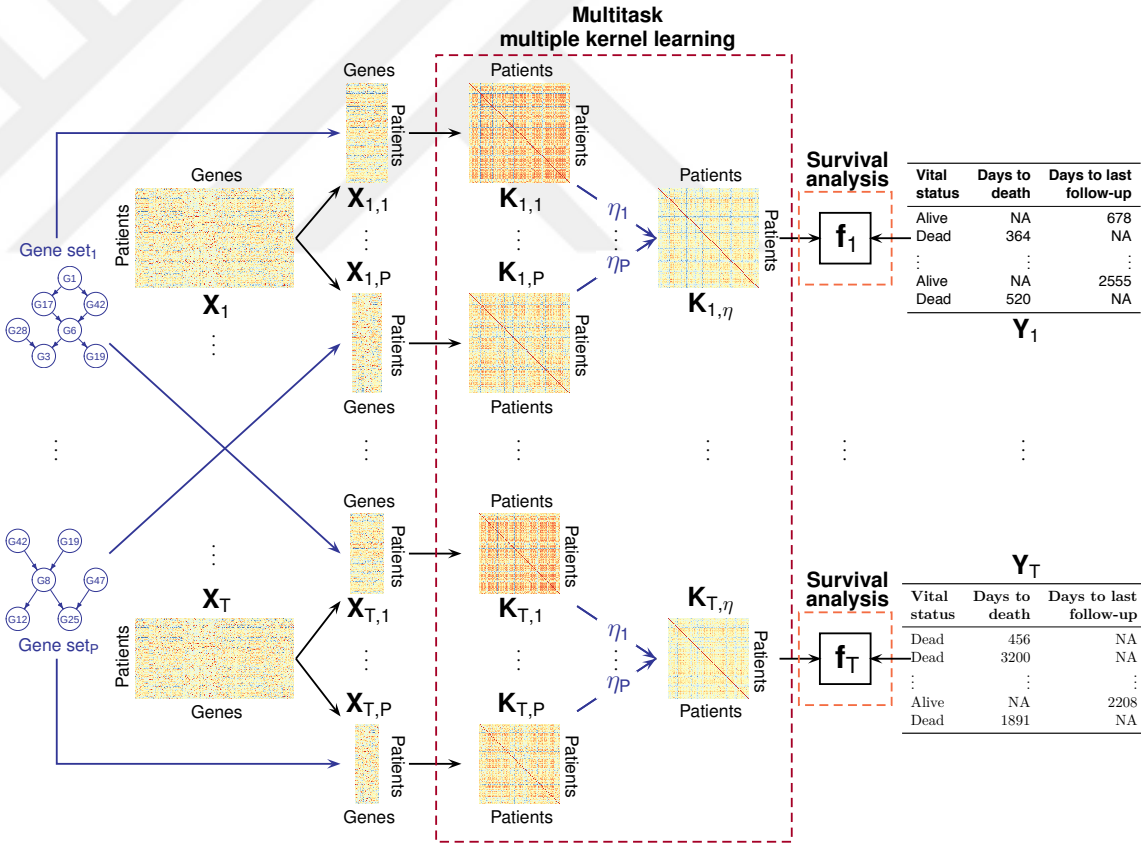

Figure 3.1: The overview of the proposed Path2MSurv algorithm. Path2MSurv algorithm takes gene expression profiles of patients from each cohort, i.e., $\{\mathbf{X}_t\}_{t=1}^{T}$, a pathway/gene set collection with P pathways/gene sets, and clinical information including vital status, days to death, and days to last follow-up, i.e., $\{\mathbf{Y}_t\}_{t=1}^{T}$, as its inputs. It then calculates kernel matrices, i.e., $\{\mathbf{K}_{t,p}\}_{p=1}^{P}$, on data matrix slices, i.e., $\{\mathbf{X}_{t,p}\}_{t=1,p=1}^{T,P}$, obtained by mapping pathways/gene sets on gene expression profiles. The weighted sums of these kernel matrices, i.e., $\{\mathbf{K}_{t,\eta}\}_{t=1}^{T}$, are used to predict survival times of cancer patients using the prediction functions, i.e., $\{f\}_{t=1}^{T}$.

We use $\{\{(\boldsymbol{x}_{ti}, \delta_{ti}, y_{ti})\}_{i=1}^{N_t}\}_{t=1}^{T}$ to represent the training datasets over multiple cancer datasets, where T is the number of cohorts, N_t is the total number of patients in cohort t, $\boldsymbol{x}_{ti}$ is the gene expression profile of tumour biopsied from patient i of cohort t, $\delta_{ti} \in \{0, 1\}$ is the binary indicator variable that represents whether the overall survival time of patient i of cohort t is censored or not (i.e. if censored, δ_{ti} is 1, 0 otherwise), and $y_{ti} \in \mathbb{R}$ is the overall survival time of patient i of cohort t (i.e. time to last follow-up if censored or time to death if uncensored). Then, the estimation function f can be formulated as follows:

$$f_t(\boldsymbol{x}) = \boldsymbol{w_t}^\top \boldsymbol{x} + b_t.$$

The primal optimization problem to obtain the estimation function parameters can be written as

$$\begin{aligned}
\text{minimize} \quad & \sum_{t=1}^{T}\left[\frac{1}{2}\boldsymbol{w}_t^\top \boldsymbol{w}_t + C\sum_{i=1}^{N_t}(\xi_{ti}^{+} + (1-\delta_{ti})\xi_{ti}^{-})\right] \\
\text{with respect to} \quad & \{\boldsymbol{w}_t \in \mathbb{R}^{D_t}\}_{t=1}^{T}, \ \{\boldsymbol{\xi}_t^{+} \in \mathbb{R}^{N_t}\}_{t=1}^{T}, \ \{\boldsymbol{\xi}_t^{-} \in \mathbb{R}^{N_t}\}_{t=1}^{T}, \ \{b_t \in \mathbb{R}\}_{t=1}^{T} \\
\text{subject to} \quad & \epsilon + \xi_{ti}^{+} \geq y_{ti} - \boldsymbol{w}_t^\top \boldsymbol{x}_{ti} - b_t \quad \forall(t,i) \\
& \epsilon + \xi_{ti}^{-} \geq \boldsymbol{w}_t^\top \boldsymbol{x}_{ti} + b_t - y_{ti} \quad \forall(t,i) \\
& \xi_{ti}^{+} \geq 0 \quad \forall(t,i) \\
& \xi_{ti}^{-} \geq 0 \quad \forall(t,i),
\end{aligned} \tag{3.1}$$

where $\boldsymbol{w}_t$ is the set of weights assigned to features for cohort t, C is the non-negative regularization parameter, $\boldsymbol{\xi}_t^{+}$ and $\boldsymbol{\xi}_t^{-}$ are the sets of slack variables for cohort t, D_t is the number of input features for cohort t, that is, the number of genes in gene expression profiles, ϵ is the non-negative tube width parameter, and b_t is the bias parameter for cohort t.

We derive the dual formulation of the optimization model 3.1 by first deriving

the corresponding Lagrangian function;

$$\mathcal{L} = \sum_{t=1}^{T} \Bigg(\frac{1}{2} \boldsymbol{w_t}^\top \boldsymbol{w_t} + C \sum_{i=1}^{N_t} (\xi_{ti}^+ + (1-\delta_{ti})\xi_{ti}^-) - \sum_{i=1}^{N_t} \alpha_{ti}^+ (\epsilon + \xi_{ti}^+ - y_{ti} + \boldsymbol{w_t}^\top \boldsymbol{x}_{ti} + b) \\ - \sum_{i=1}^{N_t} \beta_{ti}^+ \xi_{ti}^+ - \sum_{i=1}^{N_t} \alpha_{ti}^- (\epsilon + \xi_{ti}^- - \boldsymbol{w_t}^\top \boldsymbol{x}_{ti} - b + y_{ti}) - \sum_{i=1}^{N_t} \beta_{ti}^- \xi_{ti}^- \Bigg),$$

where $\{\boldsymbol{\alpha}_t^+,\ \boldsymbol{\alpha}_t^-,\ \boldsymbol{\beta}_t^+,\ \boldsymbol{\beta}_t^-\}_{t=1}^T$ are the corresponding dual sets of variables for each constraint set given in the optimization problem 3.1 and $\alpha_{ti}^+,\ \alpha_{ti}^-,\ \beta_{ti}^+,\ \beta_{ti}^- \geq 0$. The derivatives of the Lagrangian function with respect to the primal decision variables (i.e. $\boldsymbol{w}_t,\ b_t,\ \xi_{ti}^+,\ \xi_{ti}^-$) must be equal to zero;

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_t} = 0 &\Rightarrow \boldsymbol{w}_t = \sum_{i=1}^{N_t} (\alpha_{ti}^+ - \alpha_{ti}^-) \boldsymbol{x}_{ti} \qquad \forall t \\
\frac{\partial \mathcal{L}}{\partial b_t} = 0 &\Rightarrow \sum_{i=1}^{N_t} (\alpha_{ti}^+ - \alpha_{ti}^-) = 0 \qquad \forall t \\
\frac{\partial \mathcal{L}}{\partial \xi_{ti}^+} = 0 &\Rightarrow C = \alpha_{ti}^+ + \beta_{ti}^+ \qquad \forall (t,i) \\
\frac{\partial \mathcal{L}}{\partial \xi_{ti}^-} = 0 &\Rightarrow C(1-\delta_{ti}) = \alpha_{ti}^- + \beta_{ti}^- \qquad \forall (t,i).
\end{aligned}$$

We can obtain the dual formulation when we plug these partial derivatives back into the Lagrangian function as follows:

$$\begin{aligned}
\text{minimize} \quad & \sum_{t=1}^{T} J_t(\boldsymbol{\eta}) \\
\text{with respect to} \quad & \boldsymbol{\eta} \in \mathbb{R}^P \\
\text{subject to} \quad & \sum_{m=1}^{P} \eta_m = 1 \\
& \eta_m \geq 0 \quad \forall m.
\end{aligned} \tag{3.2}$$

The inner optimization model $J_t(\boldsymbol{\eta})$ for each task is:

$$\begin{aligned}
\text{minimize} \quad & -\sum_{i=1}^{N_t} y_{ti}(\alpha_{ti}^{+} - \alpha_{ti}^{-}) + \epsilon \sum_{i=1}^{N_t}(\alpha_{ti}^{+} + \alpha_{ti}^{-}) \\
& + \frac{1}{2}\sum_{i=1}^{N_t}\sum_{j=1}^{N_t}(\alpha_{ti}^{+} - \alpha_{ti}^{-})(\alpha_{tj}^{+} - \alpha_{tj}^{-}) \sum_{m=1}^{P} \eta_m k_m(\boldsymbol{x}_{ti}, \boldsymbol{x}_{tj}) \\
\text{with respect to} \quad & \boldsymbol{\alpha}_t^{+} \in \mathbb{R}^{N_t}, \;\; \boldsymbol{\alpha}_t^{-} \in \mathbb{R}^{N_t} \\
\text{subject to} \quad & \sum_{i=1}^{N_t}(\alpha_{ti}^{+} - \alpha_{ti}^{-}) = 0 \\
& C \geq \alpha_{ti}^{+} \geq 0 \quad \forall i \\
& C(1 - \delta_{ti}) \geq \alpha_{ti}^{-} \geq 0 \quad \forall i.
\end{aligned} \tag{3.3}$$

As proposed in Section 2.4, we followed an alternating optimization approach since the overall optimization problem 3.2 is not jointly convex with respect to decision variables $\{\boldsymbol{\alpha}_t^{+}, \boldsymbol{\alpha}_t^{-}\}_{t=1}^{T}$ and $\boldsymbol{\eta}$.

3.1 Solution Methodology

The closed-form update equation for kernel weights is developed by showing the connection between multitask multiple kernel learning and group Lasso as proposed by Xu et al. [2010].

Let us reformulate the optimization model 3.1 as follows:

$$\begin{aligned}
\text{minimize} \quad & \sum_{t=1}^{T}\left[\frac{1}{2}\sum_{m=1}^{P}\eta_m \boldsymbol{w}_{tm}^{\top}\boldsymbol{w}_{tm} + C\sum_{i=1}^{N_t}(\xi_{ti}^{+} + (1-\delta_{ti})\xi_{ti}^{-})\right] \\
\text{with respect to} \quad & \{\boldsymbol{w}_{tm} \in \mathbb{R}^{D_{tm}}\}_{t=1,m=1}^{T,P}, \;\; \{\boldsymbol{\xi}_t^{+} \in \mathbb{R}^{N_t}\}_{t=1}^{T}, \;\; \{\boldsymbol{\xi}_t^{-} \in \mathbb{R}^{N_t}\}_{t=1}^{T}, \;\; \{b_t \in \mathbb{R}\}_{t=1}^{T} \\
\text{subject to} \quad & \epsilon + \xi_{ti}^{+} \geq y_{ti} - \sum_{m=1}^{P}\eta_m \boldsymbol{w}_{tm}^{\top}\boldsymbol{x}_{ti} - b_t \quad \forall(t,i) \\
& \epsilon + \xi_{ti}^{-} \geq \sum_{m=1}^{P}\eta_m \boldsymbol{w}_{tm}^{\top}\boldsymbol{x}_{ti} + b_t - y_{ti} \quad \forall(t,i) \\
& \xi_{ti}^{+} \geq 0 \quad \forall(t,i) \\
& \xi_{ti}^{-} \geq 0 \quad \forall(t,i),
\end{aligned} \tag{3.4}$$

where $\boldsymbol{w}_{tm}$ is weighted by η_m and $\boldsymbol{\eta}$ lies on a simplex, i.e., $\{\boldsymbol{\eta} \in \mathbb{R}^P : \mathbf{1}^\top \boldsymbol{\eta} = 1, \boldsymbol{\eta} \geq 0\}$. The Lagrangian function of the primal model 3.4 is as follows:

$$\mathcal{L} = \sum_{t=1}^{T} \left(\frac{1}{2} \sum_{m=1}^{P} \eta_m \boldsymbol{w}_{tm}^\top \boldsymbol{w}_{tm} + C \sum_{i=1}^{N_t} (\xi_{ti}^+ + (1-\delta_{ti})\xi_{ti}^-) - \sum_{i=1}^{N_t} \beta_{ti}^+ \xi_{ti}^+ - \sum_{i=1}^{N_t} \beta_{ti}^- \xi_{ti}^- \right.$$
$$\left. - \sum_{i=1}^{N_t} \alpha_{ti}^+ (\epsilon + \xi_{ti}^+ - y_{ti} + \sum_{m=1}^{P} \eta_m \boldsymbol{w}_{tm}^\top \boldsymbol{x}_{ti} + b_t) - \sum_{i=1}^{N_t} \alpha_{ti}^- (\epsilon + \xi_{ti}^- - \sum_{m=1}^{P} \eta_m \boldsymbol{w}_{tm}^\top \boldsymbol{x}_{ti} - b_t + y_{ti}) \right),$$

and the partial derivatives with respect to the primal decision variables are:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_{tm}} = 0 \Rightarrow \boldsymbol{w}_{tm} &= \sum_{i=1}^{N_t} (\alpha_{ti}^+ - \alpha_{ti}^-) \boldsymbol{x}_{ti} \qquad \forall (t,m) \\
\frac{\partial \mathcal{L}}{\partial b_t} = 0 \Rightarrow \sum_{i=1}^{N_t} (\alpha_{ti}^+ - \alpha_{ti}^-) &= 0 \qquad \forall t \\
\frac{\partial \mathcal{L}}{\partial \xi_{ti}^+} = 0 \Rightarrow C &= \alpha_{ti}^+ + \beta_{ti}^+ \qquad \forall (t,i) \\
\frac{\partial \mathcal{L}}{\partial \xi_{ti}^-} = 0 \Rightarrow C(1-\delta_{ti}) &= \alpha_{ti}^- + \beta_{ti}^- \qquad \forall (t,i).
\end{aligned}$$

We obtain the corresponding dual formulation when we plug these partial derivatives back into the Lagrangian function;

$$\begin{aligned}
\text{minimize} \quad & \sum_{t=1}^{T} \left[- \sum_{i=1}^{N_t} y_{ti} (\alpha_{ti}^+ - \alpha_{ti}^-) + \epsilon \sum_{i=1}^{N_t} (\alpha_{ti}^+ + \alpha_{ti}^-) \right. \\
& \left. + \frac{1}{2} \sum_{i=1}^{N_t} \sum_{j=1}^{N} (\alpha_{ti}^+ - \alpha_{ti}^-)(\alpha_{tj}^+ - \alpha_{tj}^-) \sum_{m=1}^{P} \eta_m k_m(\boldsymbol{x}_{ti}^\top \boldsymbol{x}_{tj}) \right] \\
\text{with respect to} \quad & \{\boldsymbol{\alpha}_t^+ \in \mathbb{R}^{N_t}\}_{t=1}^T, \ \{\boldsymbol{\alpha}_t^- \in \mathbb{R}^{N_t}\}_{t=1}^T \qquad (3.5) \\
\text{subject to} \quad & \sum_{i=1}^{N_t} (\alpha_{ti}^+ - \alpha_{ti}^-) = 0 \qquad \forall t \\
& C \geq \alpha_{ti}^+ \geq 0 \qquad \forall (t,i) \\
& C(1-\delta_{ti}) \geq \alpha_i^- \geq 0 \qquad \forall (t,i),
\end{aligned}$$

where we can observe that the dual formulation 3.5 is equivalent to the optimization problem 3.3. Then, the closed-form update equation for the kernel weights can be obtained as follows. We define $\widetilde{\boldsymbol{w}}_{tm} = \eta_m \boldsymbol{w}_{tm}$ and rewrite the optimization problem

3.4 as:

$$\begin{aligned}
\text{minimize} \quad & \sum_{t=1}^{T}\left[\frac{1}{2}\sum_{m=1}^{P}\frac{1}{\eta_m}\widetilde{\boldsymbol{w}}_{tm}^{\top}\widetilde{\boldsymbol{w}}_{tm} + C\sum_{i=1}^{N_t}(\xi_{ti}^{+} + (1-\delta_{ti})\xi_{ti}^{-})\right] \\
\text{with respect to} \quad & \{\widetilde{\boldsymbol{w}}_{tm} \in \mathbb{R}^{D_{tm}}\}_{t=1,m=1}^{T,P}, \;\; \{\boldsymbol{\xi}_t^{+} \in \mathbb{R}^{N_t}\}_{t=1}^{T}, \;\; \{\boldsymbol{\xi}_t^{-} \in \mathbb{R}^{N_t}\}_{t=1}^{T}, \;\; \{b_t \in \mathbb{R}\}_{t=1}^{T} \\
\text{subject to} \quad & \epsilon + \xi_{ti}^{+} \geq y_{ti} - \sum_{m=1}^{P}\widetilde{\boldsymbol{w}}_{tm}^{\top}\boldsymbol{x}_{ti} - b_t \quad \forall(t,i) \\
& \epsilon + \xi_{ti}^{-} \geq \sum_{m=1}^{P}\widetilde{\boldsymbol{w}}_{tm}^{\top}\boldsymbol{x}_{ti} + b_t - y_{ti} \quad \forall(t,i) \\
& \xi_{ti}^{+} \geq 0 \quad \forall(t,i) \\
& \xi_{ti}^{-} \geq 0 \quad \forall(t,i),
\end{aligned} \tag{3.6}$$

If we minimize 3.6 with respect to $\boldsymbol{\eta}$ following the procedure given in Appendix A, we obtain the following update equation for the kernel weights:

$$\eta_m = \frac{\sqrt{\sum_{t=1}^{T}\widetilde{\boldsymbol{w}}_{tm}^{\top}\widetilde{\boldsymbol{w}}_{tm}}}{\sum_{o=1}^{P}\sqrt{\sum_{t=1}^{T}\widetilde{\boldsymbol{w}}_{to}^{\top}\widetilde{\boldsymbol{w}}_{to}}}, \tag{3.7}$$

and we can calculate $\widetilde{\boldsymbol{w}}_{tm}^{\top}\widetilde{\boldsymbol{w}}_{tm}$ as

$$\widetilde{\boldsymbol{w}}_{tm}^{\top}\widetilde{\boldsymbol{w}}_{tm} = \eta_m^2\sum_{i=1}^{N_t}\sum_{j=1}^{N_t}(\alpha_{ti}^{+} - \alpha_{ti}^{-})(\alpha_{tj}^{+} - \alpha_{tj}^{-})k_m(\boldsymbol{x_{ti}}, \boldsymbol{x_{tj}}). \tag{3.8}$$

Now we can apply the alternating optimization approach proposed in the previous section since the optimization problem 3.6 is jointly convex with respect to $\boldsymbol{\eta}$ and $\widetilde{\boldsymbol{w}}_{tm}$, and the dual formulation 3.5 is convex with respect to $\{\boldsymbol{\alpha}_t^{+}, \boldsymbol{\alpha}_t^{-}\}$ variables, which guarantees the convergence of the algorithm. We first initialize the kernel weights to uniform values, i.e., $\eta_m^{(0)} = 1/P$. At each iteration s, we solve the inner optimization problem 3.3 for each task by fixing $\boldsymbol{\eta}^{(s)}$ values to obtain its corresponding support vector coefficients $\{\boldsymbol{\alpha}_t^{+(s)}, \boldsymbol{\alpha}_t^{-(s)}\}$. Kernel weights are then updated for the next iteration $(s+1)$ using the following update equation which includes the

support vector coefficients of all tasks:

$$\eta_m^{(s+1)} = \frac{\eta_m^{(t)} \sqrt{\sum_{t=1}^{T} \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} (\alpha_{ti}^{+(s)} - \alpha_{ti}^{-(s)})(\alpha_{tj}^{+(s)} - \alpha_{tj}^{-(s)}) k_m(\boldsymbol{x_{ti}}, \boldsymbol{x_{tj}})}}{\sum_{o=1}^{P} \eta_o^{(s)} \sqrt{\sum_{t=1}^{T} \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} (\alpha_{ti}^{+(s)} - \alpha_{ti}^{-(s)})(\alpha_{tj}^{+(s)} - \alpha_{tj}^{-(s)}) k_o(\boldsymbol{x_{ti}}, \boldsymbol{x_{tj}})}}, \tag{3.9}$$

which can be obtained by plugging Equation 3.8 back into Equation 3.7. We repeat this procedure until the convergence. At the end, we obtain a sparse set of $\boldsymbol{\eta}$ values that are shared by all cancer cohorts. The uninformative pathways/gene sets are eliminated from the model by assigning zero to their corresponding kernel weights. The non-zero kernel weights show us which pathways/gene sets are included in the final model and informative during survival prediction of cancer patients.

