Advancing Cancer Staging: Leveraging Ordered Information from Multiple Ordinal Risk Factors

By

Yingzhou Liu

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

(Biomedical Data Science)

at the

UNIVERSITY OF WISCONSIN-MADISON

2023

Date of final oral examination: 08/30/2023

The dissertation is approved by the following members of the Final Oral Committee:

Menggang Yu, Professor, Biostatistics and Medical Informatics

Richard J. Chappell, Professor, Biostatistics and Medical Informatics

Jiwei Zhao, Associate Professor, Biostatistics and Medical Informatics

Sameer Deshpande, Assistant Professor, Statistics

Maureen A. Smith, Professor, Population Health Sciences

Advancing Cancer Staging: Leveraging Ordered Information from Multiple Ordinal Risk Factors

Yingzhou Liu

Under the supervision of Professor Menggang Yu

At the University of Wisconsin-Madison

Abstract

Cancer staging is a crucial process that determines the severity of an individual's cancer based on specific risk factors and clinical outcomes, such as time-to-event outcomes or the presence of a disease. This process involves classifying a heterogeneous set of cancer patients into several homogeneous groups. Accurately classifying the cancer stages helps doctors identify patients for clinical trials, understand the disease's severity and prognosis, and facilitate clinical decision-making on therapy and surveillance.

Tree methods have emerged as promising tools for cancer staging due to their ease of interpretation and ability to handle complex datasets with minimal assumptions [Bre+17; LWC13; Lin+16]. However, integrating multiple risk factors into cancer staging using tree methods presents several challenges. First, it is unclear how to leverage the ordering indicated by ordinal risk factors. Second, with a high number of categories defined by risk factors, it remains unknown whether patients in each category have a distinct prognosis. If not, it is unclear how to combine them into one stage. Finally, allowing a general grouping pattern is challenging, as most approaches have restrictions on the patterns of groupings. For instance, the classification and regression tree (CART) method only permits straight-line groupings on a partially ordered two-way grid.

To address the limitations of tree methods with ordinal variables, we introduce a new method: Ordering Partially Ordered Set Elements by Recursive Amalgamation (OPERA)[Wan20]. This approach utilizes ordered information and accommodates gen-

eral grouping patterns. OPERA, combined with pruning, demonstrates improved performance compared to traditional tree methods without pruning. This data-driven tool simplifies staging by condensing multiple groups into a single stratum based on a distinct prognosis, enhancing ease of use and interpretation. A well-trained tool can also accurately classify cancer stage and predict clinical outcomes. Beyond cancer staging, this method has implications for clustering in healthcare, aiding the identification of homogeneous patients for clinical trials and resource prioritization.

Acknowledgments

I would like to express my gratitude to my Ph.D. advisor, Prof. Menggang Yu, for his guidance throughout this journey. I vividly recall the excitement when Prof. Yu shared his latest work on cancer staging during our first Zoom meeting, along with the potential for further accomplishments. Looking back, I'm grateful that my research can build upon this amazing work and complete the story. I've always felt blessed to have Prof. Yu as my advisor, not only due to his expertise in this field but also his incredible patience with students and his constant focus on their best interests. He also demonstrates how to think as an independent researcher and communicate as a reliable collaborator. These skills will undoubtedly shape my career trajectory. He has shown great kindness and care, even driving us 6 hours to a conference where we learned a lot about ongoing clinical trial research. I also want to thank Prof. Richard Chappell, who taught me about generalized linear models and survival analysis in his class. He has been extremely supportive of my research and has always kept me aware of the underlying limitations. I also enjoy our conversations and learning from him. I appreciate his broad knowledge and sense of humor. I'm grateful to have Prof. Maureen Smith on my committee. Her perspective on clinical research has been valuable for someone like me who mainly focuses on methodology. Her input has helped me consider how important it is for our methods to address clinical questions and keep the bigger picture in mind. I'm always impressed by the depth of thought that goes into a research question and how much remains open for people to solve. I'm also thankful to Prof. Sameer Deshpande, who generously shared his related research work with us. I'm always amazed by how Prof. Deshpande can be academically successful and still be open to sharing his ongoing research with the community. Thank you as well to Prof. Jiwei Zhao, who has been very encouraging regarding my research. I appreciate Prof. Zhao's instruction and advice, which have significantly helped me refine my research.

I also want to express my gratitude to my collaborators, especially Prof. Meghan

Brennan. I've enjoyed working with Meghan, not only for the experience of solving real-world problems but also for her inspiration and encouragement with my analyses. Meghan has been incredibly considerate, always checking to ensure I have enough time to get things done without overstressing myself. I'm also grateful to Prof. Yin Li and Prof. Daniel Pimentel-Alarcón, with whom I completed two rotation research projects. Prof. Li's expertise in his field and detailed guidance have been amazing. I've learned a lot from Prof. Li about how to communicate with collaborators and leverage everyone's efforts to achieve the same goal. Additionally, working with Daniel has been enjoyable, not only because he always simplifies complicated concepts and methods but also because of our shared broader interests beyond academic research. Thank you also to Prof. Xiaofei Wang and Prof. Ricardo Henao, with whom I worked during my time at Duke, where I developed a range of well-rounded research skills. I would also like to thank Dr. Tianjie Wang, who originally came up with OPERA and laid a solid foundation for my work. I am also grateful to my internship supervisor, Dr. Jie Cheng, at Takeda, who has been extremely supportive of my work and inspiring me with new ideas. I have learned a lot about working in the industry throughout my internship with Jie.

I also thank my dear family and friends, without whom I would not have been able to overcome all the challenges in graduate school and see them through. In the past six years of studying abroad, I have met so many talented people who are not only successful in their studies and careers but also deeply nice and kind individuals. They have been real examples to learn from on many different levels. Finally, I want to thank my family—my parents, my grandparents, my aunts, my uncle, my cousin, my cousin-in-law, and my little niece. I always believe that no matter what happens, family will always stay with you, and that is something that will never change. My family is just another example of it. A six-year-long graduate study has been not only a challenge for me but also a challenge for my family. Especially with the pandemic, we could not see each other in person as before. Throughout all the difficult times, I am really grateful that my family

always believes in me, supports me, and loves me unconditionally. Especially, I want to thank my baba [1], yeye and nainai [2] for raising me, educating me, and promising me a happy childhood. I am grateful that my family always values education and is always willing to sacrifice their own needs to ensure that I will get the best education since they believe education is the key to change one's life. I want to thank my gugu [3] who has raised me, taught me, and protected me in the past twenty-eight years. I feel so blessed to have my gugu, who always puts my priorities over hers and gives her best to me. I believe her unconditional love is always the most precious treasure in my life, and I always love my gugu deeply—nothing can change that.