Chapter 4

A CLUSTERING ALGORITHM FOR SURVIVAL ANALYSIS

Let us reconsider our assumption that we made for multitask multiple kernel learning algorithm proposed in Chapter 3. We assume that all cancer types included in our Path2MSurv algorithm are related, and we force all tasks to use the same pathways/gene sets for the survival prediction. If there are cancer groups that have different underlying biological mechanisms, our assumption becomes invalid. Forcing all tasks to use the same pathways/gene sets might not be meaningful, and it would not be possible to obtain better predictive performances than the single-task version of our multitask multiple kernel learning algorithm.

Rather than performing multitask learning on all datasets by forcing them to use the same kernel weights, identification of cancer clusters and then applying multitask multiple kernel learning within each cluster would increase the robustness of our algorithm. Instead of assuming all tasks are related, we can assume that cancers within the same group are related. However, we cannot make the same assumption between the cancer groups, which means that pathways/gene sets used for survival prediction might be similar, or completely different for the distinct cancer groups. Therefore, we extend our Path2MSurv algorithm towards task clustering with a unified formulation for clustering of multiple cancer types, learning common underlying biological mechanisms for cancer types within each cluster, and survival analysis model for each cancer dataset. Figure 4.1 shows the summary of our clustering algorithm (named as Path2CSurv).

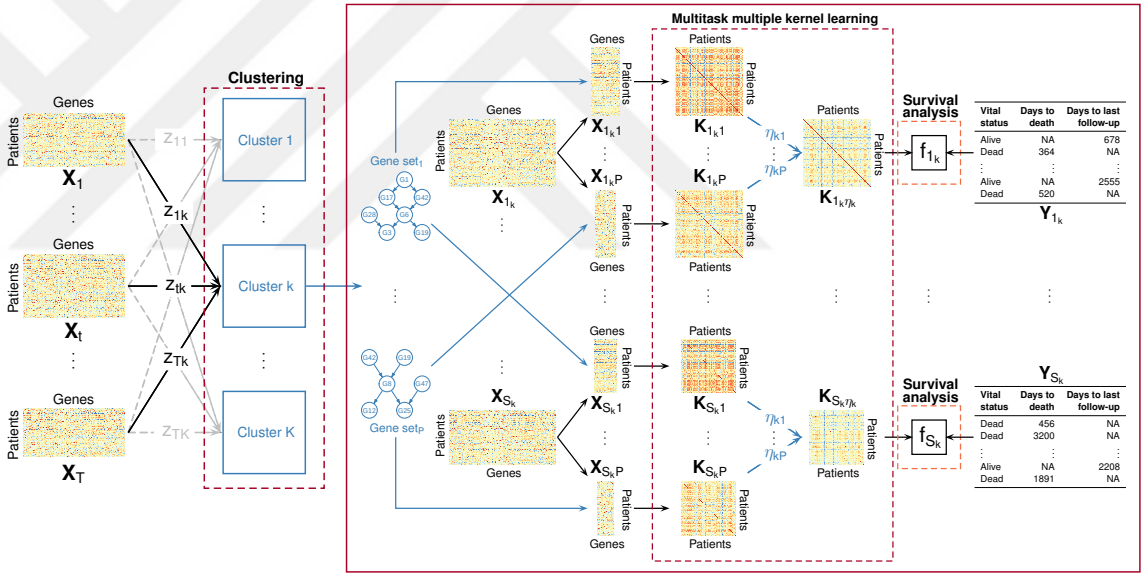

Figure 4.1: The overview figure of Path2CSurv algorithm developed for clustering-based multitask multiple kernel learning algorithm on survival analysis. Gene expression profiles of cancer patients for each dataset (denoted as $\{\mathbf{X}_t\}_{t=1}^{T}$), a gene set collection with P gene sets and clinical information of cancer patients, which includes vital status, days to death, and days to last follow-up (denoted as $\{\mathbf{Y}_t\}_{t=1}^{T}$), are taken as inputs by Path2CSurv algorithm. Then, cluster assignments for each task (i.e. $\{z_{tk}\}_{t=1,k=1}^{T,K} \in \{0,1\}$) are defined using a heuristic method. Kernel matrices (i.e. $\{\mathbf{K}_{s,p}\}_{s=1_k,p=1}^{S_k,P}$, where $\{1_k,\ldots,S_k\}$ is the set of tasks within cluster k and $\bigcup_{k=1}^{K}\{1_k,\ldots,S_k\} = \{1,\ldots,T\}$) are calculated by mapping gene sets on matrix slices derived from each gene expression matrix (denoted as $\{\mathbf{X}_{s,p}\}_{s=1_k,p=1}^{S_k,P}$). Using the kernel matrices within each cluster and the cluster specific kernel weights (i.e. η_{km}), weighted sum of these kernel matrices are calculated. The weighted sum of kernel matrices (denoted as $\{\mathbf{K}_{s,\eta_k}\}_{s=1_k}^{S_k}$) is used to predict the survival time of cancer patients by the prediction functions (i.e. $\{f_s\}_{s=1_k}^{S_k}$).

To be able to cluster cancer datasets, we modify the dual optimization model 3.2 as follows:

$$\begin{aligned}
\text{minimize} \quad & \sum_{t=1}^{T} J_t\left(\sum_{k=1}^{K} z_{tk}\boldsymbol{\eta}_k\right) \\
\text{with respect to} \quad & \{\boldsymbol{\eta}_k \in \mathbb{R}_+^P\}_{k=1}^K, \;\; \mathbf{Z} \in \{0,1\}^{T\times K} \\
\text{subject to} \quad & \sum_{m=1}^{P} \eta_{km} = 1 \quad \forall k \\
& \sum_{k=1}^{K} z_{tk} = 1 \quad \forall t \\
& \sum_{t=1}^{T} z_{tk} \geq 1 \quad \forall k,
\end{aligned} \tag{4.1}$$

where z_{tk} is the binary variable (i.e. $z_{tk} \in \{0,1\}$) that shows whether task t belongs to cluster k or not. We also define different set of kernel weights for each cluster and η_{km} represents the weight of kernel m that belongs to cluster k. The inner optimization model $J_t\left(\sum_{k=1}^{K} z_{tk}\boldsymbol{\eta}_k\right)$ can be formulated as follows:

$$\begin{aligned}
\text{minimize} \quad & -\sum_{i=1}^{N_t} y_{ti}(\alpha_{ti}^+ - \alpha_{ti}^-) + \epsilon\sum_{i=1}^{N_t}(\alpha_{ti}^+ + \alpha_{ti}^-) \\
& + \frac{1}{2}\sum_{i=1}^{N_t}\sum_{j=1}^{N_t}(\alpha_{ti}^+ - \alpha_{ti}^-)(\alpha_{tj}^+ - \alpha_{tj}^-)\sum_{k=1}^{K}\sum_{m=1}^{P} z_{tk}\eta_{km}k_m(\boldsymbol{x}_{ti}, \boldsymbol{x}_{tj}) \\
\text{with respect to} \quad & \boldsymbol{\alpha}_t^+ \in \mathbb{R}^{N_t}, \;\; \boldsymbol{\alpha}_t^- \in \mathbb{R}^{N_t} \\
\text{subject to} \quad & \sum_{i=1}^{N_t}(\alpha_{ti}^+ - \alpha_{ti}^-) = 0 \\
& C \geq \alpha_{ti}^+ \geq 0 \quad \forall i \\
& C(1-\delta_{ti}) \geq \alpha_{ti}^- \geq 0 \quad \forall i.
\end{aligned} \tag{4.2}$$

It is important to note that the optimization model 4.1 reduces to optimization model 3.2 when the number of clusters are fixed. We will use this information while developing our solution methodology.

The optimization model 4.1 is a highly non-linear non-convex model that cannot be solved globally. We also cannot follow a similar alternating optimization

method as proposed in the previous chapters since the resulted optimization model obtained by showing the relation between model 4.2 and group Lasso would not be jointly convex with respect to z_{tk}, $\boldsymbol{\eta}_k$, and $\widetilde{\boldsymbol{w}}_{tm}$. Therefore, we propose a heuristic approach for our Path2CSurv algorithm by merging the alternating optimization method proposed for Path2MSurv algorithm with simulated annealing algorithm.

4.1 Solution Methodology

Simulated annealing is a local search metaheuristic that is used to find an approximate global solution for the combinatorial optimization problems [Kirkpatrick et al., 1983]. It is a very popular technique due to its convergence properties, ease of implementation, and use of hill-climbing move to escape from the local optima. Simulated annealing is an iterative algorithm. At each iteration, the algorithm compares the current and the newly obtained solution. The new solutions that improve the objective function value are always accepted, while a fraction of the solutions that worsen the objective function value are accepted (i.e. hill-climbing moves). Allowing hill-climbing moves gives simulated annealing algorithm the possibility of escaping from the local optimal solutions in search of global optimal solution. The probability of accepting a non-improving move is determined by using a temperature parameter. The temperature parameter is decreased at each iteration so that the probability of accepting a non-improving move decreases as the temperature decreases. This feature enables the algorithm to converge to a global optimal solution.

The pseudo-code of our algorithm that solves the optimization problem 4.1 is given in Algorithm 1.

Algorithm 1: Path2CSurv – Simulated Annealing

Select initial solutions for $\mathbf{Z}^{(s)}$, $\{\boldsymbol{\eta}_k^{(s)}\}_{k=1}^K$, $\{(\boldsymbol{\alpha}_t^{+(s)}, \boldsymbol{\alpha}_t^{-(s)})\}_{t=1}^T$, $s = 0$
Select a temperature reduction function (β),
Calculate initial overall objective ($obj^{(s)}$, $s = 0$),
Initialize temperature ($temp^{(s)}$, $s = 0$),
Initialize maximum number of iterations per temperature ($nrep$).
while *Not converged* **do**
  repeat
    Randomly select a task t.
    For task t, apply a local search to explore the best possible:
      cluster membership (i.e. $\mathbf{Z}^*$),
      kernel weights (i.e. $\{\boldsymbol{\eta}_k^*\}_{k=1}^K$),
      support vector coefficients (i.e. $\{(\boldsymbol{\alpha}_t^{+*}, \boldsymbol{\alpha}_t^{-*})\}_{t=1}^T$).
    Calculate new overall objective (i.e. obj^*)
    if $obj^* < obj^{(s)}$ **then**
      Update:
        $\mathbf{Z}^{(s+1)} = \mathbf{Z}^*$,
        $\{\boldsymbol{\eta}_k^{(s+1)}\}_{k=1}^K = \{\boldsymbol{\eta}_k^*\}_{k=1}^K$,
        $\{(\boldsymbol{\alpha}_t^{+(s+1)}, \boldsymbol{\alpha}_t^{-(s+1)})\}_{t=1}^T = \{(\boldsymbol{\alpha}_t^{+*}, \boldsymbol{\alpha}_t^{-*})\}_{t=1}^T$
        $obj^{(s+1)} = obj^*$
    else
      n : uniform random number $\in (0, 1)$
      γ : $\exp(-(obj^* - obj^{(s)})/temp^{(s)})$
      if $n < \gamma$ **then**
        Update:
          $\mathbf{Z}^{(s+1)} = \mathbf{Z}^*$,
          $\{\boldsymbol{\eta}_k^{(s+1)}\}_{k=1}^K = \{\boldsymbol{\eta}_k^*\}_{k=1}^K$,
          $\{(\boldsymbol{\alpha}_t^{+(s+1)}, \boldsymbol{\alpha}_t^{-(s+1)})\}_{t=1}^T = \{(\boldsymbol{\alpha}_t^{+*}, \boldsymbol{\alpha}_t^{-*})\}_{t=1}^T$
          $obj^{(s+1)} = obj^*$
      else
        No update
  until *nrep is reached*;
  Update temperature:
    $temp^{(s+1)} = \beta(temp^{(s)})$
  if *Converged* **then**
    Stop
  else
    $s = s + 1$
Output: $\mathbf{Z}$, $\{\boldsymbol{\eta}_k\}_{k=1}^K$, $\{(\boldsymbol{\alpha}_t^+, \boldsymbol{\alpha}_t^-)\}_{t=1}^T$

Our algorithm consists of the following steps:

Initialization Step

As the first step, our algorithm starts by initializing the problem variables and simulated annealing parameters. Assuming that there are K number of clusters and T number of tasks, we first set kernel weights of each cluster to uniform values (i.e. $\eta_{km}^{(0)} = 1/P$). Then, we randomly select $\mathbf{Z}^{(0)}$ values that represent which task belongs to which cluster. $\mathbf{Z}$ is a $T \times K$ matrix where rows represent cancer datasets and columns represent clusters. The feasibility of $\mathbf{Z}^{(0)}$ must be satisfied by assigning each cancer to exactly one cluster and assigning at least one cancer to each cluster. We solve the inner optimization problem 4.2 with the current $\{\boldsymbol{\eta}_k^{(0)}\}_{k=1}^{K}$ and $\mathbf{Z}^{(0)}$ values to obtain initial support vector coefficients $(\boldsymbol{\alpha}_t^{+(0)}, \boldsymbol{\alpha}_t^{-(0)})$.

Selection of the temperature reduction function is crucial for the success of simulated annealing algorithm. The probability of accepting a non-improving move (i.e. hill-climbing move) is calculated using the temperature parameter. The convergence property of simulated annealing algorithm is achieved by gradually decreasing the acceptance probability of hill-climbing moves. This decrease in the acceptance probability is controlled by the temperature parameter within a reduction function. The temperature reduction function should be selected so that the temperature decreases gradually and converges to zero, which also decreases the acceptance probability of hill-climbing moves gradually. The temperature reduction function also controls the convergence of the simulated annealing. We select the following temperature reduction function accordingly:

$$\text{Temp}^{(s+1)} = \text{Temp}^{(0)} \left[\frac{\text{Temp}^{(|S|)}}{\text{Temp}^{(0)}} \right]^{\frac{s}{|S|}}, \tag{4.3}$$

where s is the current iteration count, $|S|$ represents the maximum number of outer iterations (i.e. how many times the temperature will be reduced), $nrep$ is the number of iterations performed at each temperature, $\text{Temp}^{(0)}$ is the initial temperature, and $\text{Temp}^{(|S|)}$ is the final temperature. All these parameters affect how fast the

temperature is reduced. These parameters should be tuned so that the time spent at lower temperatures is large enough to seek near-optimal solutions.

Search Step

The initialization step is followed by exploring the solution space to find a candidate solution set. At each temperature, a local search is applied for a randomly selected task t. The obtained candidate solution set for $\mathbf{Z}$, $\{\boldsymbol{\eta}_k\}_{k=1}^{K}$, and $\{(\boldsymbol{\alpha}_t^+, \boldsymbol{\alpha}_t^-)\}_{t=1}^{T}$ is accepted as the new solution set if the current objective function value of the overall problem 4.1 is improved. If not, the candidate solution set is called as a hill-climbing move (non-improving move). This non-improving move is accepted as the new solution set if a randomly selected uniform number between zero and one is less than the probability (γ) that is calculated using the current temperature. Otherwise, the algorithm continues without any update. This procedure is repeated for *nrep* iterations.

Temperature Reduction and Convergence Check Step

After the local search and the acceptance control steps are repeated *nrep* times, the temperature is updated using the selected temperature reduction function. The search and the temperature reduction steps are repeated until the convergence criterion is satisfied. In our case, the convergence criterion is that either the current temperature is lower than the pre-defined final temperature or the maximum number for the outer iterations is reached.

After the algorithm stops: (i) the overall objective function value converges to a stationary level, (ii) kernel weights converge to a sparse set of solutions which show the informative pathways/gene sets for the given cluster, and (iii) final $\mathbf{Z}$ values show which cohorts share the same underlying biological mechanisms.

Local search

Local search applied in the search step of our algorithm starts by selecting a task randomly. Assume that selected task t belongs to cluster k currently. After selecting task t, candidate $\mathbf{Z}$ matrices are created for every cluster that task t can be assigned to, including cluster k. In other words, assuming that there are K number of clusters, K different $\mathbf{Z}$ matrices, which show every possible cluster assignment for task t, are created. The feasibility of each candidate $\mathbf{Z}$ matrix should be checked. A cluster assignment matrix is feasible only if: (i) a task belongs to exactly one cluster, and (ii) each cluster has at least one task assigned to it. If task t is the only task in cluster k, then $\mathbf{Z}$ matrix becomes infeasible when we assign task t to a cluster other than cluster k. In such cases, we first determine the clusters that have at least two tasks assigned to them. Then, we randomly select a cluster from this cluster set, namely k^*, and, in cluster k^*, we select the task (namely task t^*) with the highest coefficient in the overall objective function in optimization problem 4.1. The feasibility of the candidate $\mathbf{Z}$ matrix is then achieved by assigning task t^* to cluster k^*.

After ensuring the feasibility of the candidate $\mathbf{Z}$ matrix, $\{\boldsymbol{\eta}_k\}_{k=1}^{K}$, and $\{(\boldsymbol{\alpha}_t^+, \boldsymbol{\alpha}_t^-)\}_{t=1}^{T}$ variables are updated. Since the overall problem 4.1 is equivalent to optimization problem 3.2 when the $\mathbf{Z}$ values are given, we can solve model 4.1 with fixed $\mathbf{Z}$ by following the same optimization approach proposed for our multitask multiple kernel learning algorithm that we update the kernel weights and the support vector variables. We can modify the closed-form kernel update function as follows:

$$\eta_{km}^{(s+1)} = \frac{\eta_{km}^{(s)} \sqrt{\sum_{t=1}^{T} z_{tk} \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} (\alpha_{ti}^{+(s)} - \alpha_{ti}^{-(s)})(\alpha_{tj}^{+(s)} - \alpha_{tj}^{-(s)}) k_m(\boldsymbol{x}_{ti}, \boldsymbol{x}_{tj})}}{\sum_{o=1}^{P} \eta_{ko}^{(s)} \sqrt{\sum_{t=1}^{T} z_{to} \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} (\alpha_{ti}^{+(s)} - \alpha_{ti}^{-(s)})(\alpha_{tj}^{+(s)} - \alpha_{tj}^{-(s)}) k_o(\boldsymbol{x}_{ti}, \boldsymbol{x}_{tj})}} \quad \forall (k, m).$$

Similarly, when the kernel weights are given with the cluster assignment variables $\mathbf{Z}$, the overall model is equivalent to standard SVM model [Khan and Zubek, 2008; Shivaswamy et al., 2007] and can be solved using quadratic optimization solvers.

Chapter 5

RESULTS

We extensively test our algorithms on 20 cancer datasets obtained from The Cancer Genome Atlas consortium. In this chapter, we compare the predictive performances of our algorithms and two baseline algorithms, namely survival RF [Breiman, 2001] and survival SVM [Khan and Zubek, 2008; Shivaswamy et al., 2007], on each dataset. We report the informative biological mechanisms for survival prediction of cancer patients obtained by our algorithms. We also report the cluster structures for these 20 datasets obtained by our clustering algorithm.

In this chapter, we give the details for the datasets that we used in our study, the experimental settings, the performance measure that we used to compare the predictive performances of baseline algorithms and our proposed algorithms, and experimental results.

5.1 Datasets

We use gene expression profiles of tumours biopsied from the cancer patients and their clinical annotation data, provided by The Cancer Genome Atlas (TCGA) at the Genomics Data Commons (GDC) data portal (`https://portal.gdc.cancer.gov`). We also use two cancer-specific pathway/gene set collections to identify informative biological mechanisms for predicting overall survival time of cancer patients.

5.1.1 TCGA Datasets

TCGA provides gene expression profiles and clinical annotation files of more than 10000 cancer patients for 33 cancer datasets. We use gene expression profiles of tumours which were generated by the `RNA-Seq` analysis pipeline of TCGA consortium. `HTSeq-FPKM` files of all primary tumours from the most recent data freeze (i.e.

`Data Release 16.0-March 26, 2019`) are downloaded for each cancer type, which leads to 9911 files in total. We exclude the metastatic tumours from our study since their underlying biological mechanisms might be different than the primary tumours. `Clinical Supplement` files of all patients for each cancer type to extract the survival characteristics of cancer patients are downloaded, which leads to 10604 files in total. These files show days to last follow-up or days to last known alive for alive patients and days to death for dead patients.

We apply a filtering process to be able to use gene expression profiles and survival characteristics of cancer patients while performing survival analysis. We include only the patients whose both gene expression profiles and survival characteristics are available. We discard the patients whose `vital_status` is `Dead` and `days_to_death` is non-positive or `NA`. We also discard the patients whose `vital_status` is `Alive` and `days_to_last_followup` is non-positive or `NA`. After these filtering steps, we obtain a data collection including 9621 patients and their corresponding gene expression profiles in 33 cancer types. We also do not include cohorts that have less than 20 patients with `vital_status` as `Dead` and at least 100 patients in total. By doing so, we aim to guarantee that each dataset has a robust performance measure. Finally, we pick 20 cancer datasets including 7655 patients in total to analyze in our experiments. The following cancer types are included in our experiments: bladder urothelial carcinoma (`BLCA`), breast invasive carcinoma (`BRCA`), cervical squamous cell carcinoma and endocervical adenocarcinoma (`CESC`), colon adenocarcinoma (`COAD`), esophageal carcinoma (`ESCA`), glioblastoma multiforme (`GBM`), head and neck squamous cell carcinoma (`HNSC`), kidney renal clear cell carcinoma (`KIRC`), kidney renal papillary cell carcinoma(`KIRP`), acute myeloid leukemia (`LAML`), brain lower grade glioma (`LGG`), liver hepatocellular carcinoma (`LIHC`), lung adenocarcinoma (`LUAD`), lung squamous cell carcinoma (`LUSC`), ovarian serous cystadenocarcinoma (`OV`), pancreatic adenocarcinoma (`PAAD`), rectum adenocarcinoma (`READ`), sarcoma (`SARC`), stomach adenocarcinoma (`STAD`), and uterine corpus endometrial carcinoma (`UCEC`). Table 5.1 shows the details about 33 datasets that we constructed.

Table 5.1: Information about 33 cancer datasets obtained from the Cancer Genome Atlas.