[1] dad in Mandarin
[2] paternal grandparents in Mandarin
[3] my father's younger sister in Mandarin

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Cancer staging plays a crucial role in medicine by identifying homogeneous patient groups for clinical trials, aiding in treatment decisions, helping to understand the disease prognosis, and facilitating communication and collaboration among medical professionals. The TNM staging system [Den52], which classifies cancers based on the extent of the tumor (T), extent of spread to the lymph nodes (N), and presence of metastasis (M), has become a wildly used benchmark in the medical community. This system has historically been associated with outcome measures like overall survival (OS) and relapse-free survival (RFS). While the TNM staging system can provide accurate predictions of outcome for heterogeneous patients groups, incorporating additional biologic determinants that recognize the intrinsic tumor biology is necessary for a more personalized approach to patient classification [Giu+17], especially for different biologic subtypes of cancers that express different biomarkers at an individual level.

In the revision of the 8th edition of the TNM classification of the AJCC, the breast cancer staging system incorporated tumor grade, proliferation rate, estrogen receptor (ER) and progesterone receptor (PR) expression, human epidermal growth factor 2 (HER2) expression, and gene expression. This update provides a more flexible and precise platform for prognostic classification based on both anatomic factors and biomarkers

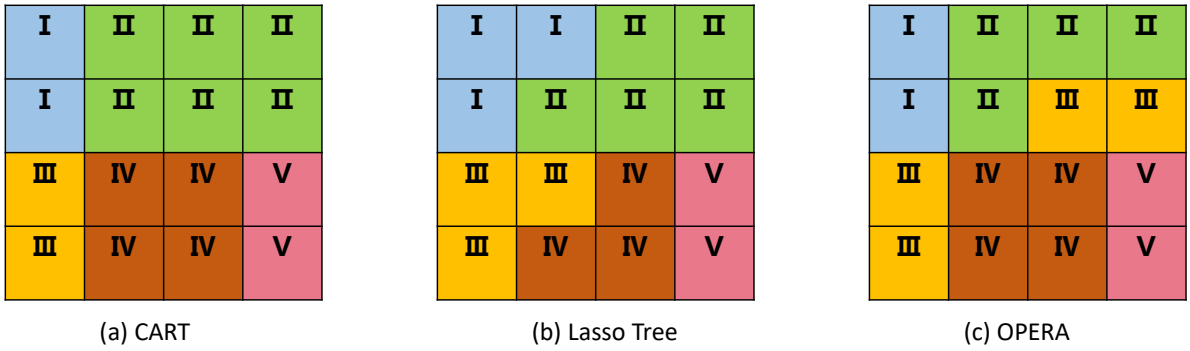
[Giu+17]. In addition to ER, PR and HER2, which can be analyzed through microarray and hierarchical clustering analysis, gene expression profiling has identified five main intrinsic or molecular subtypes of breast cancer. These subtypes, which are correlated with immunohistochemical (IHC) biomarkers, include luminal A, normal-like, luminal B, HER2-enriched and triple-negative or basal-like breast cancer in order of prognosis from best to worst [Per+00]. Studies have demonstrated that surrogate classification based on IHC biomarkers and tumor grade leads to improved separation in survival outcomes when compared to using the eight ER/PR/HER2 subtypes [PC14].

However, using a large number of risk factors to create overly refined categorizations in the staging system makes the system difficult to use. Moreover, increasing the number of risk categories has not been accompanied by increased ease of use or better prognostic ability, so that parsimony should be considered to counterbalance the instinct to increase the number of categories in future revisions of staging systems [GW10]. The challenge now is to merge a high number of categories jointly defined by multiple risk factors into a few strata while preserving clinically distinct prognosis within each stratum. For instance, the PAM50 molecular classifier is a 50-gene panel that can accurately distinguish the intrinsic subtypes of breast cancer [Par+09]. By combining PAM50 (A: luminal A, normal-like, luminal B, HER2-enriched and basal-like), tumor stage (B: I, II, III, IV) and neoplasm histologic grade (C: 1, 2, 3), we can create an $A \times B \times C$ cube with $5 \times 4 \times 3 = 60$ different categories. Therefore, a data-driven algorithm is needed to determine which categories have a distinct prognosis and to collapse certain categories into one stratum. It is important to maintain the ordering of ordinal risk factors and the difference in prognosis among strata.

This problem can potentially be solved by tree-based algorithms such as the classification and regression tree (CART) method [Bre+17; Loh14]. The CART method is suitable for either time-to-event outcome or binary outcome. It provides a tree structure that can be easily converted to a set of rules to facilitate its clinical utility, and is not

influenced by scale or specific assumptions. However, it is unclear how to incorporate the intrinsic ordering embedded in risk factors into the heuristics of the CART method. When the ordering is adopted, CART cannot be generalized to all patterns of groupings. At each split, CART must make a complete separation for each risk factor. For instance, as shown in Figure 1.1(a), in the $A \times B$ table where both A and B have four levels, CART splits thoroughly along all columns or rows conditional on the existing splits [LWC13], since CART treats ordinal variables in the same way as continuous variables. Once a cut-off value is selected, CART continues to explore each split and each leaf node corresponds to a different level of stage. Thus, CART is unable to generate a three-category grouping where categories lie in different columns and rows like III and IV in Figure 1.1(b), or a non-neighbouring grouping like III in Figure 1.1(c). Hence lasso tree for cancer staging with survival data [LWC13] and penalized logistic regression [Lin+16] with binary outcome such as mortality or the presence of disease, were proposed separately to enforce more general grouping and to select the optimal grouping by introducing partial ordering constraints. Although these two methods deal with different type of outcomes, they both use an $L1$ penalized regression method to shrink the difference of neighbouring coefficients towards zero instead of the coefficients themselves and to encourage sparsity in the grouping of the categories. Both of them also take into account partial ordering constraints for multiple categories of multiple risk factors by using partial ordering constraints, with no pre-specified number of stages. Both of these two methods can create triangular and rectangular grouping patterns as shown in Figure 1.1(b) but fail to generate non-neighbouring patterns like III in Figure 1.1(c) as sparsity is only forced on neighbouring coefficients.

Therefore, a new method called ordering partially ordered set (Poset) elements by recursive amalgamation (OPERA) was proposed to overcome the limitations of previous methods [Wan20]. OPERA treats the partially ordered two-way grid based on prognosis as a poset and shrinks all the coefficients to a reference level using $L1$ penalty during each



(a) CART (b) Lasso Tree (c) OPERA

Figure 1.1: Different grouping patterns using CART, lasso tree and OPERA

recursive step, allowing it to generate any grouping pattern, including non-neighbouring patterns like III in Figure 1.1(c). Additionally, non-risk-factor covariates can be adjusted to improve modelling. Although OPERA has shown superior performance compared to lasso tree and CART in risk stratification, it has only been applied to the cancer staging problem with two ordinal risk factors and survival outcome. Moreover, how to prune the initial result in a bottom-up way to improve accuracy and ensure a distinct prognosis for each stage by collapsing some categories and reducing the total number of stages remains an open question.