Cohort	Disease name	Number of primary tumors with available mRNA profiles	Number of patients with available survival data	Number of samples with both data sources	Number of patients with censored survival data after filtering	Number of patients with uncensored survival data after filtering	Included in the further analyses
ACC	Adrenocortical carcinoma	79	91	79	51	28	No
BLCA	Bladder urothelial carcinoma	414	406	402	226	176	Yes
BRCA	Breast invasive carcinoma	1102	1074	1067	918	149	Yes
CESC	Cervical squamous cell carcinoma and endocervical adenocarcinoma	304	294	291	220	71	Yes
CHOL	Cholangiocarcinoma	36	48	36	18	18	No
COAD	Colon adenocarcinoma	478	437	433	338	95	Yes
DLBC	Lymphoid neoplasm diffuse large B-cell lymphoma	48	47	47	38	9	No
ESCA	Esophageal carcinoma	161	184	160	97	63	Yes
GBM	Glioblastoma multiforme	156	592	152	30	122	Yes

Cohort	Disease name	Number of primary tumors with available mRNA profiles	Number of patients with available survival data	Number of samples with both data sources	Number of patients with censored survival data after filtering	Number of patients with uncensored survival data after filtering	Included in the further analyses
HNSC	Head and neck squamous cell carcinoma	500	526	498	281	217	Yes
KICH	Kidney chromophobe	65	112	64	55	9	No
KIRC	Kidney renal clear cell carcinoma	538	533	526	355	171	Yes
KIRP	Kidney renal papillary cell carcinoma	288	288	285	241	44	Yes
LAML	Acute myeloid leukemia	151	173	130	52	78	Yes
LGG	Brain lower grade glioma	511	511	506	381	125	Yes
LIHC	Liver hepatocellular carcinoma	371	371	365	235	130	Yes
LUAD	Lung adenocarcinoma	533	509	500	318	182	Yes
LUSC	Lung squamous cell carcinoma	502	496	493	282	211	Yes
MESO	Mesothelioma	86	85	84	12	72	No
OV	Ovarian serous cystadenocarcinoma	374	582	372	143	229	Yes

Cohort	Disease name	Number of primary tumors with available mRNA profiles	Number of patients with available survival data	Number of samples with both data sources	Number of patients with censored survival data after filtering	Number of patients with uncensored survival data after filtering	Included in the further analyses
PAAD	Pancreatic adenocarcinoma	177	184	176	84	92	Yes
PCPG	Pheochromocytoma and paraganglioma	178	178	177	171	6	No
PRAD	Prostate adenocarcinoma	498	499	494	484	10	No
READ	Rectum adenocarcinoma	166	161	156	131	25	Yes
SARC	Sarcoma	259	258	256	158	98	Yes
SKCM	Skin cutaneous melanoma	103	456	98	70	28	No
STAD	Stomach adenocarcinoma	375	412	348	206	142	Yes
TGCT	Testicular germ cell tumors	150	134	134	130	4	No
THCA	Thyroid carcinoma	502	506	501	485	16	No
THYM	Thymoma	119	123	118	109	9	No
UCEC	Uterine corpus endometrial carcinoma	551	544	539	449	90	Yes
UCS	Uterine carcinosarcoma	56	55	54	21	33	No
UVM	Uveal melanoma	80	80	80	57	23	No

5.1.2 Pathway/Gene Set Collections

In addition to predicting overall survival time of cancer patients using their gene expression profiles and survival characteristics, we also want to identify biological mechanisms that are relevant for this prediction process. To this end, we use a pathway and a gene set collection described in the literature. The Molecular Signatures Database (MSigDB) provides pathway/gene set databases that give information about groups of genes and their functional similarities [Subramanian et al., 2005]. We extract `Hallmark` gene sets and `Pathway Interaction Database` (`PID`) pathways from the MSigDB, which can be publicly accessed at `http://software.broadinstitute.org/gsea/msigdb`. Both of these collections were curated specifically for cancer research. `Hallmark` is a computationally curated gene set collection [Liberzon et al., 2015]. Each gene set included in this collection conveys specific biological state or process and displays coherent expression. The `Hallmark` collection includes 50 gene sets and the number of genes included in each gene set varies between 32-200. `PID` is a manually curated and peer-reviewed pathway collection that includes human molecular signalling and regulatory events and key cellular processes [Schaefer et al., 2009]. The `PID` collection is composed of 196 pathways, and the number of genes included in each pathway varies between 10-137.

5.2 Experimental Settings

For each dataset, we split the data points into two parts by randomly picking the 80% of them as training and 20% as test sets. While partitioning the data as training and test sets, we try to keep the ratio between the number of patients with `vital_status` as `Dead` and the number of patients with `vital_status` as `Alive` in the training and test sets equal as much as possible. We first apply $\log_2$-transformation to the gene expression profiles of primary tumours since they are count data. We normalize the training partition to have zero mean and unit standard deviation, whereas we normalize the test set using the mean and standard deviation that the training partition had before the normalization.

We select the hyper-parameters of each learning algorithm and the parameters of simulated annealing algorithm as follows: The number of trees generated by survival RF and the regularization parameter for kernel-based algorithms are selected using a four-fold inner cross-validation method for each training set, from the sets $\{500, 1000, \ldots, 2500\}$ and $\{10^{-4}, 10^{-3}, \ldots, 10^{5}\}$, respectively. Survival SVM, and our Path2Surv and Path2MSurv algorithms are performed 200 iterations to guarantee the convergence of these algorithms. For our Path2CSurv algorithm, the maximum number of iterations per temperature (i.e. *nrep*), the temperature reduction function (i.e. β), and the maximum number of outer iterations (i.e. how many times the temperature is updated) are specified using a full factorial experiment. We choose *nrep* as 10, the maximum outer iteration as 150 and use the given reduction function 4.3, since the objective function value starts to converge around 1000 iterations and almost half of the total number of cohorts can be spanned with 10 inner iterations per temperature. We set the initial temperature to the initial overall objective function value, so that the acceptance probability of a non-improving solution is initialized to one.

We replicate above mentioned procedures 100 times for each algorithm to obtain more robust performance measures and report the results of these 100 replications at the end. We replicate our clustering algorithm (i.e. Path2CSurv) 100 times for six different cluster counts (i.e. $\{3, 4, \ldots, 8\}$ clusters), since we do not know how many clusters exists among the included cancer cohorts.

For each kernel-based algorithm, the similarity between the gene expression profiles of primary tumours within each cohort is calculated using the following Gaussian kernel:

$$k_{\mathcal{G}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\big(-(\boldsymbol{x}_i - \boldsymbol{x}_j)^{\top}(\boldsymbol{x}_i - \boldsymbol{x}_j)/(2\sigma^2)\big),$$

where σ represents the kernel width parameter, and it is specified as the mean of pairwise Euclidean distances between training instances. We calculate the kernel matrices by using the subset of gene expression profiles obtained by using the genes included in the corresponding pathway/gene set and the kernel widths are calculated accordingly. The success of Gaussian kernels on capturing non-linear dependencies

between the gene expression profiles and different phenotypes in cancer-related problems were shown in the literature [Costello et al., 2014; Gönen et al., 2017]. We set the tube width parameter ϵ to 0 for each kernel-based algorithm in this study.

For survival RF algorithm, `randomForestSRC` R package version 2.6.0 [Ishwaran and Kogalur, 2018] is used. For survival SVM, Path2Surv, Path2MSurv, and Path2CSurv, we use our own implementations. CPLEX version 12.7.1 is used to solve quadratic optimization problems in SVM-based algorithms [IBM, 2017].

5.3 Performance Measures

We use the concordance index (C-index) as a performance measure. It can be used to evaluate predictive performances of machine learning algorithms applied on censored data. C-index shows the probability of having concordant observed and predicted survival times for a randomly chosen comparable pair. A pair is considered comparable if the pair consists of either i) two patients with `vital_status` as `Dead`, or ii) one patient with `vital_status` as `Dead` and another patient with `vital_status` as `Alive`, where `Alive` patient's `days_to_last_followup` is greater than `Dead` patient's `days_to_death`. A comparable pair is concordant if we can rank their predicted survival times as their observed survival times. We use the following formulation to calculate the C-index values for a given cancer dataset.

$$\text{C-index} = \frac{\sum_{i=1}^{N} \sum_{j \neq i} \Delta_{ij} 1((y_i - y_j)(\hat{y}_i - \hat{y}_j) > 0)}{\sum_{i=1}^{N} \sum_{j \neq i} \Delta_{ij}},$$

where $\hat{y}_i$ is the predicted survival time of patient i and

$$\Delta_{ij} = \begin{cases} 1, & (\delta_i = 0, \delta_j = 0) \text{ or } (\delta_i = 0, \delta_j = 1, y_i < y_j), \\ 0, & \text{otherwise.} \end{cases}$$

5.4 Experimental Results

5.4.1 Predictive Performance Comparisons

In this section, we compare the predictive performances of survival RF (denoted as `RF`), survival SVM (denoted as `SVM`), our Path2Surv algorithm (denoted as `MKL`), our Path2MSurv algorithm (denoted as `MTMKL`), and our Path2CSurv algorithm (denoted as `C#MTMKL` where # represents the number of clusters in the algorithm) on 20 TCGA datasets using their C-index values. We provide all available gene expression features (i.e. 19814 features in total) for `RF` and `SVM` algorithms as the input. We add `[H]` or `[P]` to the algorithm name if `Hallmark` or `PID` pathway/gene set collection is used in the corresponding algorithm. We could not test our Path2CSurv algorithm using `PID` collection due to the computational time limitations on high performance computing clusters that we used for our experiments.

Our multiple kernel-based algorithms obtain statistically significantly better or comparable predictive performances against all benchmark algorithms. Figure 5.1 and Figure 5.2 show the predictive performances of `RF`, `SVM`, our Path2Surv algorithm with `PID` pathway collection (`MKL[P]`) and with `Hallmark` gene set collection (`MKL[H]`), and our Path2MSurv algorithm with `PID` pathway collection (`MTMKL[P]`) and with `Hallmark` gene set collection (`MTMKL[H]`) on 20 cancer datasets for survival analysis of cancer patients using their gene expression profiles. For each dataset, the corresponding violin plot compares the C-index values of the algorithms over 100 replications using a two-tailed paired t-test. We observe that `RF` algorithm outperforms `SVM` algorithm on six out of 20 datasets, while `SVM` algorithm outperforms `RF` on 12 datasets. From these results, we can infer that `SVM` algorithm, which uses a non-linear Gaussian kernel, performs better than `RF` algorithm on capturing the non-linear dependency between gene expression profile and overall survival time in survival analysis problems.

When we use `Hallmark` collection as input data, our Path2Surv algorithm (i.e. `MKL[H]`), which is the extension of `SVM` algorithm towards multiple kernel learning, outperforms `RF` algorithm on 12 datasets (i.e. `BLCA`, `BRCA`, `CESC`, `GBM`, `HNSC`, `LUAD`,

LUSC, OV, PAAD, SARC, STAD, UCEC), whereas RF outperforms our MKL[H] algorithm only on COAD, LAML, and READ datasets. Our Path2Surv algorithm with PID collection, namely MKL[P], outperforms RF algorithm on 10 out of 20 datasets, while it underperforms RF algorithm on five datasets. We observe that SVM algorithm also outperforms RF algorithm on 12 out of 20 datasets, whereas RF outperforms SVM on six datasets. When we compare SVM algorithm with our MKL algorithms with both PID and Hallmark collections, Figure 5.2 shows that our algorithm outperforms SVM algorithm on 4 and 6 datasets, respectively. Similarly, SVM outperforms our algorithms with PID and Hallmark collections on 7 and 5 datasets, respectively. These results show that we cannot mention superiority of one algorithm to another in terms of the predictive performances; however, MKL[P] and MKL[H] algorithms used significantly fewer gene expression features while making survival prediction. The details for the number of genes selected by each algorithm are given in the next section.

Out of 20 datasets we use in our experiments, we observe that RF algorithm obtains median C-index values below 0.50 for GBM and LUSC datasets. In addition to that, SVM algorithm obtains median C-index values below 0.5 on READ dataset. For all datasets, our MKL algorithms' median C-index values are above 0.50. These results support the fact that kernel-based algorithms are more suitable than RF algorithm for survival analysis on cancer datasets in terms of predictive power.

Figure 5.1 and Figure 5.2 indicate that our Path2MSurv algorithms with PID and Hallmark collections, namely MTMKL[P] and MTMKL[H], obtain statistically significantly better or comparable predictive performances against RF and SVM algorithms. Our MTMKL[P] and MTMKL[H] algorithms outperform RF algorithm on 13 out of 20 datasets. On the other hand, RF outperforms MTMKL[P] and MTMKL[H] algorithms on 2 and 1 datasets, respectively. For CESC, GBM, HNSC, LUAD, LUSC, PAAD, and UCEC datasets, our Path2MSurv algorithm outperforms RF algorithm by improving the C-index values more than 4%. Figure 5.2 shows that MTMKL[P] algorithm outperforms SVM on 14 out of 20 datasets, whereas SVM outperforms MTMKL[P] on BRCA and LUSC. When we use Hallmark collection as input data for MTMKL algorithm, we observe

that MTMKL[H] outperforms SVM on 13 datasets, whereas SVM outperforms MTMKL[H] on BLCA, BRCA, and LUSC datasets.

We also compare MKL and MTMKL algorithms in terms of their predictive performances. We observe that MTMKL[P] algorithm outperforms MKL[P] algorithm on 15 out of 20 datasets. For BLCA, CESC, GBM, OV, PAAD, SARC, and STAD datasets, this increase is more than 2%. Similarly, MTMKL[H] algorithm outperforms MKL[H] algorithm on 14 out of 20 datasets, especially with a more than 2% increase in the predictive performances of BLCA, LAML, and UCEC datasets. On the other hand, MKL algorithm outperforms MTMKL algorithm with both PID and Hallmark collections on BRCA and LUSC datasets. These results clearly show the benefit of using multi-task learning approach for survival analysis of cancer patients rather than modeling each cohort separately as in RF, SVM, and the single-task variant of our Path2MSurv algorithm (i.e. MKL).

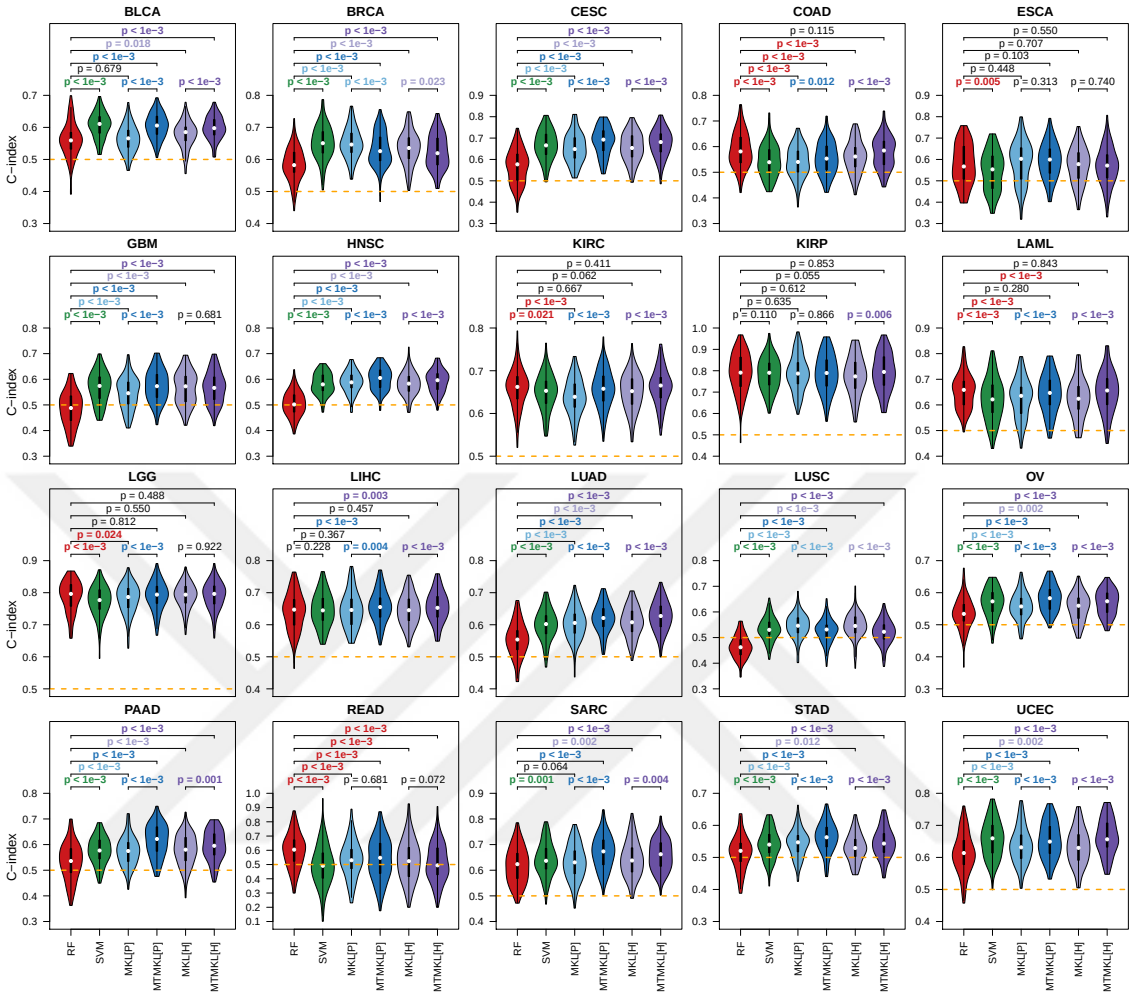

Figure 5.1: The predictive performances of survival RF (`RF`) algorithm, survival SVM (`SVM`) algorithm, single-task MKL algorithm Path2Surv with `PID` pathway collection (`MKL[P]`) and with `Hallmark` gene set collection (`MKL[H]`), multitask MKL algorithm Path2MSurv with `PID` pathway collection (`MTMKL[P]`) and with `Hallmark` gene set collection (`MTMKL[H]`) on 20 cancer datasets. Each violin plot shows C-index values over 100 replications. Two-tailed paired t-tests are used to see whether there are significant differences between pairs of algorithms. For P-value results, **red**: RF is better; **green**: `SVM` is better; **light blue**: `MKL[P]` is better; **dark blue**: `MTMKL[P]` is better; **light magenta**: `MKL[H]` is better; **dark magenta**: `MTMKL[H]` is better; black: no difference. **Orange**: baseline performance level where C-index = 0.5.

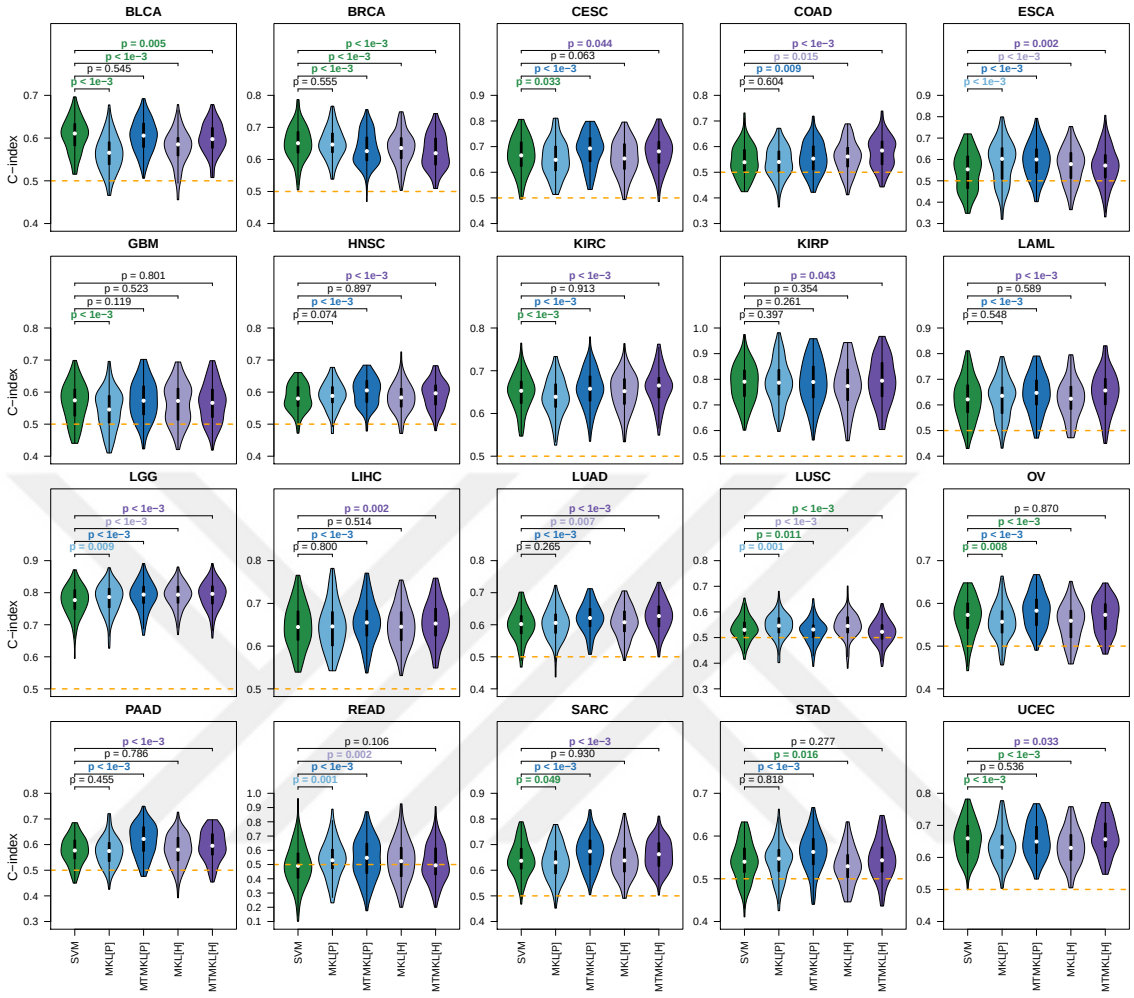

Figure 5.2: The predictive performances of survival SVM (`SVM`) algorithm, single-task MKL algorithm Path2Surv with `PID` pathway collection (`MKL[P]`) and with `Hallmark` gene set collection (`MKL[H]`), multitask MKL algorithm Path2MSurv with `PID` pathway collection (`MTMKL[P]`) and with `Hallmark` gene set collection (`MTMKL[H]`) on 20 cancer datasets. Each violin plot shows C-index values over 100 replications. Two-tailed paired t-tests are used to see whether there are significant differences between pairs of algorithms. For P-value results, **green**: `SVM` is better; **light blue**: `MKL[P]` is better; **dark blue**: `MTMKL[P]` is better; **light magenta**: `MKL[H]` is better; **dark magenta**: `MTMKL[H]` is better; black: no difference. **Orange**: baseline performance level where C-index = 0.5.