This thesis extends OPERA to binary outcomes and scales it up to handle multiple risk factors. Additionally, a pruning algorithm is proposed to improve the accuracy of cancer staging and generate better separations of stages. Furthermore, pruning can deal with continuous risk factors by categorizing them and merging them into fewer categories based on the same grouping patterns. Simulation studies and real-world data both demonstrate very promising results.

Chapter 2 of this thesis presents a generalized version of OPERA that can handle binary outcomes with multiple risk factors. In addition to the initial OPERA result, we introduce a pruning step that is similar to tree methods. This step combines overly refined stages and improves accuracy. We also extend the lasso tree method to binary outcomes with multiple risk factors and evaluate our approach through a series of simulation studies.

In Chapter 3, we apply OPERA, along with the pruning step, to time-to-event out-

comes with multiple risk factors. Additionally, we extend the lasso tree method to survival outcomes with multiple risk factors, which serves as a comparison with our approach.

Chapter 4 shows how OPERA, with the pruning step, can handle continuous risk factors by categorizing them into ordinal categories and merging them based on the same staging patterns.

Chapter 5 evaluates the performance of OPERA on several real datasets, including an advanced colorectal neoplasia study and a few cancer studies, such as breast cancer, colorectal cancer, lung cancer, and prostate cancer.

Chapter 6 presents an R package that can implement cancer staging methods, including the lasso tree method and OPERA with the pruning step as an available option.

Finally, in Chapter 7, we apply OPERA beyond cancer staging and use it as a clustering method to identify homogeneous patients with advanced illnesses. This approach helps prioritize clinical resources for patients in higher stages while also simplifying the advanced illness trigger groupings designed by clinicians. This is accomplished by taking into account more clinical and demographic risk factors.

Chapter 2

Binary Outcome

This chapter provides a comprehensive description of OPERA and its application to binary outcomes with multiple risk factors. Additionally, we introduce a pruning step that improves a potentially overly refined staging system. Taking inspiration from the topological sorting problem, this chapter begins by introducing important topological concepts that are used to describe OPERA.

2.1 Topological Concepts

Topological concepts and relevant notations are adopted from Brualdi (2010) [Bru10] and Garg (2015) [Gar15].

Definition 2.1.1. (Relation) Let X be a set. A relation on X is a subset R of the set $X \times X$ of ordered pairs of elements of X. We write $a\ R\ b$ (a is related to b), provided that the ordered pair (a, b) belongs to R; we also write $a \not R b$ whenever (a, b) is not in R (a is not related to b).

The following are special properties that a relation R on a set X may have:

- R is reflexive, provided that $x\ R\ x$ for all x in X.

- R is antisymmetric, provided that, for all x and y in X with $x \neq y$, whenever we have $x\ R\ y$, we also have $y \not R x$. Equivalently, for all x and y in X, $x\ R\ y$ and $y\ R\ x$ together imply that $x = y$.

- R is transitive, provided that, for all x, y, z in X, whenever we have $x\ R\ y$ and $y\ R\ z$, we also have $x\ R\ z$.

Example: The relation of "less than or equal" on a set of numbers, denoted by $\leq$, is a reflexive, antisymmetric, and transitive relation. ■

Definition 2.1.2. (Partial order) A partial order on a set X is a reflexive, antisymmetric, and transitive relation R. If a relation R is a partial order, we generally use the usual inequality symbol $\leq$ instead of R.

Definition 2.1.3. (Poset) A set X on which a partial order $\leq$ is defined is usually called a partially ordered set (or more simply, a poset) and denoted by $(X, \leq)$.

Example: The disease prognoses for breast cancer patients can be compared by using the $A \times B \times C$ system, which can be viewed as a partially ordered set (poset). A patient with luminal A breast cancer in the first neoplasm histologic grade and the first tumor stage ($A1B1C1$) is less likely to experience death than someone with basal-like breast cancer in the second neoplasm histologic grade and the third tumor stage ($A5B2C3$). The partial order is defined as *less or the same likely to experience death*, which can be denoted as $A1B1C1 \leq A5B2C3$. However, not all elements in the poset are comparable, such as $A4B2C3$ and $A5B1C3$. ■

Definition 2.1.4. (Total order) A partial order R on a set X is a total order, provided that every pair of elements of X is comparable.

Definition 2.1.5. (Cross Product of Posets) Given two posets $(P, \leq)$ and $(Q, \leq)$, the cross product forms a new poset denoted as $P \times Q$ is defined as $(P \times Q, \leq)$ where

$(p_1, q_1) \leq (p_2, q_2) \stackrel{\text{def}}{=} (p_1 \leq p_2) \cap (q_1 \leq q_2)$.

Example: The standard relation $\leq$ on a set of numbers is a total order. Another example in breast cancer staging can be a subset of the $A \times B \times C$ system, such as $\{A5B2C1, A5B2C2, A5B2C3, A5B2C4\}$. The $A \times B \times C$ staging system is the Cartesian product of the total ordered sets A, B and C and such a product set is a poset. ■