We test our clustering algorithm, namely Path2CSurv, with six different cluster counts (i.e. three, four, ..., eight) since we do not know the number of clusters. Also, we test our Path2CSurv algorithm only with `Hallmark` collection due to the computation time limitations on the high performance computing cluster that we run our algorithms. Our Path2CSurv algorithm reduces to MKL algorithm when the number of clusters is equal to the number of cohorts. When six or more clusters are selected, we observe that the number of datasets in which there is no statistical difference between Path2CSurv and MKL algorithms in terms of the number of gene expression features selected starts to increase. This is why we focus on the results obtained for five clusters in more detail. In Appendix C, Figures C.1–C.5 show the predictive performance comparisons of Path2CSurv algorithm with three, four, six, seven, and eight clusters.

In Figure 5.3, we show the predictive performance comparisons of Path2CSurv algorithm against `RF`, `SVM`, `MKL[H]`, and `MTMKL[H]` algorithms when the number of clusters is set to five. Path2CSurv algorithm outperforms RF algorithm on 14 out of 20 datasets, while RF algorithm outperforms Path2CSurv only on `READ` dataset. We observe that Path2CSurv algorithm outperforms `SVM` algorithm on 7 datasets, whereas it underperforms `SVM` on `BLCA`, `BRCA`, and `OV` datasets. When we consider single-task single-cluster variant of Path2CSurv algorithm (i.e. `MKL[H]`), Path2CSurv outperforms `MKL[H]` on 6 out of 20 datasets, while `MKL[H]` outperforms Path2CSurv on only `LUSC` dataset. We notice that Path2CSurv algorithm uses the highest number of gene sets (i.e. more than 11 gene sets on average) for `LUSC` while performing survival analysis, which shows that the survival prediction process is more difficult than the other datasets.

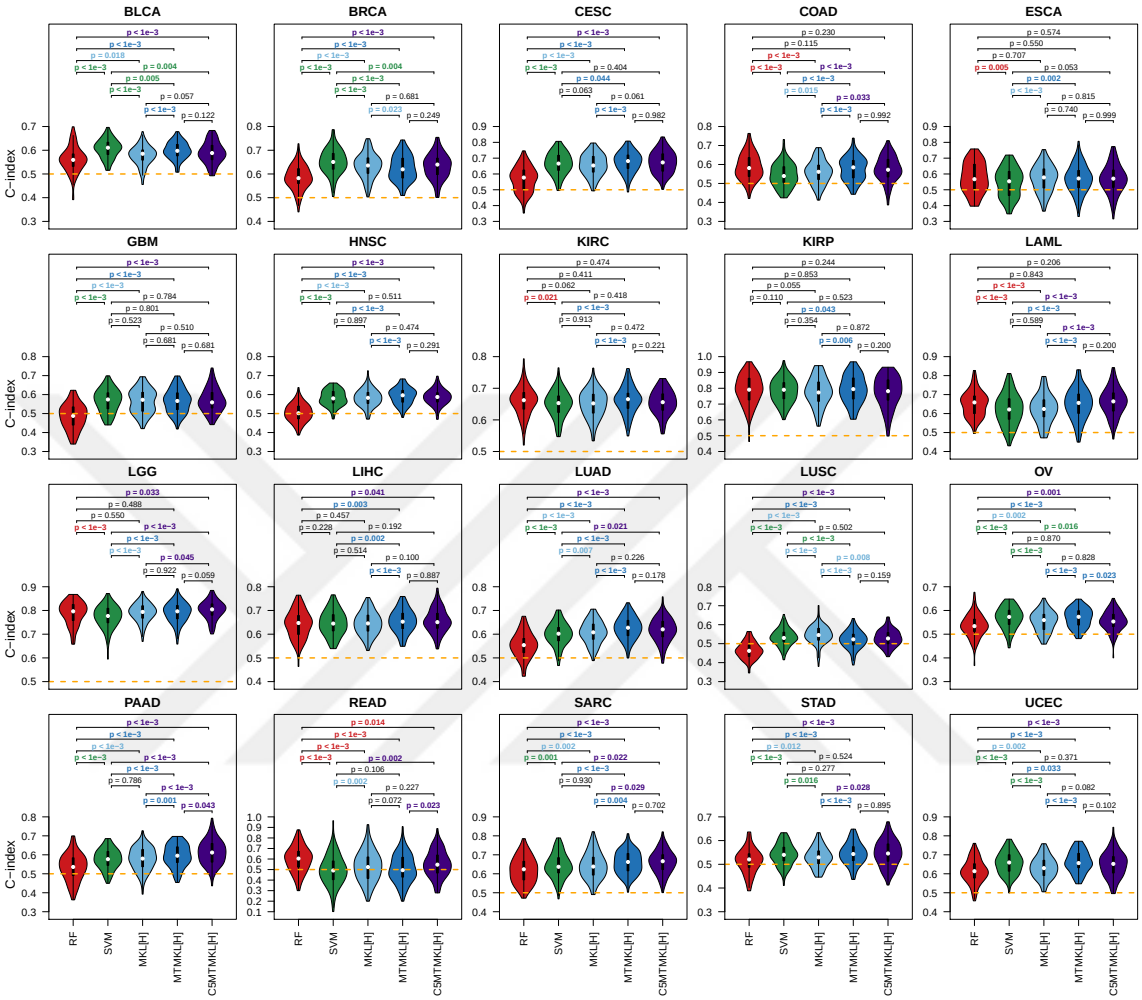

Figure 5.3: The predictive performance comparisons on 20 TCGA datasets for survival RF (`RF`), survival SVM (`SVM`), `MKL[H]`, and `MTMKL[H]` against Path2CSurv algorithm with five clusters (`C5MTMKL[H]`). The concordance index (C-index) values of each algorithm obtained over 100 replications for 20 datasets are compared using a two-tailed paired t-test to report the statistical significance between each algorithm pair. For P-value results, **red**: RF is better; **green**: `SVM` is better; **light blue**: `MKL[H]` is better; **dark blue**: `MTMKL[H]` is better; **magenta**: `C5MTMKL[H]` is better; black: no difference. The dashed lines show the baseline performance level (i.e. C-index = 0.5)

We also report that there is no statistical difference in terms of the predictive performances of MTMKL[H] and Path2CSurv algorithms, except OV, PAAD, and READ datasets. These results show that our clustering-based Path2CSurv algorithm obtains better predictive performances than RF, SVM, and MKL[H] algorithms for survival analysis using gene expression profiles of cancer patients. The predictive performance results also indicate that Path2CSurv algorithm is able to achieve similar predictive performances with its single-cluster multitask variant (i.e. MTMKL) for a harder problem, which includes finding cancer subgroups additionally.

5.4.2 Informative Pathways/Gene Sets for Survival Analysis

In addition to analyzing the predictive performances of our algorithms, we also report the informative pathways/gene sets for the survival prediction of cancer patients which are selected by our algorithms. Our multiple kernel-based algorithms perform survival analysis using significantly fewer gene expression features than RF and SVM algorithms. This decrease can be explained by the fact that multiple kernel learning algorithms decrease the number of features used in the model by assigning zero to the kernel weights of the uninformative pathways/gene sets. Table 5.2 shows the average numbers of gene expression features and pathways/gene sets selected by each algorithm over 100 replications. A pathway/gene set is considered to be selected by our algorithms if its kernel weight is greater than 0.01.

RF and SVM use all available gene expression features (i.e. 19814 in total). The average numbers of gene expression features used are between 71 (LUSC) and 808 (BRCA) for MKL[P], are between 379 (LUSC) and 1922 (BRCA) for MKL[H] algorithms. Since we force our multitask learning-based algorithms to use the same pathways/gene sets for each cohort, the average numbers of gene expression features selected for all datasets are 888 and 1995 for MTMKL[P] and MTMKL[H] algorithms, respectively.

We note that predicting overall survival times in some cancer types is much harder by looking at the pathway/gene set selection frequencies in Table 5.2. For example, MKL[H] algorithm uses less than ten gene sets (i.e. 7.34) for LIHC dataset in the final model, whereas it uses more than ten gene sets (i.e. 13.79) for OV dataset

on the average even though these two datasets are almost the same size. When we consider MKL[P] algorithm, we observe that it uses very few pathways (i.e. 8.55) for LIHC dataset in the final model, whereas the number of pathways selected for OV dataset is 22.75 in the final model.

We also observe that multitask MKL algorithms (i.e. MTMKL[P] and MTMKL[H]) use slightly more pathways/gene sets than MKL algorithms (i.e. MKL[P] and MKL[H]), which results in an increase for the predictive performances of MTMKL algorithms. The increase in the number of pathways/gene sets used by MTMKL algorithms in the final model can be explained by the need for more pathways/gene sets than MKL algorithms to capture underlying survival mechanisms of all cohorts modelled simultaneously. Although the number of pathways/gene sets selected by MTMKL algorithms increases, it is still significantly fewer than RF and SVM algorithms.

Table 5.2 shows that our clustering-based Path2CSurv algorithm, namely C5MTMKL, uses significantly fewer gene expression features than RF and SVM algorithms. We also notice that our clustering-based algorithm uses fewer gene expression features and gene sets than its single-cluster multitask variant (i.e. MTMKL[H]), whereas it uses higher number of gene expression features and gene sets than its single-task variant (i.e. MKL[H]). Figure 5.4 shows that Path2CSurv algorithms with three, four or five cluster count use significantly fewer gene expression features for all datasets than MTMKL algorithm. The reason is that Path2CSurv algorithm clusters the related cancer types, and by this way, it predicts survival time of cancer patients using fewer gene expression features. MKL algorithm uses fewer number of gene expression features than MTMKL and Path2CSurv algorithms as expected for almost all datasets, due to being the single-task version of Path2CSurv. However, for BRCA and OV datasets, Path2CSurv algorithm uses statistically significantly fewer gene expression features than MKL algorithm. We note that there is no statistical difference between MKL and Path2CSurv algorithms for LGG and PAAD datasets in terms of gene expression features used when the number of clusters is set to five.

Table 5.2: The average numbers of gene expression features used by RF, SVM, MKL[P], MTMKL[P], MKL[H], MTMKL[H], and C5MTMKL[H] algorithms and the average numbers of pathways/gene sets used by MKL[P], MTMKL[P], MKL[H], MTMKL[H], and C5MTMKL[H] algorithms

Dataset	RF	SVM	MKL[P]	MTMKL[P]	MKL[H]	MTMKL[H]	C5MTMKL[H]	MKL[P]	MTMKL[P]	MKL[H]	MTMKL[H]	C5MTMKL[H]
BLCA	19814	19814	559	888	1242	1955	1415	18.81	27.95	10.81	17.16	11.63
BRCA	19814	19814	808	888	1922	1955	1289	27.41	27.95	15.54	17.16	10.56
CESC	19814	19814	242	888	983	1955	1278	10.30	27.95	8.25	17.16	10.25
COAD	19814	19814	331	888	860	1955	1151	11.57	27.95	8.00	17.16	9.61
ESCA	19814	19814	365	888	859	1955	1305	13.50	27.95	6.63	17.16	10.89
GBM	19814	19814	273	888	690	1955	1179	10.05	27.95	6.24	17.16	9.62
HNSC	19814	19814	501	888	1257	1955	1413	17.23	27.95	9.97	17.16	11.88
KIRC	19814	19814	316	888	1060	1955	1230	10.07	27.95	9.16	17.16	9.98
KIRP	19814	19814	317	888	689	1955	1102	10.10	27.95	5.04	17.16	8.69
LAML	19814	19814	355	888	780	1955	1275	12.72	27.95	6.38	17.16	10.06

Dataset	RF	SVM	MKL[P]	MTMKL[P]	MKL[H]	MTMKL[H]	C5MTMKL[H]	MKL[P]	MTMKL[P]	MKL[H]	MTMKL[H]	C5MTMKL[H]
LGG	19814	19814	269	888	1082	1955	1132	9.67	27.95	8.18	17.16	8.92
LIHC	19814	19814	266	888	977	1955	1369	8.55	27.95	7.34	17.16	10.87
LUAD	19814	19814	363	888	880	1955	1303	13.07	27.95	6.78	17.16	10.44
LUSC	19814	19814	71	888	379	1955	1358	2.91	27.95	3.90	17.16	11.61
OV	19814	19814	670	888	1727	1955	1496	22.75	27.95	13.79	17.16	11.57
PAAD	19814	19814	369	888	1250	1955	1360	12.46	27.95	8.95	17.16	10.84
READ	19814	19814	275	888	528	1955	1247	10.26	27.95	4.55	17.16	10.38
SARC	19814	19814	391	888	861	1955	1175	13.50	27.95	8.35	17.16	10.18
STAD	19814	19814	378	888	770	1955	1288	14.75	27.95	6.80	17.16	10.92
UCEC	19814	19814	386	888	1009	1955	1295	12.59	27.95	8.77	17.16	10.77

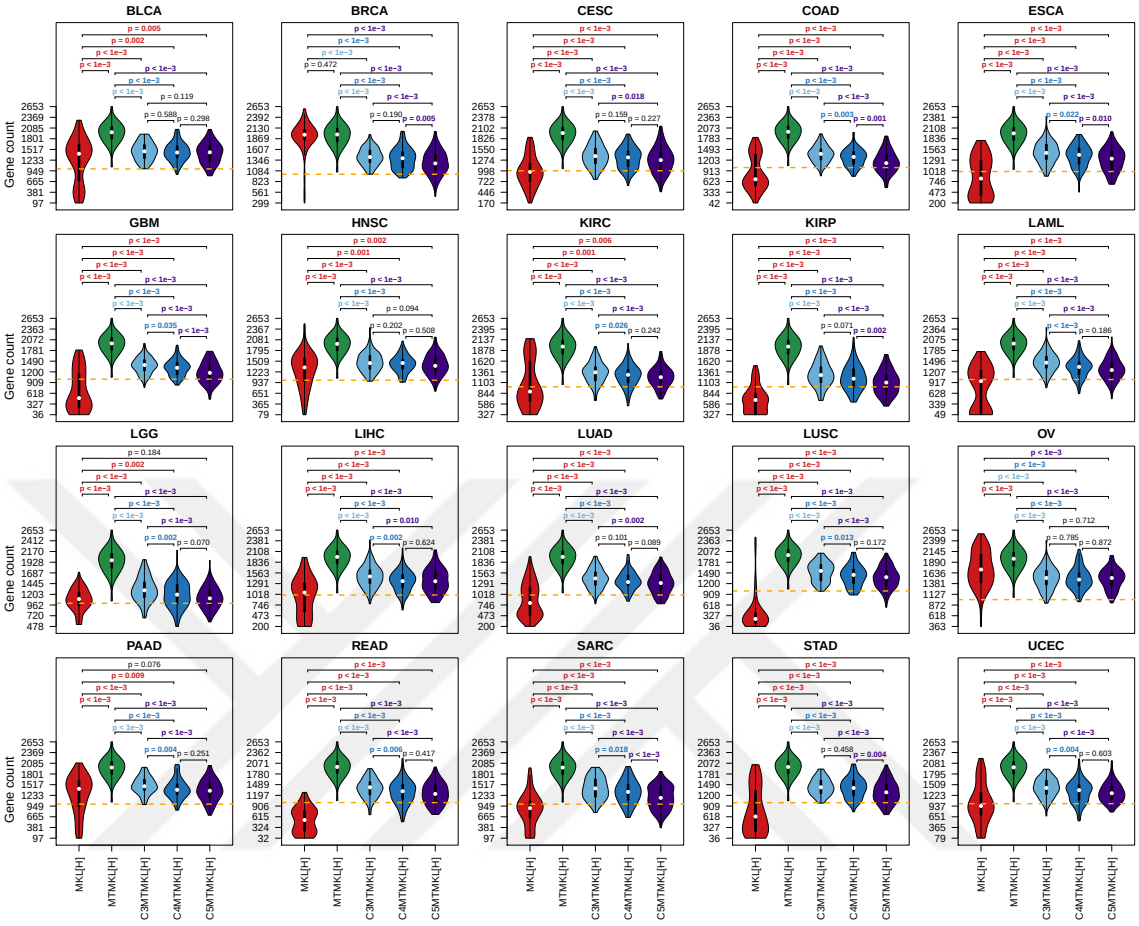

Figure 5.4: The comparisons of number of genes selected on 20 TCGA datasets for single-task variant of Path2CSurv (MKL[H]), single-cluster multitask variant of Path2CSurv (MTMKL[H]), and Path2CSurv with three, four, and five clusters (i.e. C3MTMKL[H], C4MTMKL[H], C5MTMKL[H]). Numbers of genes selected by each algorithm in 100 replications for each dataset are compared in each corresponding violin plot using a two-tailed paired t-test to report the statistical significance between each algorithm pair. The dashed lines represents 1000 genes

In Figure 5.5, we compare the number of gene sets selected by our kernel-based algorithms. We observe that Path2Csurv algorithm uses significantly fewer number of gene sets for all datasets when compared to `MTMKL` algorithm, whereas `MKL` algorithm uses fewer number of gene sets than Path2CSurv algorithm except for `BLCA`, `BRCA`, and `OV` datasets.

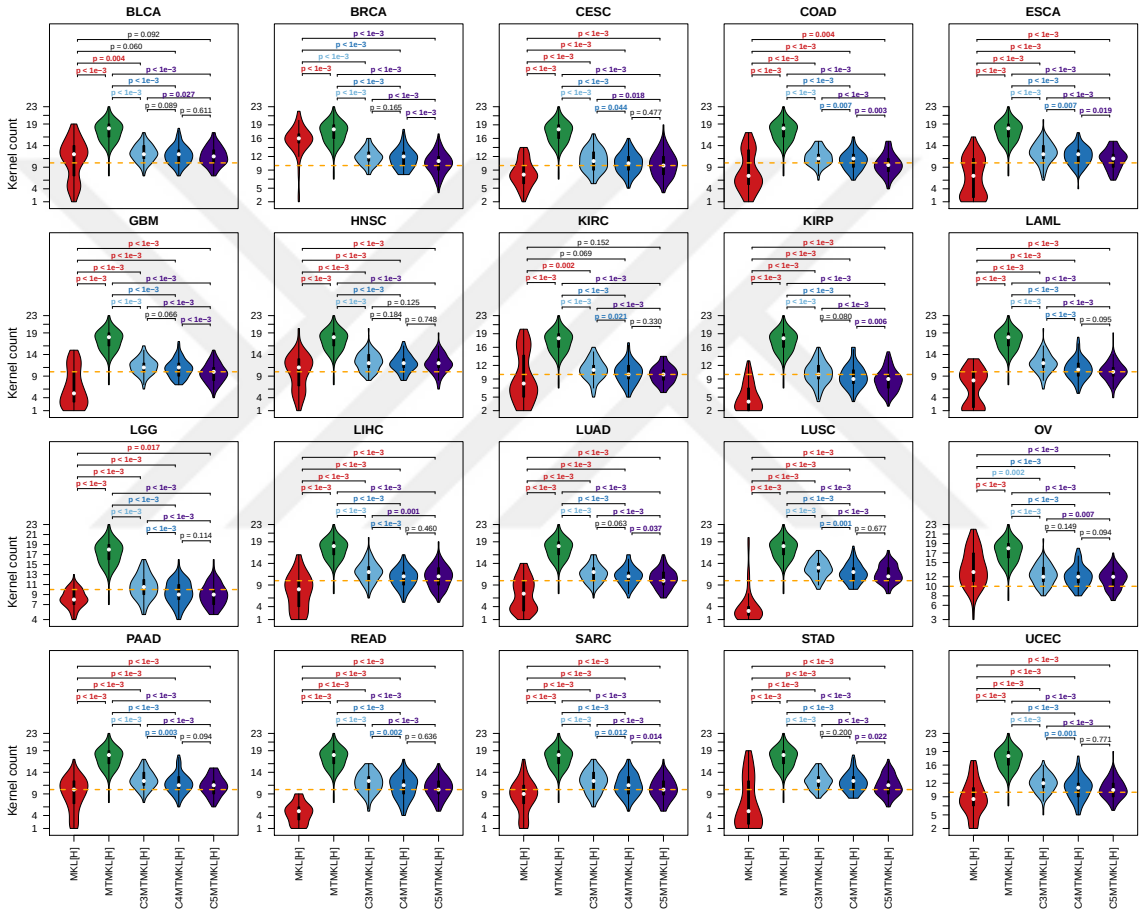

Figure 5.5: The comparisons of numbers of gene sets selected on 20 TCGA datasets for single-task variant of Path2CSurv (MKL[H]), single-cluster multitask variant of Path2CSurv (MTMKL[H]), and Path2CSurv with three, four, and five clusters (i.e. C3MTMKL[H], C4MTMKL[H], and C5MTMKL[H]). Numbers of `Hallmark` gene sets selected by each algorithm in 100 replications for each dataset are compared using two-tailed paired t-tests to report the statistical significance between each algorithm pair. The dashed lines represents 10 gene sets.

Figure 5.6 and Figure 5.7 report the selection frequencies of 50 `Hallmark` gene sets and top 50 `PID` pathways over 100 replications by Path2Surv algorithm, respectively. As mentioned before, we consider a gene set is selected to be included in the prediction model if its kernel weight is greater than 0.01. By looking at the row sums of the selection frequencies in these figures, we can identify the informative and uninformative pathways/gene sets for survival analysis of cancer patients. Figure 5.6 shows that `BILE_ACID_METABOLISM`, `APICAL_SURFACE`, `KRAS_SIGNALING_DN`, `SPERMATOGENESIS`, and `ANGIOGENESIS` are selected in the final model by Path2Surv algorithm for more than 6 out of 20 datasets on the average. These results seem reasonable when we check the functions of these gene sets. For example, `ANGIOGENESIS` is responsible from the process which forms new blood vessels, and it plays a vital role in formation of cancer since tumours need blood while they are growing. We also notice that some of the immune response related gene sets (i.e. `COMPLEMENT`, `IL2_STAT5_SIGNALING`, and `INTERFERON_GAMMA_RESPONSE`) are picked as informative for less than 2 out of 20 datasets on the average. When we look at the `PID` pathway selection frequencies in Figure 5.7, we notice that `CONE_PATHWAY`, `ERBB_NETWORK_PATHWAY`, `RHODOPSIN_PATHWAY`, `HNF3B_PATHWAY`, and `CIRCADIAN_PATHWAY`, which are known key biological mechanisms in cancer, are included in the final model by Path2Surv algorithm for more than 5 out of 20 datasets on the average.

Figure 5.6: The selection frequencies of 50 gene sets in the `Hallmark` collection over 100 replications by Path2Surv algorithm. The rows and the columns are clustered using the hierarchical clustering algorithm with the Euclidean distance and complete linkage functions. The column sums of selection frequencies are reported to identify datasets that use higher number of gene sets on the average. The row sums of selection frequencies are reported to identify frequently selected gene sets across different datasets.

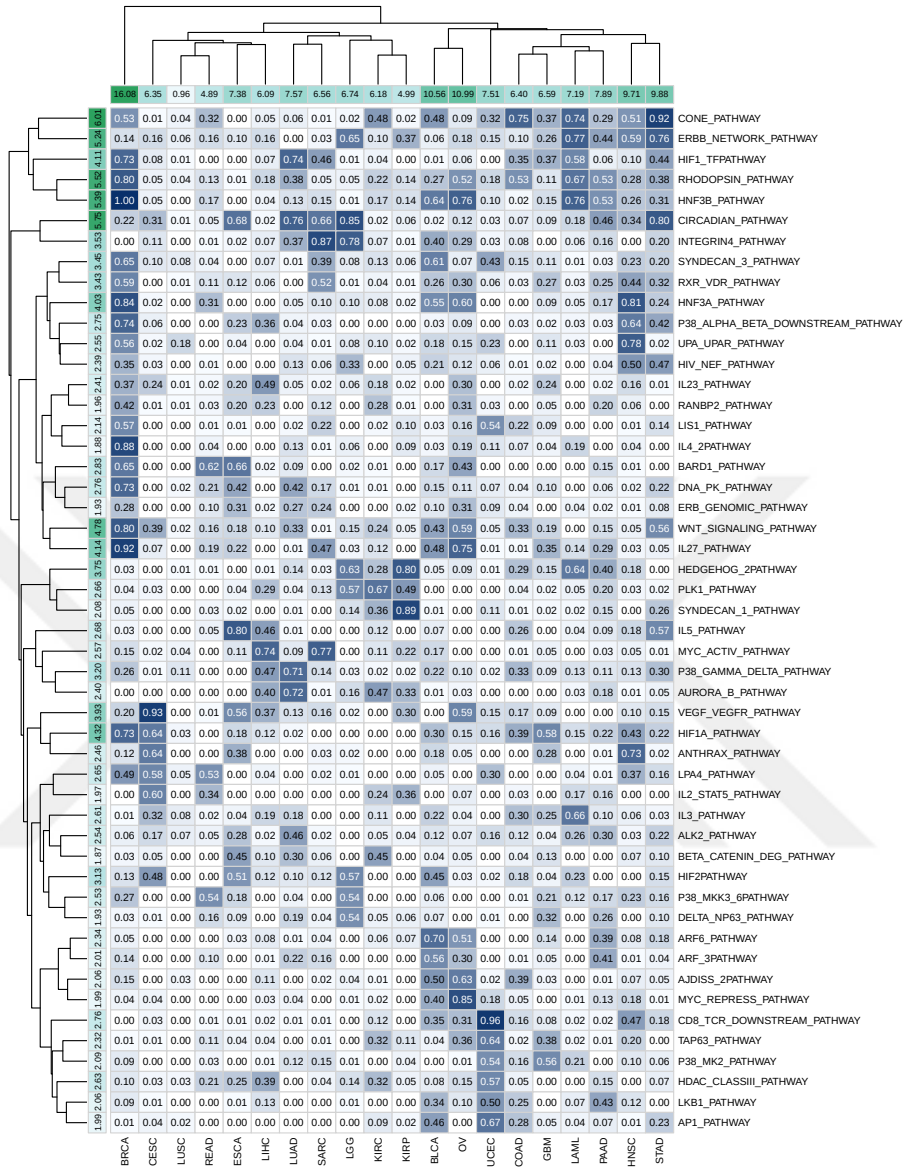

Figure 5.7: The selection frequencies of top 50 pathways in the `Pathway Interaction Database` collection over 100 replications by Path2Surv algorithm. The rows and the columns are clustered using the hierarchical clustering algorithm with the Euclidean distance and complete linkage functions. The column sums of selection frequencies are reported to identify datasets that use higher number of pathways on the average. The row sums of selection frequencies are reported to identify frequently selected pathways across different datasets.

Our Path2MSurv algorithm uses a shared set of kernel weights to identify informative pathways/gene sets for each dataset. In Figure 5.8 and Figure 5.9, we report the selection frequencies of 50 `Hallmark` gene sets and top 50 `PID` pathways over 100 replications by Path2MSurv algorithm. In Figure 5.8, we notice that 19 out of 50 `Hallmark` gene sets are picked by our Path2MSurv algorithm as informative in at least 50 replications. `GLYCOLYSIS` and `ANGIOGENESIS` gene sets are the most informative gene sets for all cancer datasets with 100% selection frequencies, which are known to be key biological mechanisms that cancer cells benefit from. `KRAS_SIGNALING_DN`, `SPERMATOGENESIS`, `APOPTOSIS`, `APICAL_SURFACE`, and `BILE_ACID_METABOLISM` are selected in more than 90 replications. Figure 5.9 shows that 26 out of 196 `PID` pathways are picked by our Path2MSurv algorithm as informative in at least 50 replications. The most informative pathways are `P73PATHWAY`, `BETA_CATENIN_NUC_PATHWAY`, `HIF2PATHWAY`, `CONE_PATHWAY`, `HNF3B_PATHWAY`, `MYC_ACTIV_PATHWAY`, `WNT_SIGNALING_PATHWAY`, and `IL23_PATHWAY`, which are selected in almost all replications and are also known to be key biological mechanisms in cancer.

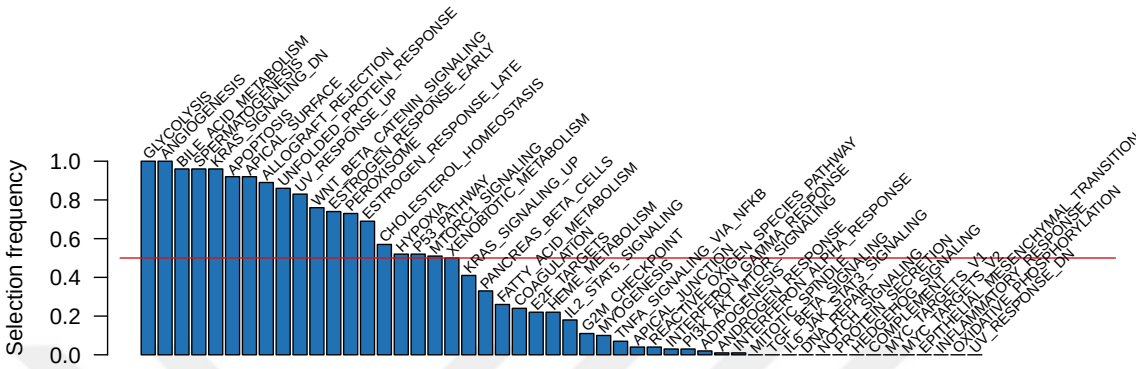

Figure 5.8: Selection frequencies of 50 gene sets in the `Hallmark` collection over 100 replication by Path2MSurv algorithm. The red line shows where the selection frequency is 50%.

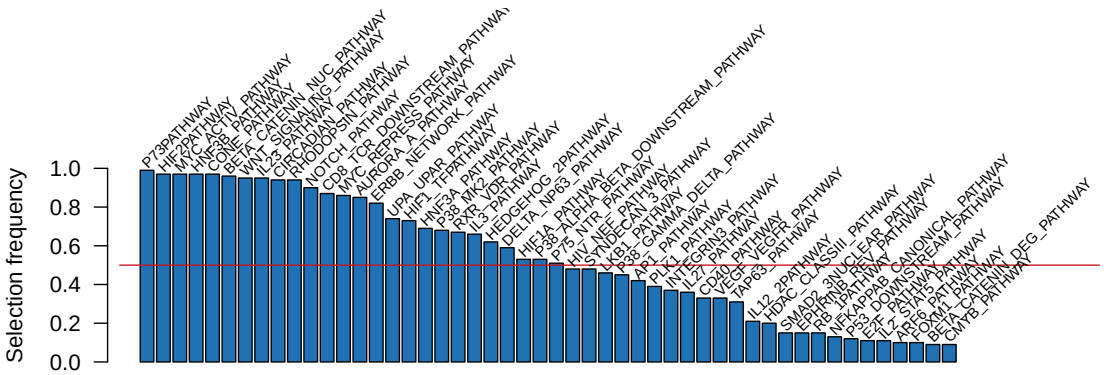

Figure 5.9: Selection frequencies of top 50 out of 196 pathways in the `PID` collection over 100 replications by Path2MSurv algorithm. The red line shows where the selection frequency is 50%.

Our Path2CSurv algorithm identifies groups of cancer types that share the same biological mechanisms during survival prediction task. It also forces each cohort within a cluster to use the same set of kernel weights for `Hallmark` gene sets. Figure 5.10 reports the selection frequencies of 50 `Hallmark` gene sets over 100 replications by Path2CSurv algorithm when the number of clusters is set to five. We observe that some of the key biological mechanisms for formation and progression of cancer (namely `KRAS_SIGNALING_DN`, `GLYCOLYSIS`, `ANGIOGENESIS`, `SPERMATOGENESIS`, and `BILE_ACID_METABOLISM` gene sets) are selected as informative by our Path2CSurv algorithm for more than 10 out of 20 datasets on the average. We also notice that some of the immune response related gene sets (namely `IL2_STAT5_SIGNALING`, `INTERFERON_GAMMA_RESPONSE`, and `COMPLEMENT`) are included in the final model for less than 2 out of 20 datasets on the average. In Appendix D, we report the selection frequencies of 50 `Hallmark` gene sets selected by Path2MSurv algorithm when the number of clusters is set to three, four, six, seven, and eight.

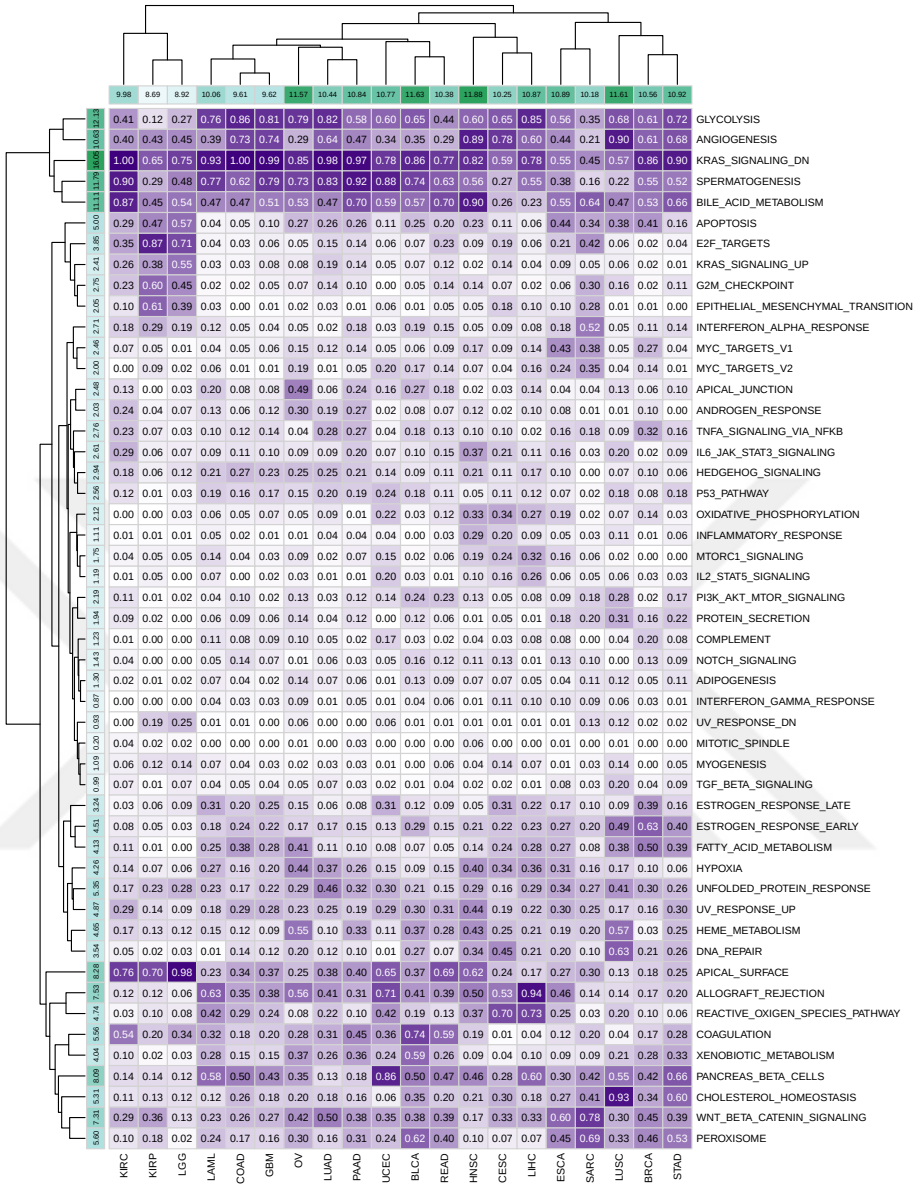

Figure 5.10: The selection frequencies of 50 gene sets over 100 replications by Path2CSurv algorithm when the number of clusters is set to five. The rows and the columns are clustered using the hierarchical clustering algorithm with the Euclidean distance and complete linkage functions. The row and the column sums report the frequently selected gene sets across different datasets and the number of gene sets used by each dataset on the average, respectively.

5.4.3 Cluster Structures Identified by Path2CSurv

Our Path2CSurv algorithm has the ability to identify clusters among multiple cancer cohorts. Since we do not know initially how many clusters should be formed, we try different cluster counts as the initial parameter for Path2CSurv. The following procedure is applied to determine the cluster structures. The assignment frequencies of each dataset pair to the same cluster in 100 replications are calculated using the cluster assignments. To visualize the obtained cluster structures, we benefit from the *t*-Distributed Stochastic Neighbour Embedding (*t*-SNE) algorithm [Maaten and Hinton, 2008]. *t*-SNE algorithm is basically a non-linear dimensionality reduction algorithm that is used to visualize input data by embedding high-dimensional structure of the given data into a low-dimensional space with the help of observed similarities/distances between the data points.

In this section, we focus on the cluster structure obtained when we set the number of clusters to five. The reason is that Path2CSurv algorithm with five clusters has a better visualization than the other cluster counts in terms of clustering. Network representations showing the cluster structures for three, four, six, seven, and eight cluster counts can be seen in Appendix E.

Figure 5.11 shows the cluster structure obtained by Path2CSurv when the cluster count is set to five. The cluster assignments are as follows: i) (`BLCA`, `KIRC`, `LUAD`, `PAAD`, `READ`), ii) (`BRCA`, `COAD`, `GBM`, `LAML`, `OV`, `STAD`, `UCEC`), iii) (`CESC`, `HNSC`, `LIHC`), iv) (`ESCA`, `LUSC`, `SARC`), v) (`KIRP`, `LGG`). We evaluate the clusters according to their histological type and primary tumour sites. In pan-cancer studies, it was shown that different cancer types can be classified according to their tissue types where the cancer originates (i.e. histological type) or the location of the cancer where it is first developed (i.e. the primary site of tumour in the body) [Hoadley et al., 2018]. There are many types of cancers based on the histology of tumours, such as adenocarcinoma, squamous cell carcinoma, sarcoma, myeloma, leukemia, lymphoma, and mixed types. The cancer types can also be grouped according to their primary tumour site, such as urologic, gynecologic, gastrointestinal, central nervous system, head and neck, hematologic and lymphatic, thoracic, and soft tissue cancers.

We observe that four of the adenocarcinomas (i.e. `STAD`, `OV`, `COAD`, and `BRCA`) are grouped in the same cluster. It is also known that `BRCA`, `OV`, and `UCEC` are gynecologic cancer types that were proved to have biological similarities [Hoadley et al., 2018]. In addition, `STAD` and `COAD` are gastrointestinal cancers that have the tissue similarities. Squamous cell carcinomas are divided into two different clusters as `LUSC`–`ESCA` and `CESC`–`HNSC`. We also observe that two urologic (i.e. `KIRC` and `BLCA`) and two gastrointestinal (i.e. `PAAD` and `READ`) cancers are clustered together. We notice that `READ`, `PAAD`, and `LUAD` are the adenocarcinomas that are assigned to the same cluster.

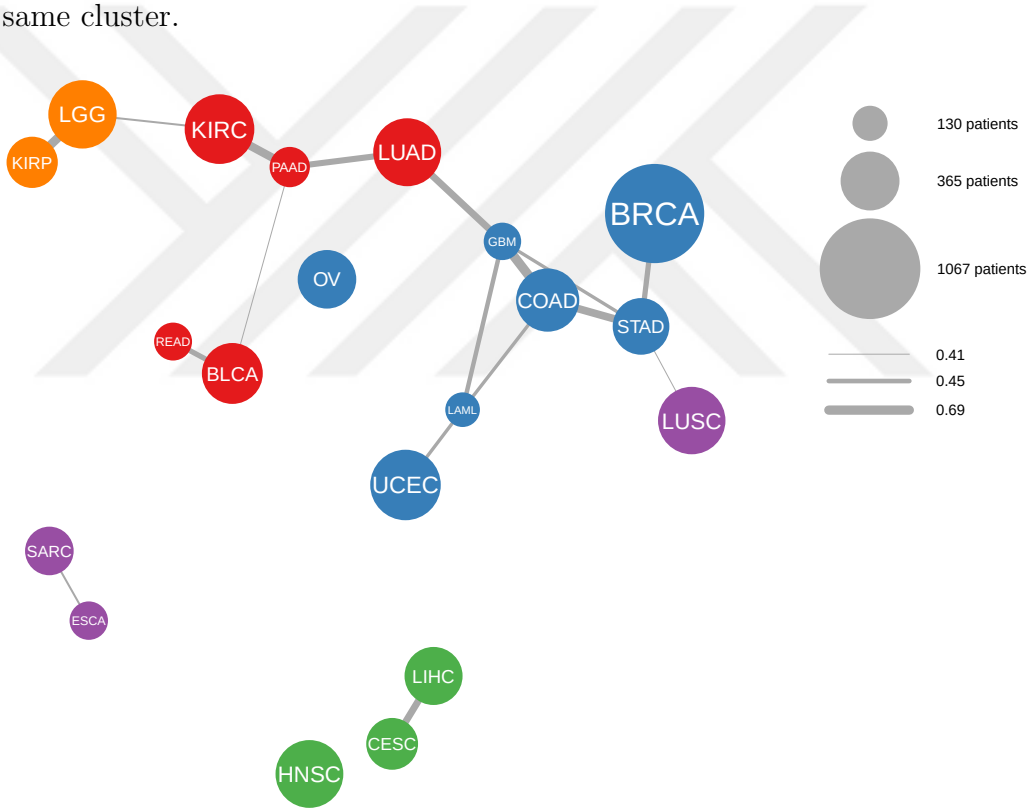

Figure 5.11: Cluster structure of 20 TCGA datasets obtained by Path2CSurv algorithm when the number of clusters is set to five. Each node represents a cancer, and the edges between nodes show the assignment frequencies of connected pairs to the same cluster over 100 replications. Node and edge sizes are proportional to the number of patients in the cohorts and the assignment frequencies, respectively. The edges that have the frequencies below 0.40 were not shown in the figure. Each color refers to a different cluster.

Chapter 6

CONCLUSION

Survival analysis using genomic characterizations of cancer patients plays a vital role in understanding the disease progression and formation mechanisms. Integrating prior information about pathways/gene sets into survival analysis enables us to identify informative biological mechanisms in survival prediction of cancer patients. In this thesis, we focus on developing novel machine learning algorithms for survival analysis to identify survival associated pathways/gene sets, to understand the common underlying biological mechanisms between multiple cancer types, and to determine cluster structures among multiple cancer types.

Existing approaches perform survival analysis and knowledge extraction steps separately. Rather than performing one of these two steps before the other one, we propose conjoint modelling approaches that can pick informative pathways/gene sets from a given collection and perform survival analysis using only these selected pathways/gene sets. The first contribution of this thesis is that we develop a multiple kernel learning-based survival analysis algorithm (namely Path2Surv) that can integrate pathway/gene set information into the survival analysis model during training step and use the subset of genomic characterizations mapped to the selected pathways/gene sets during the survival prediction [Dereli et al., 2019b].

As the second contribution, we propose the extended version our Path2Surv algorithm towards multitask learning [Caruana, 1997], namely Path2MSurv, which is known to increase the predictive performances of machine learning algorithms by exploiting the commonalities between multiple related tasks [Dereli et al., 2019a]. By doing so, we aim to identify common underlying biological mechanisms between multiple cancer types.

The third contribution of this thesis is as follows. Instead of simultaneous mod-

elling of all cancer types, which assumes all tasks included into the model shares the same biological mechanisms, we develop a unified formulation for clustering of cancer datasets, learning the common underlying biological mechanisms for each cluster and predicting the overall survival time of cancer patients (Path2CSurv) as the third contribution of this thesis. By applying these three steps conjointly, we aim to identify underlying mechanisms of multiple cancer types in a more robust manner. Our Path2CSurv algorithm is also able to cluster cancer types into meaningful groups. Results show that some cancer types that have similar histopathology or primary tumour site are grouped in the same cluster by our clustering algorithm.

Our algorithms are tested using 20 different cancer cohorts obtained from TCGA (Table 5.1). We also use two cancer-specific pathway/gene set collections (i.e. `Hallmark` [Liberzon et al., 2015] and `PID` [Schaefer et al., 2009]) to identify the informative biological mechanisms. The computational results show that our multiple kernel learning-based algorithms (i.e. Path2Surv, Path2MSurv, and Path2CSurv) obtain statistically significantly better or comparable predictive performances against survival RF [Ishwaran et al., 2008] and survival SVM [Khan and Zubek, 2008; Shivaswamy et al., 2007] algorithms. Our algorithms are able to solve a harder problem (i.e. clustering multiple cancer datasets and identifying informative biological mechanisms for survival prediction) by using significantly fewer gene expression features than survival RF and survival SVM algorithms. Therefore, our algorithms improve the interpretability of gene expression features while they are decreasing the model complexity by selecting less molecular mechanisms as informative for disease progression mechanisms. This contribution also decreases the data acquisition cost for the studies that are performed for understanding the cancer.

Our algorithms can also be applied to other diseases with a suitable genomic data and pathway/gene set collection. While applying these algorithms to other diseases, it must be considered that pathways/gene sets should include biological mechanisms that are relevant with the disease in question, so that the algorithm could identify the informative biological mechanisms and make reliable survival predictions.