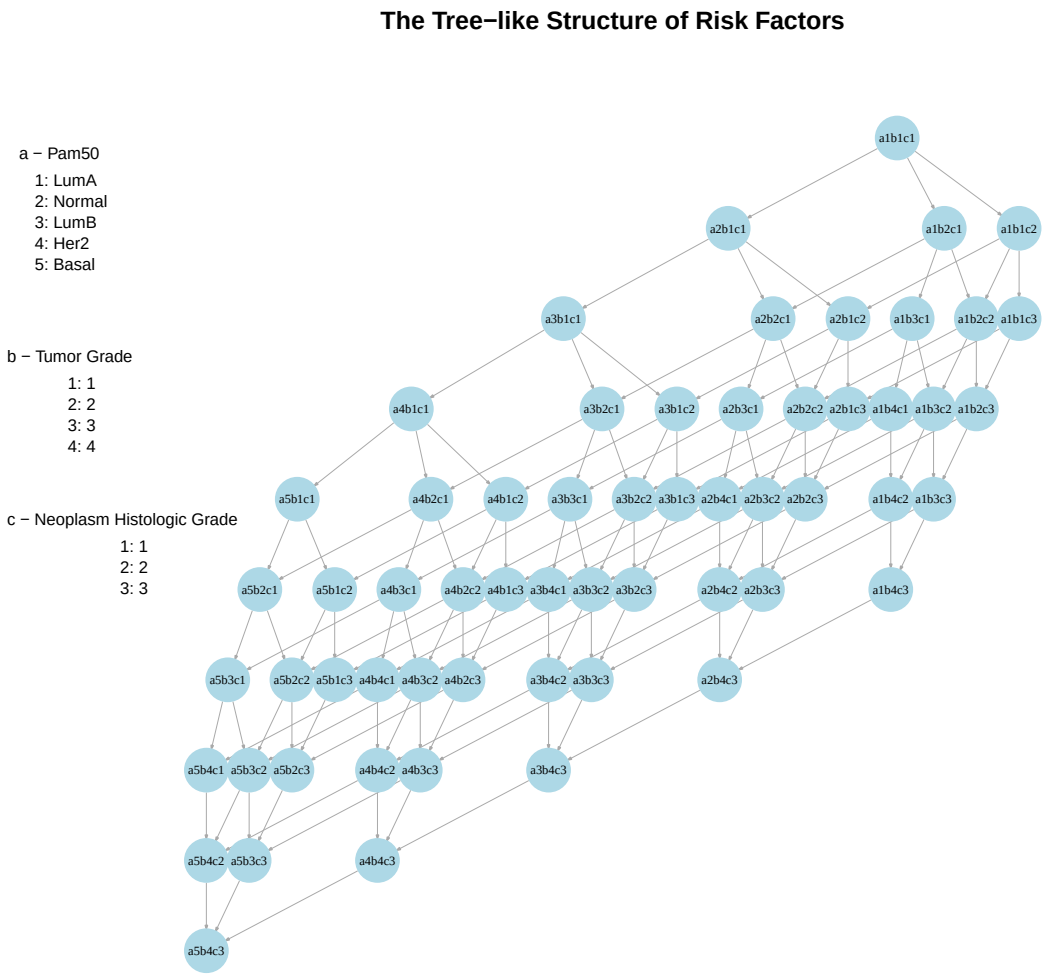


Figure 2.1: An example of a Hasse diagram

Definition 2.1.6. (Cover relation) Let a and b be in $(X, \leq)$. Then a is covered by b (also expressed as b covers a), denoted $a <_c b$, provided that $a < b$ and no element x can be squeezed between a and b; that is, there does not exist an element x such that both $a < x$ and $x < b$ hold.

Definition 2.1.7. (Hasse diagram) A diagram of a finite partially ordered set (X, $\leq$) is obtained by taking a point in the plane for each element of X, being careful to put the point for x below the point for y if $x <_c y$, and connecting x and y by a line segment if and only if x is covered by y.

Example: The Hasse diagram of a poset is a directed acyclic graph (DAG) that has at least one topological ordering. It consists of a sequence of vertices, and every edge is directed from one vertex to another, representing an ordering. For example, when considering three risk factors related to breast cancer (PAM50 (A), tumor stage (B), and neoplasm histologic grade (C)), each factor can be considered as a set with a total order. The relation on each set is based on *having no worse prognosis*. The Hasse diagram for the poset $(A \times B \times C, \leq)$ is shown in Figure 2.1, where each edge represents a cover relation indicating no worse prognosis between two elements. Each node represents an element in the poset $(A \times B \times C, \leq)$, and the arrow points from an element indicating no worse prognosis to an element indicating no worse prognosis. ■

Appendix Section A.1 contains more noteworthy properties regarding the Hasse diagram of a poset in the context of cancer staging problem.

Definition 2.1.8. (Mapping) We define a mapping ξ from the poset $(S, \leq)$ to the real numbers $\mathbb{R}$ as the regression coefficients of risk categories.

Definition 2.1.9. (Order-preserving condition) For any two elements, a and b, in a poset S, if a has no worse prognosis than b, denoted as $a \leq_S b$, then the corresponding coefficients $\xi(a)$ and $\xi(b)$ also follow this relation, i.e., $\xi(a) \leq \xi(b)$. In cancer staging,

the poset is formed from the cross product of total ordered sets, such as A, B, and C, where the relation among poset elements is defined as having no worse prognosis. Alternatively, this relation can be represented as a no-greater-than relation among the coefficients obtained from regression models. By enforcing sparsity, elements with the same coefficient can be classified into the same stage.

Given $(S, \leq) = S_1 \cup S_2 \cup \cdots \cup S_m$, where $S_1, S_2, \ldots, S_m$ represent disjoint cancer stages and $\forall i, j, S_i \cap S_j = \emptyset$,

- if $i < j$ and $\forall a \in S_i$ and $\forall b \in S_j$, then $\xi(a) < \xi(b)$.
- if $\forall a, b \in S_i$, then $\xi(a) = \xi(b)$.

In cancer staging, categories defined by multiple risk factors can be represented as a poset $(S, \leq)$. For each subject i, the observed data includes the outcome $y_i \in \mathbb{R}$, the risk category r_i, which is a $|S|$-dimensional one-hot vector indicating the category the subject falls into, the covariates $Z_i \in \mathbb{R}^p$, and the censoring status $\delta_i \in \{0, 1\}$ for survival outcomes. To model survival outcomes, we adopt the penalized Cox proportional hazard model, while for binary outcomes, we use the penalized logistic regression model.

Definition 2.1.10. (Down-set) Let $(S, \leq)$ be any poset. A subset D of S is known as a down-set or an order ideal if it satisfies the following condition: for any $y, z \in S$ such that $z \in D$ and $y \leq_S z$, we have $y \in D$.

2.2 OPERA without Pruning

The process of staging cancer can be seen as an iterative process of selecting a down-set S_k, where $k = 1, 2, \ldots, m$, and m is not pre-specified, from a residual set $\bigcup_{i=k}^{m} S_i = S - \bigcup_{i=1}^{k-1} S_i$. OPERA, which stands for *ordering poset elements by recursive amalgamation*, orders the poset elements by selecting those that do not cover any other elements, and recursively adding them to a down-set from the residual poset, resulting in the new stage,

as shown in Algorithm 1. This idea is similar to Kahn's algorithm for topological ordering [Kah62] on a directed acyclic graph, where the nodes without dependencies or incoming edges are selected and removed recursively.

Algorithm 1: OPERA

$k \leftarrow 0$; /* Initialization: No stages are found. */
$S_0 \leftarrow \emptyset$
while $\cup_{i=0}^{k} S_i \neq S$; /* Recursion: A down-set is identified from the residual set */
do
    $k \leftarrow k + 1$
    To satisfy the partial order constraints, the poset elements need to be ordered based on the partial ordering.
    Then, the elements that are close to the reference level can be amalgamated to form a new stratum S_k. This new stratum S_k should be a down-set of the residual poset $S - \cup_{i=0}^{k-1} S_i$.
    If an $L1$ penalty is included, the elements which are the nearest to the reference level can be chosen as the new stratum S_k. γs indicate how close each element is to the reference level. Therefore, the elements with zero γs, which are the nearest to the reference level, are selected to form the new stratum S_k.
end

Figure 2.2 illustrates the visualization of the OPERA algorithm on the Hasse diagram of an $A \times B$ table, where both A and B are four-level ordinal risk factors. An ordinal risk factor implies a monotonic relationship between the likelihood of experiencing an outcome and each level of the risk factor. That is, the higher the level goes, the more probable it is for a patient to experience an outcome. Each node in the diagram is labelled with a γ, which represents the difference in the mapped regression coefficient from the reference level.

The algorithm initially applies an $L1$ penalty to all γs and combines all elements with zero γs to create S_1. Subsequently, the elements in S_1 are assigned the same coefficient, and S_2 is identified using the same approach. This process is repeated until all elements are classified. After each iteration of finding a stage, the mapped coefficient of the reference level monotonically increases.

It is worth noting that OPERA has the capability to produce any grouping patterns, including non-neighboring grouping patterns, such as the one demonstrated in S_3 in Figure 2.2. This is due to the absence of restrictions during each iteration regarding which elements are combined to form a stage, as long as the partial ordering is maintained.

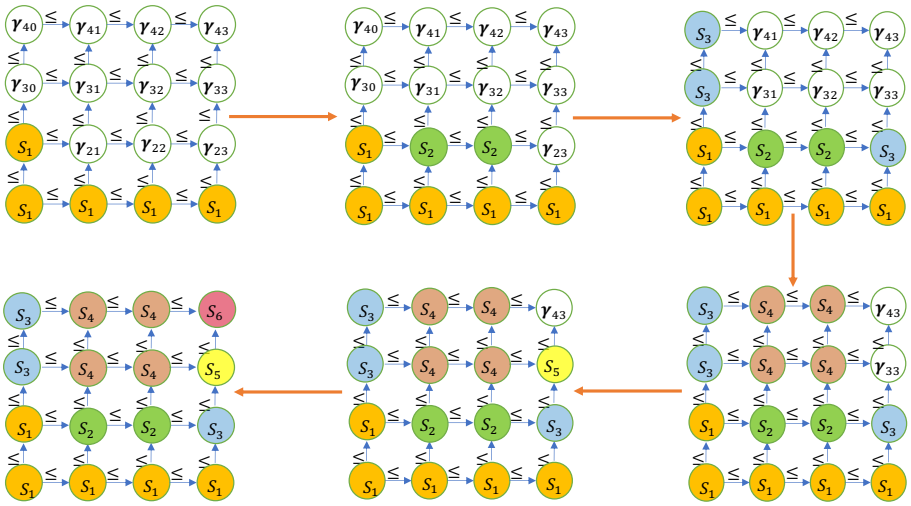


Figure 2.2: An example visualization of OPERA

OPERA is applicable to different types of outcomes, with the main difference being the likelihood function or objective function. In this chapter, we illustrate our method and algorithms using OPERA based on a binary outcome. However, details on how OPERA is used for survival outcomes can be found in the next chapter.

2.2.1 A Lasso-Type Modeling Procedure to Select a Down-Set

The primary objective of OPERA is to identify a down-set S_k from a residual set $S - \cup_{i=1}^{k-1} S_i$. To accomplish this, a lasso-type modeling procedure is employed. This strategy is based on two types of posets: posets with a unique minimum element and posets with no minimum element but multiple minimal elements.

Definition 2.2.1. (Minimum Element) An element l is deemed as the minimum element of a poset $(S, \leq)$ if $l \in S$ and $\forall y \in S : l \leq_S y$.

The uniqueness of a minimum element is ensured due to the antisymmetric property of the partial order. For instance, as demonstrated in Figure 2.1, the minimum element is $A1B1C1$.

When provided with data $(y_i, r_i, Z_i)_{i=1}^n$, where for each subject i, the observed data includes the binary outcome $y_i \in \{0, 1\}$, the risk category r_i that indicates the category the subject i falls into, and the covariates $Z_i \in \mathbb{R}^p$, the logarithm of the likelihood can be expressed as follows under a logistic regression model:

$$l(\xi, \alpha) = \sum_{i=1}^{n} [y_i(\xi(r_i) + Z_i^T \alpha) - \log(1 + \exp(\xi(r_i) + Z_i^T \alpha))] \tag{2.1}$$

Therefore, if the poset contains a unique minimum element, a fundamental approach for finding the target down-set involves selecting the minimum element l as the reference level and fusing other elements r into it with a coefficient $\xi(r)$ through penalization. When a fixed reference l is established, an intercept $\mu \in \mathbb{R}$ is introduced, where $\gamma(r) = \xi(r) - \mu$. If $\gamma(r) = 0$, then the element r can be amalgamated into the minimum element l with $\gamma(l) = 0$. Based on the equation $\xi(r) = \gamma(r) + \mu$, the lasso-type modeling for binary outcomes can be defined as follows:

$$\begin{aligned} \underset{\mu,\gamma,\alpha,\gamma(l)=0}{\arg\min} \; -l(\mu, \gamma, \alpha) + \lambda \sum_{r \in S} |\gamma(r)| = \underset{\mu,\gamma,\alpha,\gamma(l)=0}{\arg\min} \{ &- \sum_{i=1}^{n} [y_i(\mu + \gamma(r_i) + Z_i^T \alpha) - \\ &\log(1 + \exp(\mu + \gamma(r_i) + Z_i^T \alpha))] + \lambda \sum_{r \in S} \gamma(r) \} \end{aligned} \tag{2.2}$$

The partial order constraints for this problem are defined as follows: $\forall a, b \in (S, \leq)$, if $a \leq_S b$, then $\gamma(a) \leq \gamma(b)$, and γ is greater than or equal to 0. The parameter λ is used to control the level of sparsity. The stratum S_1 corresponds to the first stage of cancer,

and is defined as the set of elements $r \in (S, \leq)$ such that $\gamma(r) = 0$.

It is possible, however, for a poset to not have a minimum element.

Definition 2.2.2. (Minimal Element) A minimal element of a poset S is defined as an element x such that there exists no element $y \in S$ satisfying $y \leq_S x$.

Although the original poset may have only one minimum element, after several stratification steps, the residual poset can contain multiple minimal elements. In Figure 2.2, for instance, both γ_{30} and γ_{23} correspond to minimal elements after S_1 is identified. In cases where there are multiple minimal elements, our proposal is to follow the optimization approach of the basic method and use the minimum instead of arbitrarily selecting a minimal reference. To achieve this, we adopt a parameterization strategy[OV17] denoted by $\xi(r) = \gamma(r) + \mu$ for $\forall r \in S$, which involves $(|S| + 1 + p)$ parameters. Here, μ is an artificial parameter that does not correspond to any element, and generalizes the decomposition technique used for the minimum element.