As a future work, our Path2CSurv algorithm can be extended towards clustering

of complete set of tumours biopsied from multiple cancer types. In this study, we assume that each tumour in a cohort belongs to the same group. However, it is also possible that two tumour samples belonging to the same cohort might have different underlying biological mechanisms. Therefore, a model that aims to identify molecular subgroups by evaluating all tumour samples as a single dataset might give more robust predictive performances.

Appendix A

DERIVATION OF KERNEL UPDATE FUNCTION

Let us define

$$\eta_P = 1 - \sum_{o=1}^{P-1} \eta_o,$$

and rewrite the objective function in 2.9 as follows:

$$\mathcal{L} = \frac{1}{2}\left(\frac{1}{\eta_1}\widetilde{\boldsymbol{w}}_1^\top \widetilde{\boldsymbol{w}}_1 + \cdots + \frac{1}{\eta_m}\widetilde{\boldsymbol{w}}_m^\top \widetilde{\boldsymbol{w}}_m + \cdots + \frac{1}{1 - \sum_{o=1}^{P-1} \eta_o}\widetilde{\boldsymbol{w}}_P^\top \widetilde{\boldsymbol{w}}_P \right),$$

where $\{\boldsymbol{\eta} \in \mathbb{R}^P : \mathbf{1}^\top \boldsymbol{\eta} = 1,\ \boldsymbol{\eta} \geq 0\}$. The derivative of $\mathcal{L}$ is calculated as

$$\frac{\partial \mathcal{L}}{\partial \eta_m} = 0 \Rightarrow \eta_m^2 = \frac{-\widetilde{\boldsymbol{w}}_m^\top \widetilde{\boldsymbol{w}}_m \left[1 - \sum_{o=1}^{P-1} \eta_o\right]^2}{\widetilde{\boldsymbol{w}}_P^\top \widetilde{\boldsymbol{w}}_P} \qquad \forall m \in [1, P-1].$$

Then,

$$\sum_{m=1}^{P-1} \eta_m = \frac{\sum_{m=1}^{P-1} \sqrt{\widetilde{\boldsymbol{w}}_m^\top \widetilde{\boldsymbol{w}}_m}\left[1 - \sum_{o=1}^{P-1} \eta_o\right]}{\sqrt{\widetilde{\boldsymbol{w}}_P^\top \widetilde{\boldsymbol{w}}_P}}.$$

Since $\sum_{m=1}^{P} \eta_m = 1$ and $\eta_P = 1 - \sum_{o=1}^{P-1} \eta_o$,

$$\sum_{m=1}^{P} \eta_m = \frac{\sum_{m=1}^{P-1} \sqrt{\widetilde{\boldsymbol{w}}_m^\top \widetilde{\boldsymbol{w}}_m}\left[1 - \sum_{o=1}^{P-1} \eta_o\right]}{\sqrt{\widetilde{\boldsymbol{w}}_P^\top \widetilde{\boldsymbol{w}}_P}} + (1 - \sum_{o=1}^{P-1} \eta_o).$$

$$\rightarrow 1 = \frac{\sum_{m=1}^{P-1} \sqrt{\widetilde{\boldsymbol{w}}_m^\top \widetilde{\boldsymbol{w}}_m}\left[\eta_P\right]}{\sqrt{\widetilde{\boldsymbol{w}}_P^\top \widetilde{\boldsymbol{w}}_P}} + (\eta_P)$$

$$\rightarrow \eta_P = \frac{\sqrt{\widetilde{\boldsymbol{w}}_P^\top \widetilde{\boldsymbol{w}}_P}}{\sum_{m=1}^{P} \sqrt{\widetilde{\boldsymbol{w}}_m^\top \widetilde{\boldsymbol{w}}_m}}$$

Appendix B

STATISTICAL TESTS USED

Two-tailed paired t-test

We use two-tailed paired t-test to compare the C-index values obtained for each algorithm over 100 replications. A paired t-test is a statistical technique that is used to compare the means of two populations where observations in one population can be paired with the observations in the other population.

Assume that we have C-index values obtained for two different algorithms using the same dataset. Suppose that each of these algorithms has P number of observations for C-index values. By comparing their means for C-index observations, we want to find out which algorithm obtains better C-index values.

Let x_i and y_i are the C-index values for each algorithm, where $i = 1, \ldots, P$. We define our null hypothesis H_0 as the true mean difference is zero (i.e. $\mu_d = 0$). The test statistics is calculated as follows:

$$t = \frac{\bar{d} - \mu_d}{s_{\bar{d}}},$$

where $\bar{d}$ is the mean of differences between pairs of observations (i.e. $(1/P)\sum_i^P x_i - y_i$) and $s_{\bar{d}}$ is the standard deviation of the sample differences. When the null hypothesis is true, test statistics is distributed as t distribution with $P - 1$ degrees of freedom. Let us define the probability of rejecting the null hypothesis H_0 is $\alpha = 0.05$. If the corresponding p-value of the test statistics with $P - 1$ degrees of freedom is greater than α, we fail to reject H_0. Otherwise, a positive test statistic refers to that the mean of the observations of the first algorithm (i.e. $(1/P)\sum_i^P x_i$) is greater than the mean of the observations of the second algorithm (i.e. $(1/P)\sum_i^P y_i$); whereas a negative test statistic refers to that the mean of the observations of the second algorithm is greater than the mean of the observations of the first algorithm.

Appendix C

PREDICTIVE PERFORMANCE COMPARISONS OF PATH2CSURV ALGORITHM

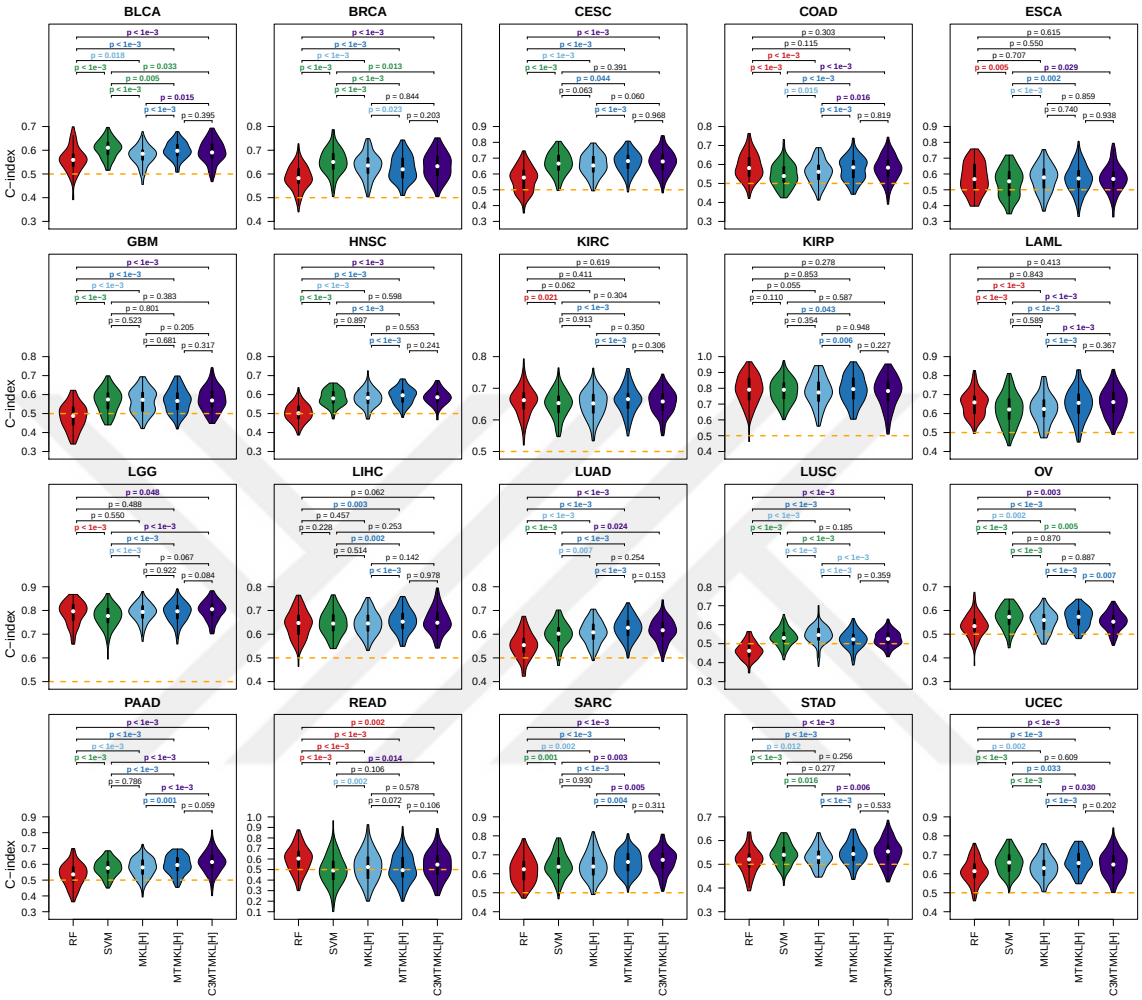

Figure C.1: The predictive performance comparisons on 20 TCGA datasets for survival RF (RF), survival SVM (SVM), `MKL[H]`, and `MTMKL[H]` against Path2CSurv algorithm with three clusters (C3MTMKL[H]). The concordance index (C-index) values of each algorithm obtained over 100 replications for 20 datasets are compared using a two-tailed paired t-test to report the statistical significance between each algorithm pair. For P-value results, **red**: RF is better; **green**: `SVM` is better; **light blue**: `MKL[H]` is better; **dark blue**: `MTMKL[H]` is better; **magenta**: `C3MTMKL[H]` is better; black: no difference. The dashed lines show the baseline performance level (i.e. C-index = 0.5)

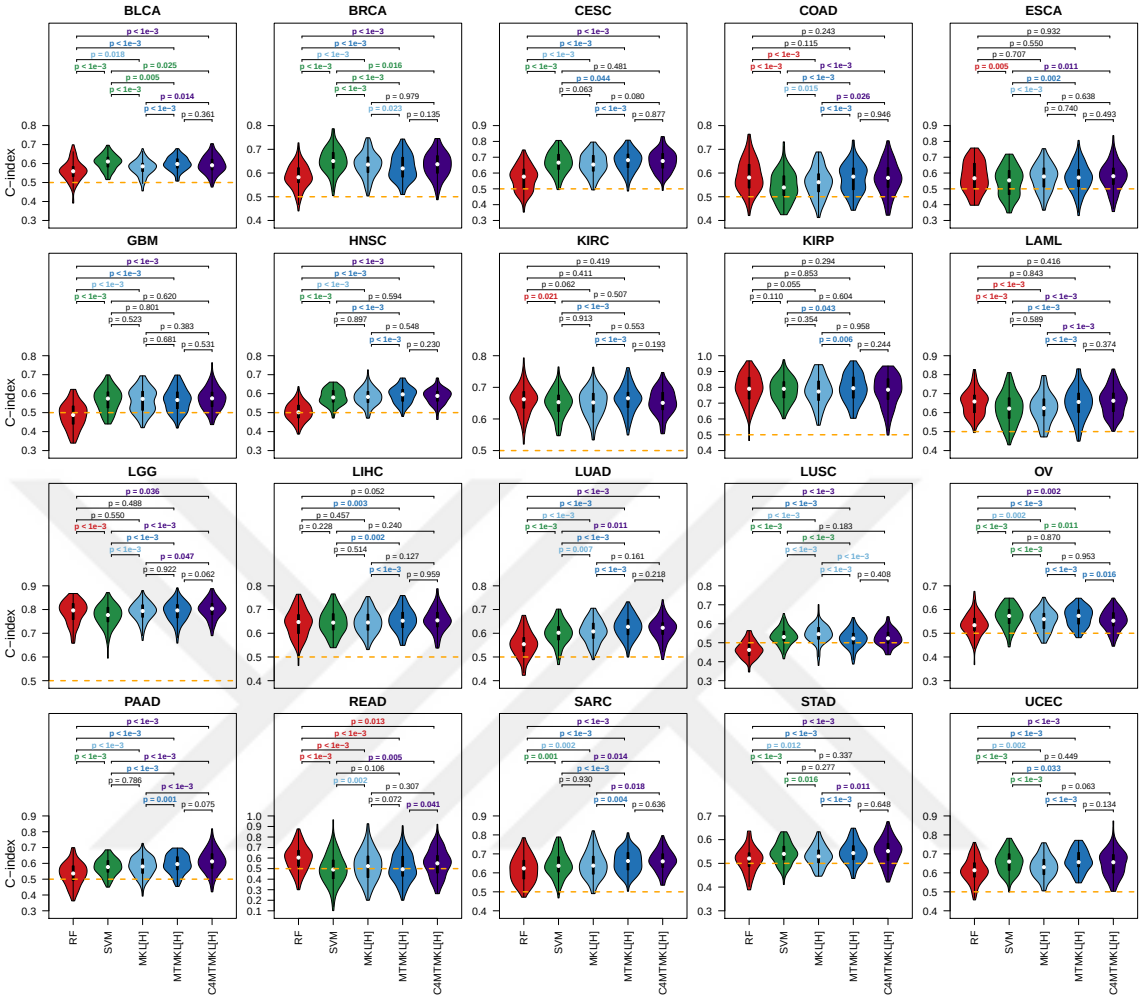

Figure C.2: The predictive performance comparisons on 20 TCGA datasets for survival RF (RF), survival SVM (SVM), `MKL[H]`, and `MTMKL[H]` against Path2CSurv algorithm with four clusters (C4MTMKL[H]). The concordance index (C-index) values of each algorithm obtained over 100 replications for 20 datasets are compared using a two-tailed paired t-test to report the statistical significance between each algorithm pair. For P-value results, **red**: RF is better; **green**: `SVM` is better; **light blue**: `MKL[H]` is better; **dark blue**: `MTMKL[H]` is better; **magenta**: `C4MTMKL[H]` is better; black: no difference. The dashed lines show the baseline performance level (i.e. C-index = 0.5)

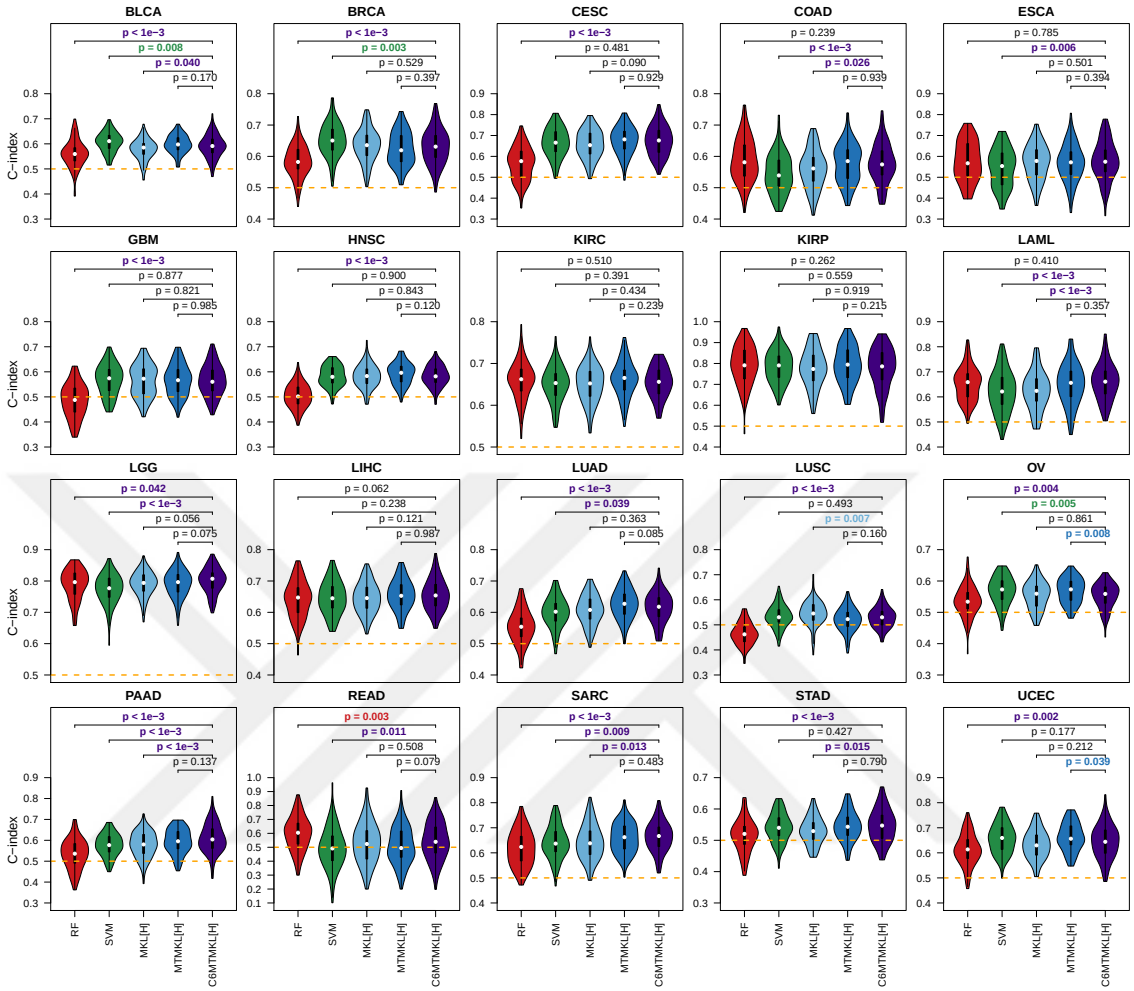

Figure C.3: The predictive performance comparisons on 20 TCGA datasets for survival RF (RF), survival SVM (SVM), `MKL[H]`, and `MTMKL[H]` against Path2CSurv algorithm with six clusters (C6MTMKL[H]). The concordance index (C-index) values of each algorithm obtained over 100 replications for 20 datasets are compared using a two-tailed paired t-test to report the statistical significance between each algorithm pair. For P-value results, **red**: RF is better; **green**: `SVM` is better; **light blue**: `MKL[H]` is better; **dark blue**: `MTMKL[H]` is better; **magenta**: `C6MTMKL[H]` is better; black: no difference. The dashed lines show the baseline performance level (i.e. C-index = 0.5)

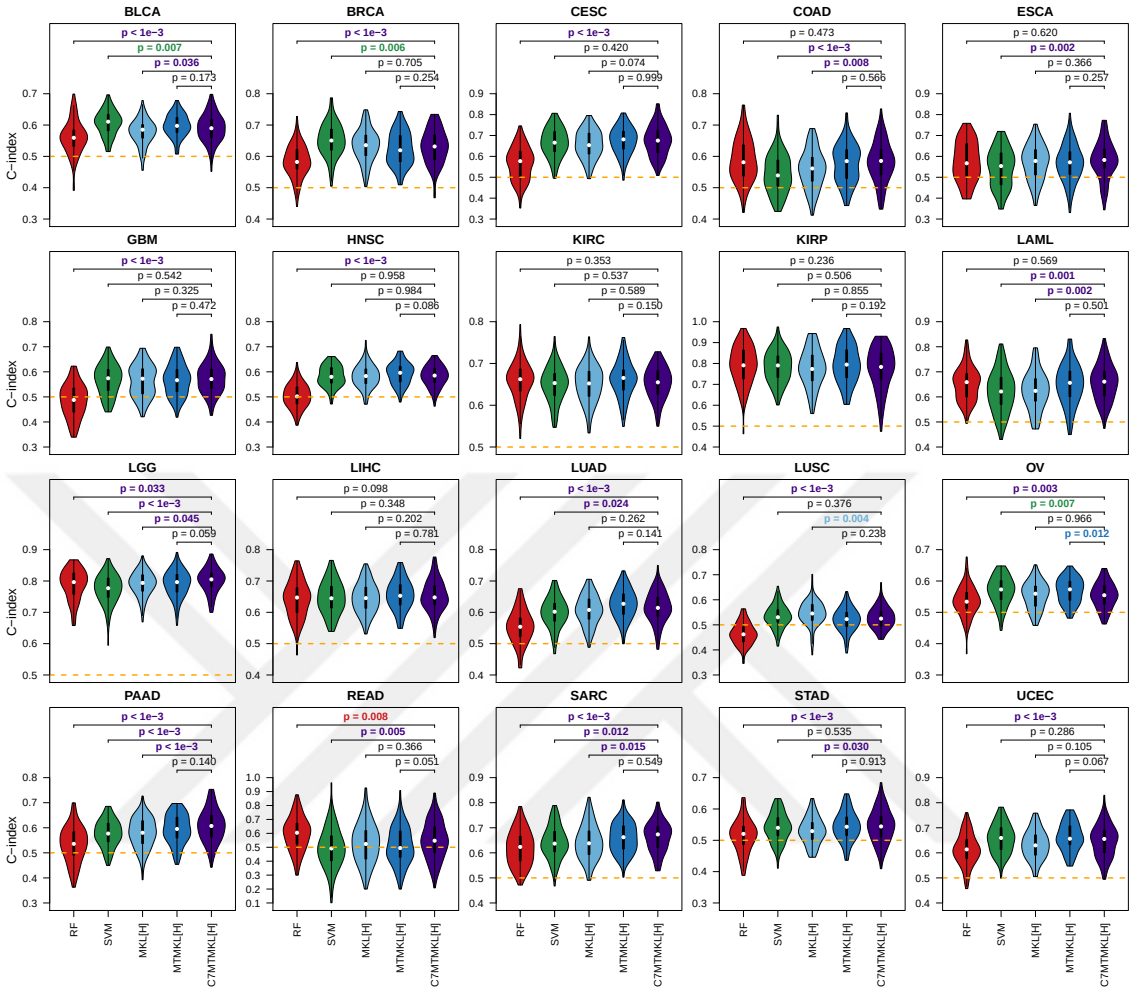

Figure C.4: The predictive performance comparisons on 20 TCGA datasets for survival RF (RF), survival SVM (SVM), `MKL[H]`, and `MTMKL[H]` against Path2CSurv algorithm with seven clusters (C7MTMKL[H]). The concordance index (C-index) values of each algorithm obtained over 100 replications for 20 datasets are compared using a two-tailed paired t-test to report the statistical significance between each algorithm pair. For P-value results, **red**: RF is better; **green**: `SVM` is better; **light blue**: `MKL[H]` is better; **dark blue**: `MTMKL[H]` is better; **magenta**: `C7MTMKL[H]` is better; black: no difference. The dashed lines show the baseline performance level (i.e. C-index = 0.5)