Based on $\xi(r) = \gamma(r) + \mu$, the lasso-type modeling for binary outcome can be defined as follows:

$$\begin{aligned}\arg\min_{\mu,\gamma,\alpha} -l(\mu,\gamma,\alpha) + \sum_{r\in S}\lambda|\gamma(r)| = \arg\min_{\mu,\gamma,\alpha}\{-\sum_{i=1}^{n}[y_i(\mu+\gamma(r_i)+Z_i^T\alpha)- \\ \log(1+\exp(\mu+\gamma(r_i)+Z_i^T\alpha))] + \sum_{r\in S}\lambda\gamma(r)\}\end{aligned} \tag{2.3}$$

Similar to the case where there is a unique minimum element, the partial order constraints for this problem are defined as follows: $\forall a, b \in (S, \leq)$, if $a \leq_S b$, then $\gamma(a) \leq \gamma(b)$, and γ is greater than or equal to 0. The reference level is selected from the set of minimal elements. Furthermore, these constraints help eliminate the absolute value symbol since all elements are non-negative, which provides significant computational convenience.

If $S_1, S_2, \ldots, S_{k-1}$ have already been found, the down-set can be defined as $D_{k-1} = \bigcup_{i=1}^{k-1} S_i$, and the residual poset as $U_{k-1} = S - D_{k-1}$. To find a down-set S_k from a residual

poset U_{k-1}, one way is to identify all the elements in U_{k-1} that are minimal with respect to the partial order, and then add them to S_k. More formally, the lasso-type modeling for binary outcomes can be defined as follows:

$$\begin{aligned}\arg\min_{\mu,\gamma,\alpha} -l(\mu,\gamma,\alpha) + \sum_{u\in U_{k-1}} \lambda|\gamma(u)| = \arg\min_{\mu,\gamma,\alpha}\{-\sum_{i=1}^{n}[y_i(\mu+\gamma(r_i)+Z_i^T\alpha)- \\ \log(1+\exp(\mu+\gamma(r_i)+Z_i^T\alpha))] + \sum_{u\in U_{k-1}} \lambda\gamma(u)\}\end{aligned} \tag{2.4}$$

The partial order constraints include

- $\forall 1 \le i < k, \forall r \in S_i, \gamma(r) = \Gamma_i$
- $\forall a, b \in (S, \le) : a \le_S b \Rightarrow \gamma(a) \le \gamma(b)$
- $\forall d \in D_{k-1}, \gamma(d) \le 0$; $\forall u \in U_{k-1}, \gamma(u) \ge 0$

After removing S_1 and optimizing (2.4) for binary outcomes, the set $u \in U_1 : \gamma(u) = 0$ should be included in the stratum S_2. By recursively applying a similar method to select a down-set as a new cancer stage from the residual poset, the entire risk factor set $(S, \le)$ can be stratified step by step. At the kth iteration, $(k + |U_{k-1}| + p)$ parameters need to be estimated.

2.2.2 Optimization

To solve the $L1$ penalized logistic regression [Lin+16], an iterative procedure can be used. This involves expressing the standard Newton-Raphson update as the method of iteratively reweighted least squares (IRLS), followed by replacing the weighted least squares step with a constrained weighted least squares procedure. Because our problem does not involve high-dimensional data, this procedure is suitable for accurately computing its estimates.

Define

$$\begin{aligned}
\pi_i &= \frac{1}{1+\exp(-\eta_i)} \\
y_i &\sim BIN(1,\pi_i) \\
\eta = Z^T\alpha + R^T\gamma + \mathbf{1}\mu &= X^T\beta \\
Z &= [Z_1, Z_2, \ldots, Z_n] \\
R &= [r_1, r_2, \ldots, r_n] \\
X^T &= [\mathbf{1}|R^T|Z^T] \\
\beta^T &= [\mu|\gamma^T|\alpha^T] \\
\pi &= [\pi_1, \pi_2, \ldots, \pi_n]^T \\
y &= [y_1, y_2, \ldots, y_n]^T \\
\eta &= [\eta_1, \eta_2, \ldots, \eta_n]^T
\end{aligned} \tag{2.5}$$

In equation (2.5), the vector r_i denotes the category of risk factors or the stage that the subject i belongs to, at the kth iteration of OPERA. This vector has a dimension of $(|U_{k-1}| + k - 1)$.

Define

$$\begin{aligned}
u &= \frac{\partial l}{\partial \eta} \\
u_i &= \frac{\partial l}{\partial \eta_i} = y_i - \pi_i \\
A = \frac{-\partial l^2}{\partial \eta \eta^T} &= \mathrm{diag}(\pi_i(1-\pi_i)) \\
z &= \eta + A^{-1}u
\end{aligned} \tag{2.6}$$

To obtain a down-set S_k from a residual poset U_{k-1}, using a binary outcome, OPERA employs an iterative procedure at the kth iteration, which can be outlined as follows:

1. Fix a value of λ and initialize $\hat{\beta}$
2. Compute η, u, A and z based on the current value of $\hat{\beta}$
3. Minimize $(z-\eta)^T A(z-\eta) + \lambda \mathbf{1}^T \gamma$ subject to the constraints including
   - $\forall 1 \leq i < k, \forall s \in S_i, \gamma(s) = \Gamma_i$ and $\Gamma_1 \leq \Gamma_2 \leq \cdots \leq \Gamma_{k-1} \leq 0$
   - $\forall a, b \in (U_{k-1}, \leq)$, if $a \leq_{U_{k-1}} b$ then $\gamma(a) \leq \gamma(b)$.
   - For all $d \in D_{k-1} = \cup_{i=1}^{k-1} S_i$, we have $\gamma(d) \leq 0$ and for all $u \in U_{k-1}$, we have $\gamma(u) \geq 0$.
4. Repeat Steps 2 and 3 until $\hat{\beta}$ converges.