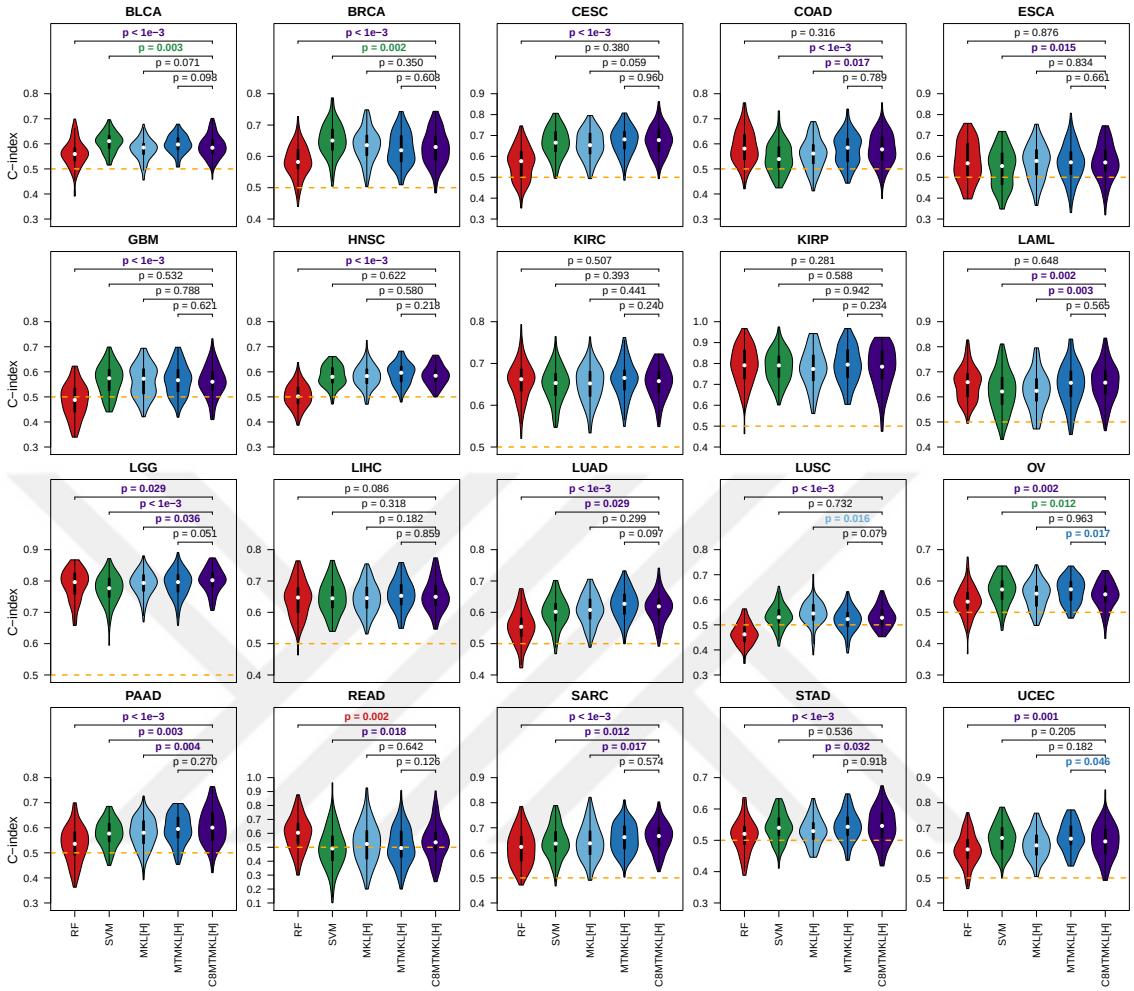

Figure C.5: The predictive performance comparisons on 20 TCGA datasets for survival RF (RF), survival SVM (SVM), `MKL[H]`, and `MTMKL[H]` against Path2CSurv algorithm with eight clusters (C8MTMKL[H]). The concordance index (C-index) values of each algorithm obtained over 100 replications for 20 datasets are compared using a two-tailed paired t-test to report the statistical significance between each algorithm pair. For P-value results, **red**: RF is better; **green**: `SVM` is better; **light blue**: `MKL[H]` is better; **dark blue**: `MTMKL[H]` is better; **magenta**: `C8MTMKL[H]` is better; black: no difference. The dashed lines show the baseline performance level (i.e. C-index = 0.5)

Appendix D

GENE SET SELECTION FREQUENCIES BY PATH2CSURV ALGORITHM

Figure D.1: The selection frequencies of 50 gene sets over 100 replications by Path2CSurv algorithm when the number of clusters is set to three. The rows and the columns are clustered using the hierarchical clustering algorithm with the Euclidean distance and complete linkage functions. The row and the column sums report the frequently selected gene sets across different datasets and the number of gene sets used by each dataset on the average, respectively.

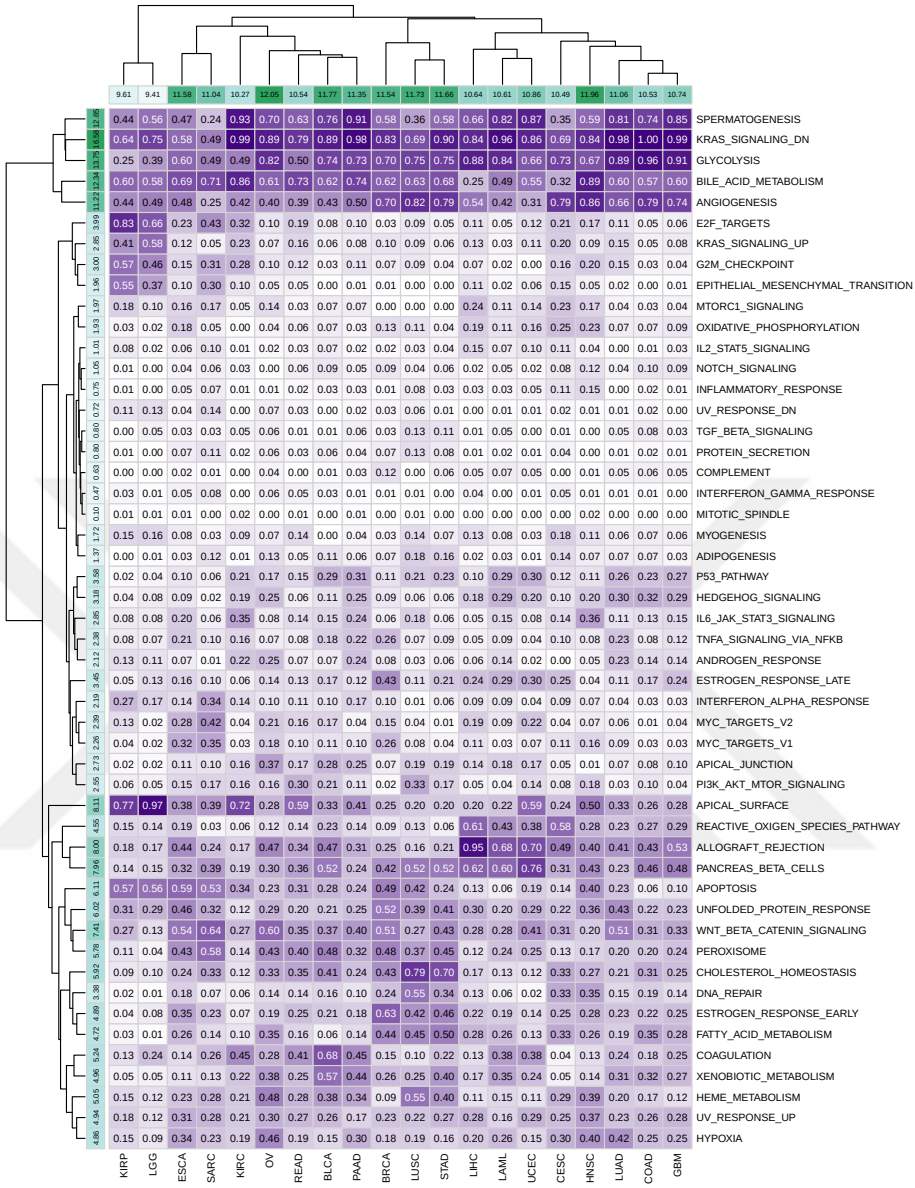

Figure D.2: The selection frequencies of 50 gene sets over 100 replications by Path2CSurv algorithm when the number of clusters is set to four. The rows and the columns are clustered using the hierarchical clustering algorithm with the Euclidean distance and complete linkage functions. The row and the column sums report the frequently selected gene sets across different datasets and the number of gene sets used by each dataset on the average, respectively.

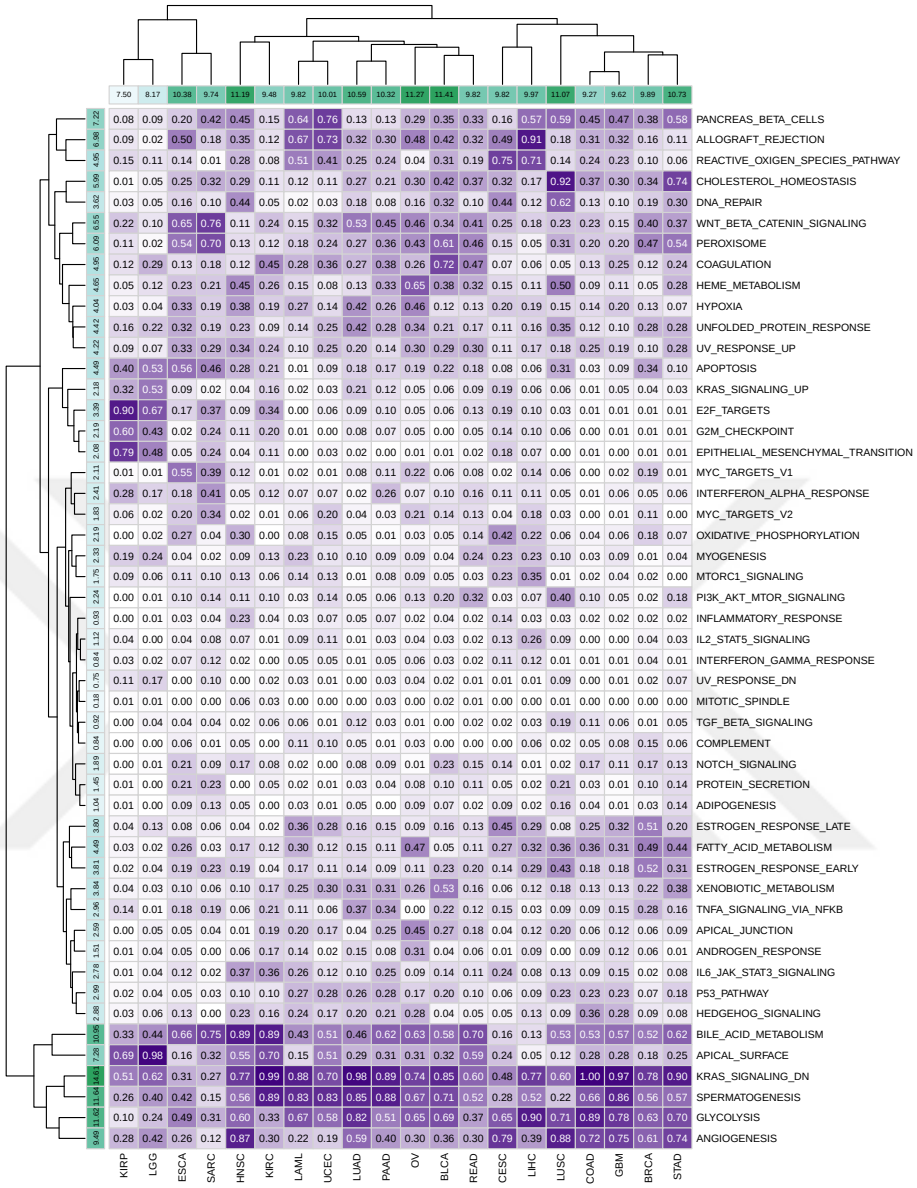

Figure D.3: The selection frequencies of 50 gene sets over 100 replications by Path2CSurv algorithm when the number of clusters is set to six. The rows and the columns are clustered using the hierarchical clustering algorithm with the Euclidean distance and complete linkage functions. The row and the column sums report the frequently selected gene sets across different datasets and the number of gene sets used by each dataset on the average, respectively.

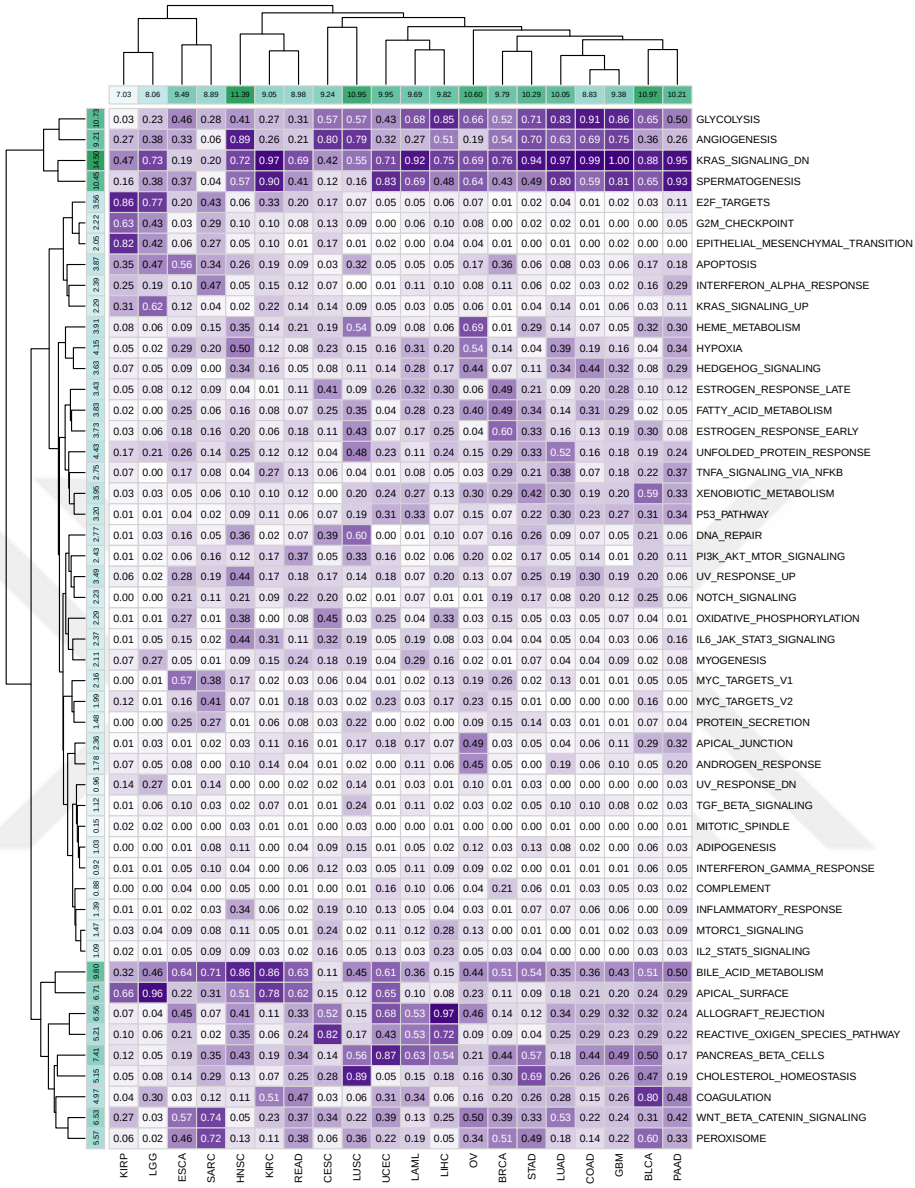

Figure D.4: The selection frequencies of 50 gene sets over 100 replications by Path2CSurv algorithm when the number of clusters is set to seven. The rows and the columns are clustered using the hierarchical clustering algorithm with the Euclidean distance and complete linkage functions. The row and the column sums report the frequently selected gene sets across different datasets and the number of gene sets used by each dataset on the average, respectively.

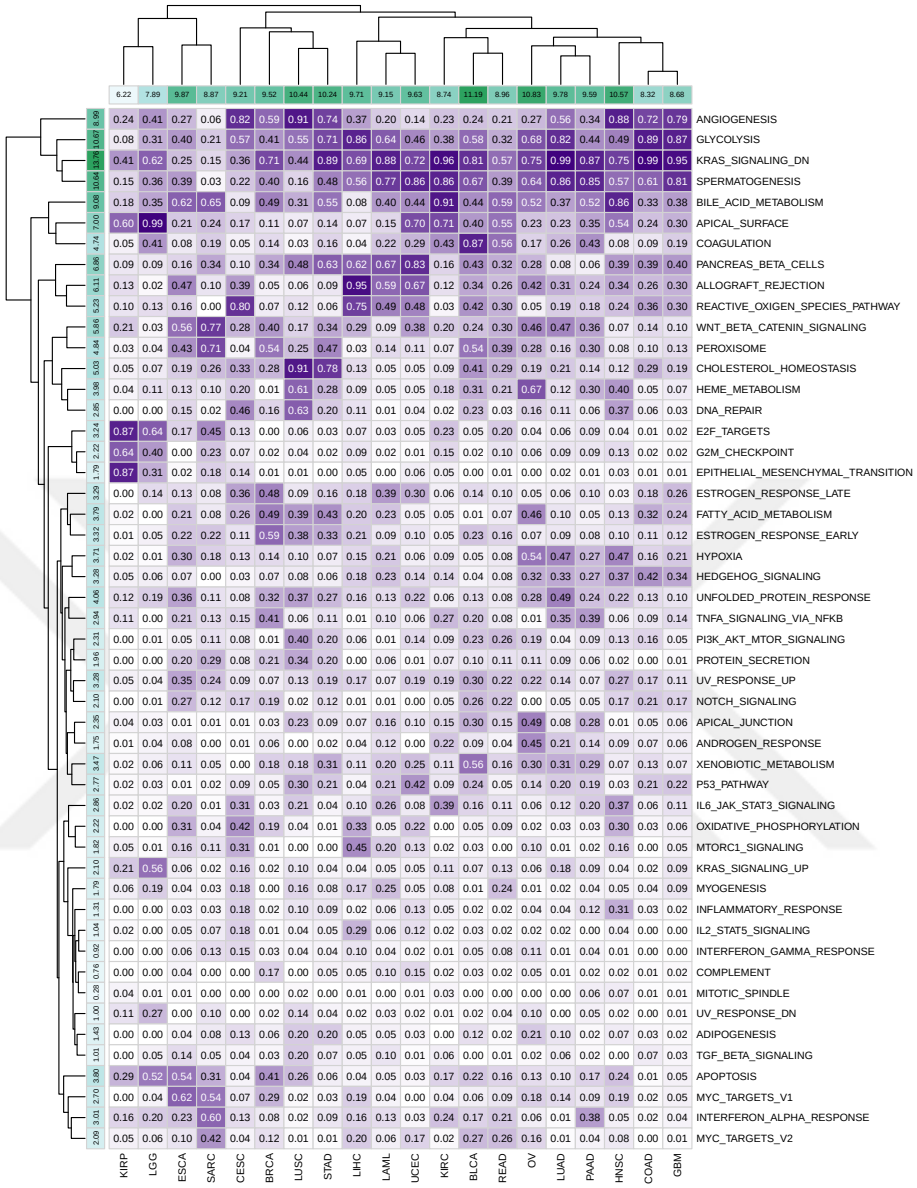

Figure D.5: The selection frequencies of 50 gene sets over 100 replications by Path2CSurv algorithm when the number of clusters is set to eight. The rows and the columns are clustered using the hierarchical clustering algorithm with the Euclidean distance and complete linkage functions. The row and the column sums report the frequently selected gene sets across different datasets and the number of gene sets used by each dataset on the average, respectively.

Appendix E

CLUSTER STRUCTURES OBTAINED BY PATH2CSURV ALGORITHM

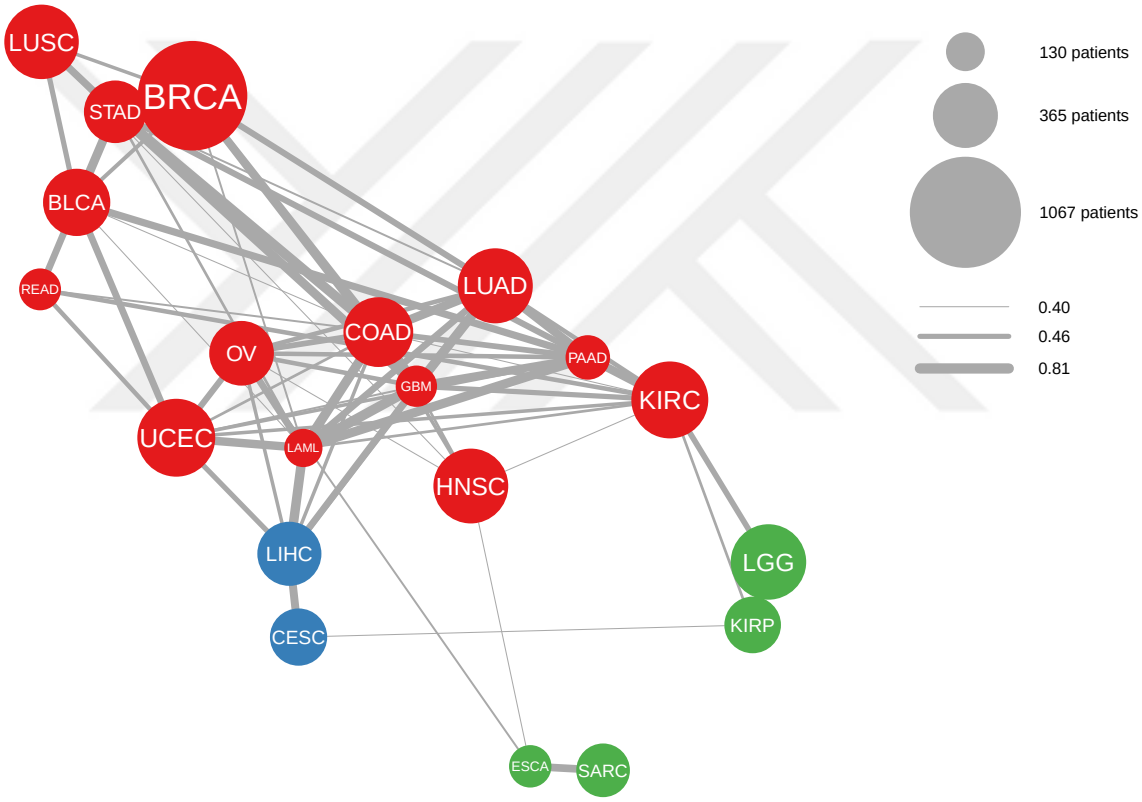

Figure E.1: Cluster structure of 20 TCGA datasets obtained by Path2CSurv algorithm when the number of clusters is set to three. Each node represents a cancer, and the edges between nodes show the assignment frequencies of connected pairs to the same cluster over 100 replications. Node and edge sizes are proportional to the number of patients in the cohorts and the assignment frequencies, respectively. The edges that have the frequencies below 0.40 were not shown in the figure. Each color refers to a different cluster.