Algorithm 2: IRLS for Binary Outcome

Data: $(y_i, r_i, Z_i)_{i=1}^n, \lambda, k, \beta_0$; /* $\beta_0 \in \mathbb{R}^d$ */
Result: $\beta_\lambda^T = [\mu_\lambda, \gamma_\lambda^T, \alpha_\lambda^T]$

$Z \leftarrow [Z_1, Z_2, \ldots, Z_n]$, $R \leftarrow [r_1, r_2, \ldots, r_n]$, $X^T \leftarrow [\mathbf{1}|R^T|Z^T]$, $d \leftarrow |U_{k-1}| + k + p$, $q \leftarrow 0$; /* $\mathbf{1} \in \mathbb{R}^n$ */

while $\frac{1}{d}|\beta_q - \beta_{q-1}|_1 > \epsilon$ *and* $q \leq 10$; /* $\epsilon = 10^{-4}, \beta_q \in \mathbb{R}^d$ */

do

  $\eta_q \leftarrow X^T \beta_q$; /* $\eta = Z^T\alpha + R^T\gamma + \mathbf{1}\mu = X^T\beta$ */

  $\pi_q \leftarrow \frac{1}{1+\exp(-\eta_q)}$, $u_q \leftarrow y_q - \pi_q$, $A_q \leftarrow \mathrm{diag}(\pi_q \odot (1 - \pi_q))$

  $z_q \leftarrow \eta_q + A_q^{-1} u_q$; /* $z_i = 0$ `if` $\frac{\partial l^2}{\partial \eta_i^2} = 0$ */

  $\beta_q \leftarrow \arg\min_{\beta,\gamma}(z_q - X^T\beta)^T A_q (z_q - X^T\beta) + 2\lambda \mathbf{1}^T\gamma =$
  $\arg\min_{\beta,\gamma} \beta^T X A_q X^T \beta - 2 z_q^T A_q X^T \beta + 2\lambda \mathbf{1}^T \gamma$ s.t.
   - $\forall 1 \leq i < k, \forall s \in S_i, \gamma(s) = \Gamma_i$, $\Gamma_1 \leq \Gamma_2 \leq \cdots \leq \Gamma_{k-1} \leq 0$
   - $\forall a, b \in (U_{k-1}, \leq) : a \leq_{U_{k-1}} b \Rightarrow \gamma(a) \leq \gamma(b)$
   - $\forall d \in D_{k-1} = \cup_{i=1}^{k-1} S_i, \gamma(d) \leq 0$; $\forall u \in U_{k-1}, \gamma(u) \geq 0$

  $q \leftarrow q + 1$

end

By dropping the absolute values, the minimization problem in Step 3 reduces to a simple quadratic program with linear inequality constraints. This can be efficiently solved using the R Package *quadprog* [Tur19][GI82][GI83]. Based on empirical experience, the

iterative procedure typically converges rapidly. All computational tasks are performed using R version 4.2.0.

The IRLS method is described in detail in Algorithm 2. To obtain the estimates denoted as $(\mu_\lambda, \gamma_\lambda, \alpha_\lambda)$, the tuning parameter λ is provided, and the method is executed for find the $(k+1)$th stage when k stages have been found. The initial value of $\hat{\beta}$ can be set as β_0 by utilizing the logistic regression coefficients with no imposed constraints and setting λ to zero. The number of iterations is kept track of by the variable q. Line 7 employs 2λ to simplify computation and implementation. Moreover, XA_qX^T is substituted with $XA_qX^T + \epsilon I$ in consideration of the possibility that X may be singular. Here, ϵ is a small positive value that helps prevent numerical instability.

2.2.3 Tuning Parameter Selection

Algorithm 3 highlights the parameter tuning step in OPERA, which is a necessary step before identifying a down-set. Assuming that the total number of stages found at the end is m, we can define the poset $S = \cup_{j=1}^{m} S_j = \cup_{h=1}^{|S|} s_h$ as formed by the union of m stages or $|S|$ categories defined by risk factors, and the variable k keeps track of the current number of discovered stages. Before finding a down-set, a grid of λ's is uniformly selected on the logarithmic scale between λ_{min} and λ_{max}, which can be determined through binary search. The number of zero r_i's is counted before finding λ_{min} and λ_{max} by setting λ to zero. In the search for λ_{max}, λ is initially set to 1. If any r_i's are non-zero, λ is doubled until all r_i's are zero. If all r_i's are zero, λ is halved until some r_i is non-zero. Similarly, in the search for λ_{min}, λ is initially set to 1. If any r_i's are zero, λ is halved until the number of zero r_i's matches the count when λ equals 0. If all r_i's are non-zero, λ is doubled until some r_i is zero. N denotes the total number of λ's to be searched, and all λ's are evenly spaced on the logarithmic scale.

The Akaike information criterion (AIC) [Aka73] is a method proposed to select the optimal tuning parameter λ, defined as:

Algorithm 3: OPERA with Parameter Tuning

Data: $(y_i, r_i, Z_i)_{i=1}^n$

Result: $s_h \in S_j$, where $j = 1, 2, \ldots, m$, $h = 1, 2, 3, \ldots, |S|$ and $s_h \in (S, \leq)$

$D_0 = \emptyset, U_0 = S, k = 0$; /* Initialization */

while $U_k! = \emptyset$ **do**

  $k_1 \leftarrow 0, \lambda \leftarrow 1$

  while $\sum 1[\gamma_\lambda! = 0] > 0$ *and* $k_1 <= 30$ **do**

    $(\mu_\lambda, \gamma_\lambda, \alpha_\lambda) \leftarrow IRLS((y_i, r_i, Z_i)_{i=1}^n, \lambda)$

    $\lambda \leftarrow \lambda \times 2, k_1 \leftarrow k_1 + 1$; /* while any r_i's are non-zero */

  end

  if $k_1 > 1$ **then**

    $\lambda_{max} \leftarrow \lambda/2$

  else

    $k_2 = 0, \lambda \leftarrow 1$

    while $\sum 1[\gamma_\lambda! = 0] = 0$ *and* $k_2 <= 30$ **do**

      $(\mu_\lambda, \gamma_\lambda, \alpha_\lambda) \leftarrow IRLS((y_i, r_i, Z_i)_{i=1}^n, \lambda)$

      $\lambda \leftarrow \lambda/2, k_2 \leftarrow k_2 + 1$; /* while all r_i's are zero */

    end

    $\lambda_{max} \leftarrow \lambda \times 4$

  end

  $k_3 = 0, \lambda \leftarrow 1$

  while $\sum 1[\gamma_\lambda = 0] = 0$ *and* $k_3 <= 30$ **do**

    $(\mu_\lambda, \gamma_\lambda, \alpha_\lambda) \leftarrow IRLS((y_i, r_i, Z_i)_{i=1}^n, \lambda)$

    $\lambda \leftarrow \lambda \times 2, k_3 \leftarrow k_3 + 1$; /* while all r_i's are non-zero */

  end

  if $k_3 > 1$ **then**

    $\lambda_{min} \leftarrow \lambda/4$

  else

    $k_4 = 0, \lambda \leftarrow 1$

    while $\sum 1[\gamma_\lambda = 0] > \sum 1[\gamma_{\lambda=0}]$ *and* $k_4 <= 30$ **do**

      $(\mu_\lambda, \gamma_\lambda, \alpha_\lambda) \leftarrow IRLS((y_i, r_i, Z_i)_{i=1}^n, \lambda)$

      $\lambda \leftarrow \lambda/2, k_4 \leftarrow k_4 + 1$; /* until the number of zero r_i's matches the count when λ equals 0 */

    end

    $\lambda_{min} \leftarrow \lambda \times 2$

  end

  for λ *in* $\{\exp(\log \lambda_{min} + \frac{\log \lambda_{max} - \log \lambda_{min}}{N-1} i)\}_{i=0}^{N-1}$, $N \leftarrow 30$ **do**