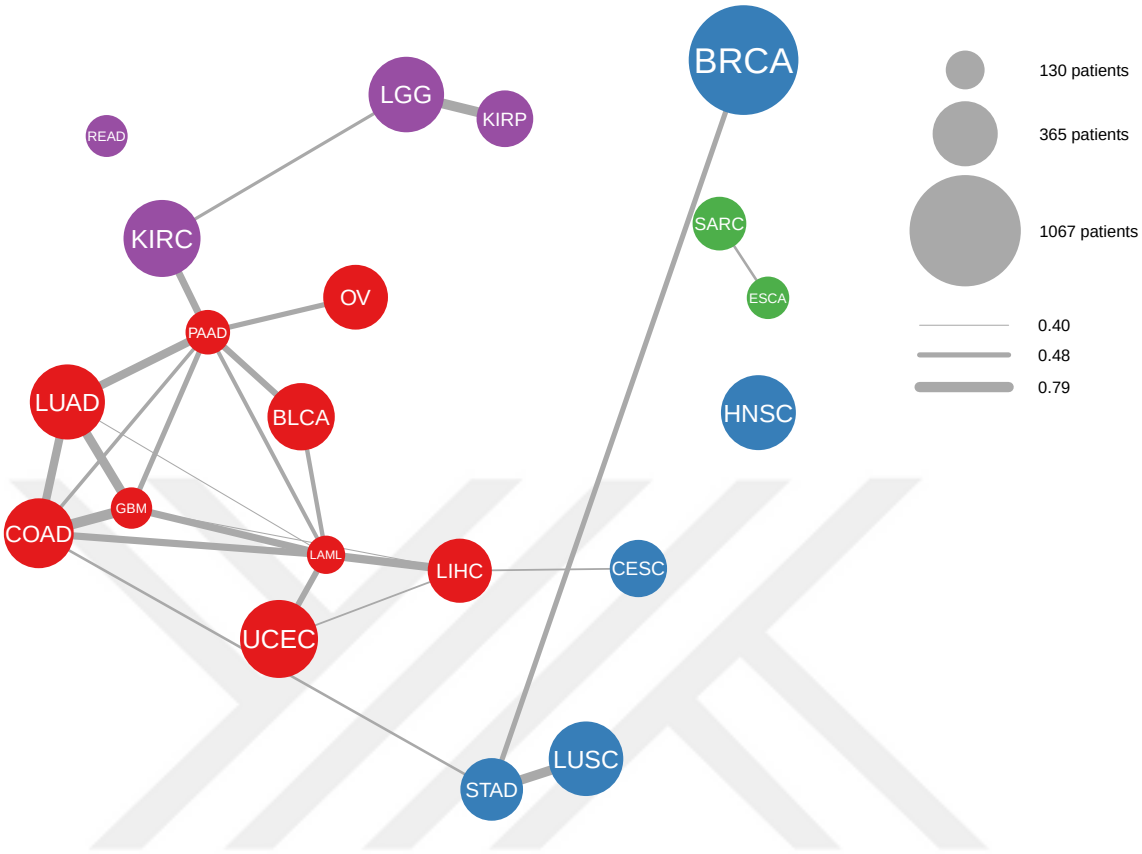

Figure E.2: Cluster structure of 20 TCGA datasets obtained by Path2CSurv algorithm when the number of clusters is set to four. Each node represents a cancer, and the edges between nodes show the assignment frequencies of connected pairs to the same cluster over 100 replications. Node and edge sizes are proportional to the number of patients in the cohorts and the assignment frequencies, respectively. The edges that have the frequencies below 0.40 were not shown in the figure. Each color refers to a different cluster.

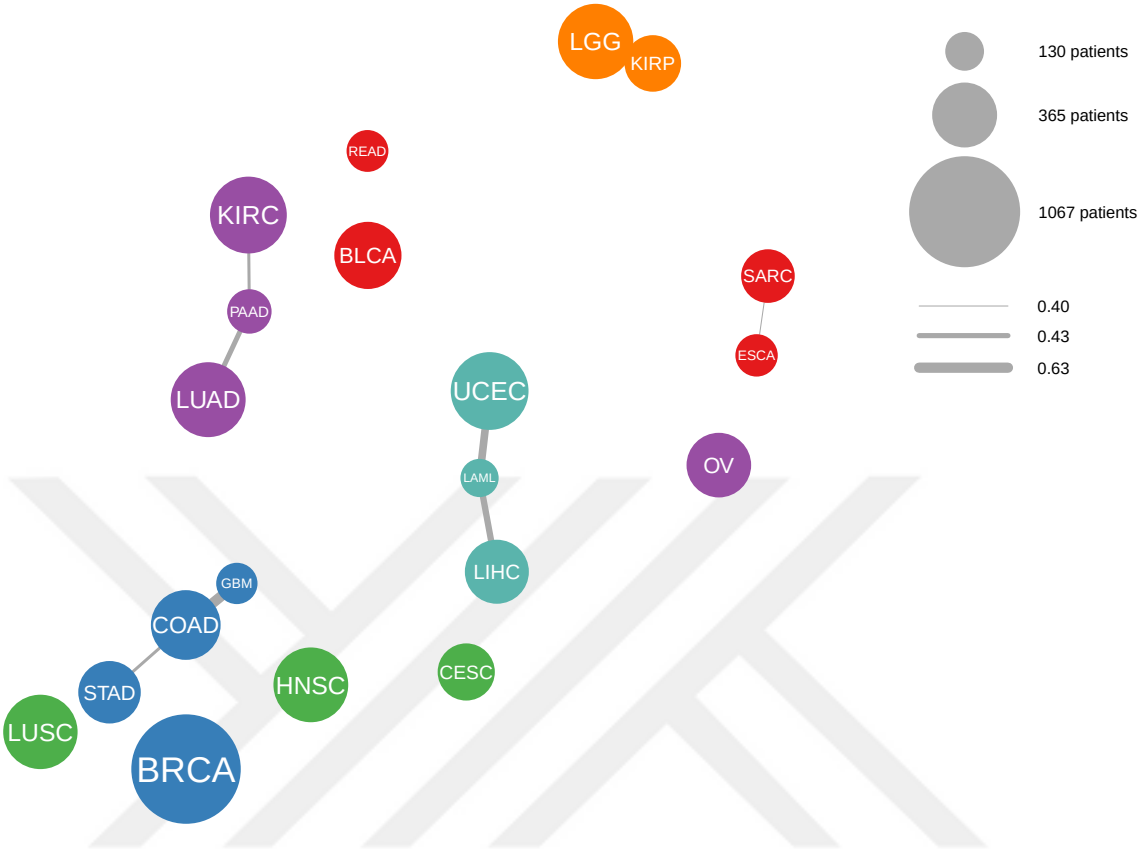

Figure E.3: Cluster structure of 20 TCGA datasets obtained by Path2CSurv algorithm when the number of clusters is set to six. Each node represents a cancer, and the edges between nodes show the assignment frequencies of connected pairs to the same cluster over 100 replications. Node and edge sizes are proportional to the number of patients in the cohorts and the assignment frequencies, respectively. The edges that have the frequencies below 0.40 were not shown in the figure. Each color refers to a different cluster.

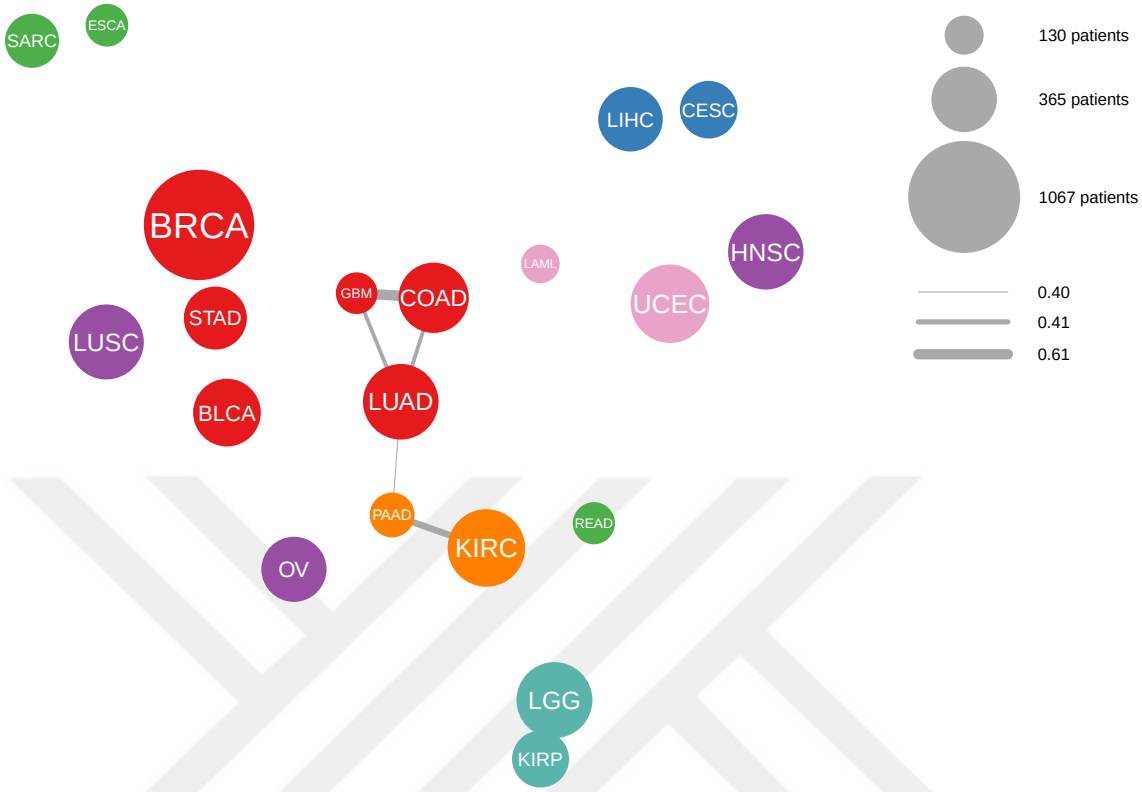

Figure E.4: Cluster structure of 20 TCGA datasets obtained by Path2CSurv algorithm when the number of clusters is set to seven. Each node represents a cancer, and the edges between nodes show the assignment frequencies of connected pairs to the same cluster over 100 replications. Node and edge sizes are proportional to the number of patients in the cohorts and the assignment frequencies, respectively. The edges that have the frequencies below 0.40 were not shown in the figure. Each color refers to a different cluster.

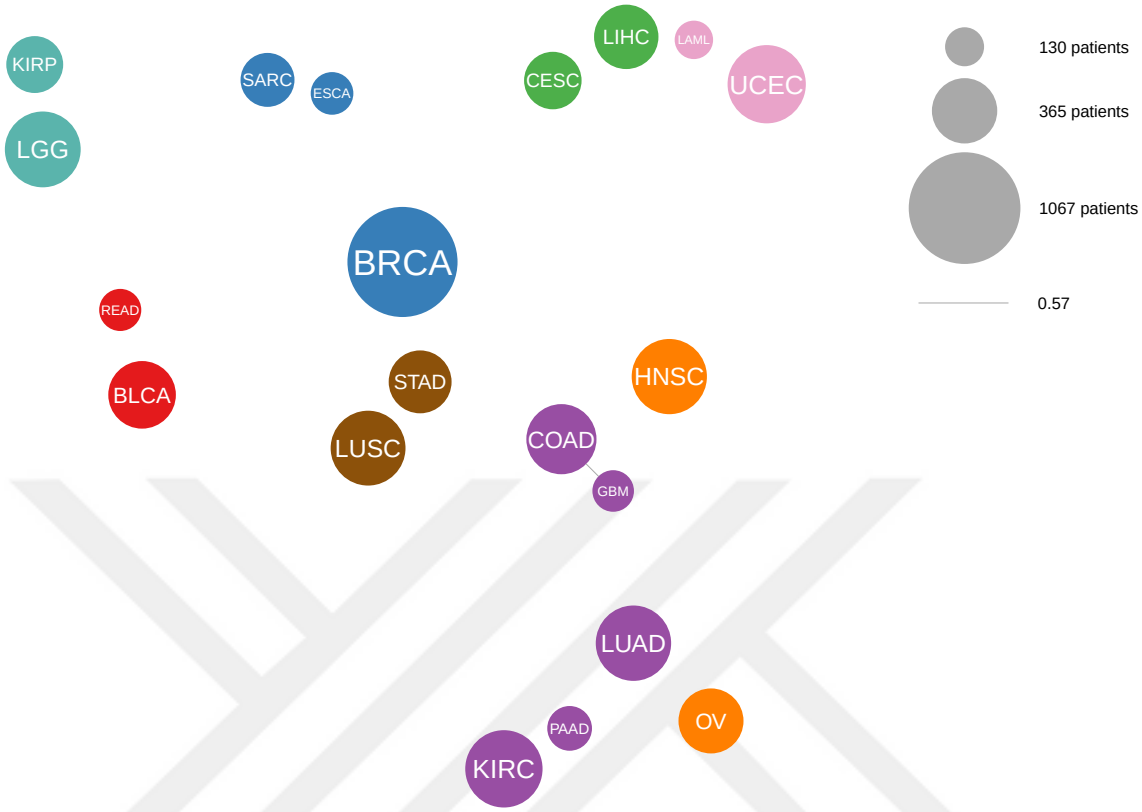

Figure E.5: Cluster structure of 20 TCGA datasets obtained by Path2CSurv algorithm when the number of clusters is set to eight. Each node represents a cancer, and the edges between nodes show the assignment frequencies of connected pairs to the same cluster over 100 replications. Node and edge sizes are proportional to the number of patients in the cohorts and the assignment frequencies, respectively. The edges that have the frequencies below 0.40 were not shown in the figure. Each color refers to a different cluster.

BIBLIOGRAPHY

Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

Anaya, J., Reon, B., Chen, W.-M., Bekiranov, S., and Dutta, A. (2016). A pan-cancer analysis of prognostic genes. *PeerJ*, 3:e1499.

Bakker, B., Heskes, T., Neijt, J., and Kappen, B. (2004). Improving Cox survival analysis with a neural-Bayesian approach. *Statistics in Medicine*, 23:2989–3012.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

Caruana, R. (1997). Multitask learning. *Machine Learning*, 28:41–75.

Choi, W., Porten, S., Kim, S., Willis, D., Plimack, E., et al. (2014). Identification of distinct basal and luminal subtypes of muscle-invasive bladder cancer with different sensitivities to frontline chemotherapy. *Cancer Cell*, 25:152–165.

Clark, T. G., Bradburn, M. J., Love, S. B., and Altman, D. G. (2003). Survival analysis part i: basic concepts and first analyses. *British Journal of Cancer*, 89(2):232–238.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20:273–297.

Costello, J. C., Heiser, L. M., Georgii, E., Gönen, M., Menden, M. P., et al. (2014). A community effort to assess and improve drug sensitivity prediction algorithms. *Nature Biotechnology*, 32:1202–1212.

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 34:187–220.

Cox, D. R. and Oakes, D. (1984). *Analysis of Survival Data*. Chapman and Hall, London.

Damrauer, J., Hoadley, K., Chism, D., Fan, C., Tiganelli, C., et al. (2014). Intrinsic subtypes of high-grade bladder cancer reflect the hallmarks of breast cancer biology. *Proceedings of the National Academy of Sciences*, 111:3110–3115.

Dereli, O., Oğuz, C., and Gönen, M. (2019a). A multitask multiple kernel learning algorithm for survival analysis with application to cancer biology. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 1576–1585. PMLR.

Dereli, O., Oğuz, C., and Gönen, M. (2019b). Path2Surv: Pathway/gene set-based survival analysis using multiple kernel learning. *Bioinformatics*, 35(24):5137–5145.

Evers, L. and Messow, C. M. (2008). Sparse kernel methods for high-dimensional survival data. *Bioinformatics*, 24:1632–1638.

Gönen, M. and Alpaydın, E. (2011). Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268.

Gönen, M., Weir, B. A., Cowley, G. S., Vazquez, F., Guan, Y. F., et al. (2017). A community challenge for inferring genetic predictors of gene essentialities through analysis of a functional screen of cancer cell lines. *Cell Systems*, 5:485–497.

Hoadley, K. A., Yau, C., Hinoue, T., Wolf, D. M., Lazar, A. J., Drill, E., Shen, R., Taylor, A. M., Cherniack, A. D., Thorsson, V., et al. (2018). Cell-of-origin patterns dominate the molecular classification of 10,000 tumors from 33 types of cancer. *Cell*, 173(2):291–304.

Hoadley, K. A., Yau, C., Wolf, D. M., Cherniack, A. D., Tamborero, D., et al. (2014). Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin. *Cell*, 158:929–944.

IBM (2017). *ILOG CPLEX Interactive Optimizer. Version 12.7.1.0.*

Ishwaran, H. and Kogalur, U. B. (2018). *randomForestSRC: Random Forests for Survival, Regression, and Classification (RF-SRC) R package version 2.6.0.*

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., and Lauer, M. S. (2008). Random survival forests. *The Annals of Applied Statistics*, 2:841–860.

Khan, F. M. and Zubek, V. B. (2008). Support vector regression for censored data (SVRc): A novel tool for survival analysis. In *Proceedings of the 8th IEEE International Conference on Data Mining*, pages 863–868.

Khirade, M. F., Lal, G., and Bapat, S. A. (2015). Derivation of a fifteen gene prognostic panel for six cancers. *Scientific Reports*, 5:13248.

Kiaee, F., Sheikhzadeh, H., and Mahabadi, S. E. (2016). Relevance vector machine for survival analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 27:648–660.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Lawrence, M., Stojanov, P., Mermel, C., Robinson, J., Garraway, L., et al. (2014). Discovery and saturation analysis of cancer genes across 21 tumour types. *Nature*, 505:495–501.

Li, Y., Wang, J., Ye, J., and Reddy, C. K. (2016). A multi-task learning formulation for survival analysis. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1715–1724.

Liberzon, A., Birger, C., Thorvaldsdottir, H., Ghandi, M., Mesirov, J. P., and Tamayo, P. (2015). The Molecular Signatures Database (MSigDB) hallmark gene set collection. *Cell Systems*, 1:417–425.

Maaten, L. V. D. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

Mogensen, U. B. and Gerds, T. A. (2013). A random forest approach for competing risks based on pseudo-values. *Statistics in Medicine*, 32:3102–3114.

Pang, H., Datta, D., and Zhao, H. (2010). Pathway analysis using random forests with bivariate node-split for survival outcomes. *Bioinformatics*, 26:250–258.

Pang, H., George, S. L., Hui, K., and Tong, T. (2012). Gene selection using iterative feature elimination random forests for survival outcomes. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9:1422–1431.

Pang, H., Hauser, M., and Minvielle, S. (2011). Pathway-based identification of SNPs predictive of survival. *European Journal of Human Genetics*, 19:704–709.

Pappa, K. I., Polyzos, A., Jacob-Hirsch, J., Amariglio, N., Vlachos, G. D., et al. (2015). Profiling of discrete gynecological cancers reveals novel transcriptional modules and common features shared by other cancer types and embryonic stem cells. *PLoS One*, 10:e0142229.

Schaefer, C. F., Anthony, K., Krupa, S., Buchoff, J., Day, M., et al. (2009). PID: The Pathway Interaction Database. *Nucleic Acids Research*, 37:D674–D679.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA.

Shivaswamy, P. K., Chu, W., and Jansche, M. (2007). A support vector approach to censored targets. In *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 655–660.

Sinnott, J. A. and Cai, T. (2018). Pathway aggregation for survival prediction via multiple kernel learning. *Statistics in Medicine*, 37(16):2501–2515.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550.

The Cancer Genome Atlas Research Network, Weinstein, J. N., Collisson, E. A., Mills, G. B., Shaw, K. R. M., et al. (2013). The Cancer Genome Atlas Pan-Cancer analysis project. *Nature Genetics*, 45:1113–1120.

Van Belle, V., Pelckmans, K., Van Huffel, S., and Suykens, J. A. (2011a). Improved performance on high-dimensional survival data by application of Survival-SVM. *Bioinformatics*, 27:87–94.

Van Belle, V., Pelckmans, K., Van Huffel, S., and Suykens, J. A. (2011b). Support vector methods for survival analysis: A comparison between ranking and regression approaches. *Artificial Intelligence in Medicine*, 53:107–118.

Wan, Q., Dingerdissen, H., Fan, Y., Gulzar, N., Pan, Y., et al. (2015). BioXpress: An integrated RNA-seq-derived gene expression database for pan-cancer analysis. *Database*.

Wang, L., Li, Y., Zhou, J., Zhu, D., and Ye, J. (2017). Multi-task survival analysis. In *Proceedings of the 17th IEEE International Conference on Data Mining*, pages 485–494.

Wang, Y., Chen, T., and Zeng, D. (2016). Support vector hazards machine: A counting process framework for learning risk scores for censored outcomes. *Journal of Machine Learning Research*, 17:1–37.

Xu, Z., Jin, R., Yang, H., King, I., and Lyu, M. (2010). Simple and efficient multiple kernel learning by group Lasso. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1175–1182.

Yang, Y., Han, L., Yuan, Y., Li, J., Hei, N., et al. (2014). Gene co-expression network analysis reveals common system-level properties of prognostic genes across cancer types. *Nature Communications*, 5:3231.

Yousefi, S., Amrollahi, F., Amgad, M., Dong, C., Lewis, J. E., et al. (2017). Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. *Scientific Reports*, 7:11707.

Yuan, Y., Van Allen, E. M., Omberg, L., Wagle, N., Amin-Mansour, A., et al. (2014). Assessing the clinical utility of cancer genomic and proteomic data across tumor types. *Nature Biotechnology*, 32:644–652.

Zhang, X., Li, Y., Akinyemiju, T., Ojesina, A. I., Buckhaults, P., et al. (2017). Pathway-structured predictive model for cancer survival prediction: A two-stage approach. *Genetics*, 205:89–100.