    $(\mu_\lambda, \gamma_\lambda, \alpha_\lambda) \leftarrow IRLS((y_i, r_i, Z_i)_{i=1}^n, \lambda)$, $AIC_\lambda \leftarrow AIC(\mu_\lambda, \gamma_\lambda, \alpha_\lambda)$

    if $AIC_\lambda < AIC_{best}$ **then**

      $AIC_{best} \leftarrow AIC_\lambda$, $(\mu_{best}, \gamma_{best}, \alpha_{best}) \leftarrow (\mu_\lambda, \gamma_\lambda, \alpha_\lambda)$

    end

  end

  $(\hat{\mu}, \hat{\gamma}, \hat{\alpha}) \leftarrow (\mu_{best}, \gamma_{best}, \alpha_{best})$

  $k \leftarrow k + 1$

  $S_k \leftarrow \{u \in U_{k-1} : \hat{\gamma}(u) = 0\}$, $D_k \leftarrow D_{k-1} \cup S_k$, $U_k \leftarrow S - D_k$

end

$$AIC(\lambda) = -2l(\mu_\lambda, \gamma_\lambda, \alpha_\lambda) + 2df_\lambda \tag{2.7}$$

where $l(\mu_\lambda, \gamma_\lambda, \alpha_\lambda)$ is the log-partial likelihood for the constrained fit with λ, and df_λ is the degree of freedom in the model, estimated by the number of non-zero γ's. The AIC enhances the negative log-partial likelihood by a penalty term that is proportional to the effective number of parameters. Each time OPERA attempts to find a down-set, the AIC is computed over a range of values of λ, uniformly distributed on the log scale from λ_{min} to λ_{max}. The best λ that minimizes the estimated AIC is chosen.

2.3 Coarse Pruning

Algorithm 1 generates outputs $s_h \in S_j$, where $j = 1, 2, \ldots, m$, $h = 1, 2, 3, \ldots, |S|$, and $s_h \in (S, \leq)$, assigning each category defined by risk factors with a cancer stage. The assigned stages follow an ordering from the least advanced to the most advanced cancer stage. However, as Algorithm 1 does not have prior knowledge about the number of stages, it may over-partition the tree-like structure, as illustrated in Figure 2.1. To overcome this potential issue, a pruning procedure is introduced.

An approach to refining the results obtained from Algorithm 1 involves iteratively reducing one stage. In each iteration, we also iteratively merge each adjacent pair of stages and use the likelihood to determine which stage to prune. The stopping rules for pruning can be either the AIC, the Brier Score (BS)[Bri+50], or the likelihood ratio test (LRT) with a p-value smaller than α. This helps us choose the best combination that fits the data.

The process, as outlined in Algorithm 4, includes an outer iteration and an inner iteration. In the outer loop, we iteratively reduce the number of stages by 1, calculate the BS or AIC, and check if the stopping criterion is met. When the BS or AIC is used as the stopping rule, we need to iteratively reduce the number of stages to 2. However, when

the LRT is used, we can break the loop anytime the p-value is no more than α. In each inner loop, we iteratively choose the best set of coefficients with the maximum likelihood. The new coefficients are estimated using the IRLS algorithm, subject to constraints that ensure all coefficients comply with the ordering embedded in adjacent cancer stages.

Algorithm 4: Coarse Pruning

1 From Algorithm 1, m stages are found.
2 **while** $m - 2 > 0$ **do**
3 Estimate the coefficients for cancer stages given the total ordering among different stages (Algorithm 5)
4 Calculate the corresponding likelihood l_m, the BS BS_m and the AIC AIC_m
5 $m \leftarrow m - 1$
6 **for** *j in 2:(m+1)* **do**
7 Merge S_j, S_{j-1} into one stage S_j
8 Estimate coefficients using IRLS under the following constraint: $\zeta(S_1) \leq \dots \zeta(S_{j-2}) \leq \zeta(S_j) \leq \zeta(S_{j+1}) \leq \cdots \leq \zeta(S_{m+1})$
9 Calculate the corresponding likelihood $l_{m,j}$
10 **end**
11 Select S_j for pruning if it yields the maximum likelihood $l_m = \max_j l_{m,j}$
12 Calculate the BS BS_m and the AIC AIC_m
13 **if** *the LRT is used* **then**
14 $-2(l_m - l_{m+1}) \sim \chi^2(1)$
15 **if** *p-value* $\leq \alpha$ **then**
16 **break**
17 **end**
18 **end**
19 **end**
20 **if** *the BS or AIC is used* **then**
21 Select the final result with the smallest BS or AIC
22 **end**

It is essential to note that in line 8 of Algorithm 4, the process of estimating coefficients for each cancer stage involves IRLS under linear constraints. The coefficients can be initialized as β_0 using the logistic regression model. To estimate the coefficients, the total ordering is then incorporated into IRLS as linear constraints, as shown in Algorithm 5 line 7. This ensures that the coefficients satisfy the ordering among different stages. Here, r_i represents an m-dimensional one-hot vector indicating the original stage obtained from OPERA or the new combined stage to which subject i belongs.

Algorithm 5: Iteratively Reweighted Least Squares to Estimate the Coefficients for Stages (IRLSG)

Data: $(y_i, r_i, Z_i)_{i=1}^n, S = \tilde{S}_1 \cup \tilde{S}_2 \cup \cdots \cup \tilde{S}_{\tilde{m}}, \beta_0$
Result: $\beta^T = [\zeta^T, \alpha^T], \zeta \in \mathbb{R}^{\tilde{m}}, \alpha \in \mathbb{R}^p$
$Z \leftarrow [Z_1, Z_2, \ldots, Z_n]$, $R \leftarrow [r_1, r_2, \ldots, r_n]$, $X^T \leftarrow [R^T | Z^T], q \leftarrow 0$
while $\frac{1}{\tilde{m}+p}|\beta_q - \beta_{q-1}|_1 > \epsilon$ *and* $q \leq 20$; /* $\epsilon = 10^{-4}, \beta_q \in \mathbb{R}^{\tilde{m}+p}$ */
do
  $\eta_q \leftarrow X^T \beta_q$; /* $\eta = Z^T\alpha + R^T\zeta = X^T\beta$ */
  $\pi_q \leftarrow \frac{1}{1+\exp(-\eta_q)}$, $u_q \leftarrow y_q - \pi_q$, $A_q \leftarrow \mathrm{diag}(\pi_q \odot (1 - \pi_q))$
  $z_q \leftarrow \eta_q + A_q^{-1} u_q$; /* $z_i = 0$ `if` $\frac{\partial l^2}{\partial \eta_i^2} = 0$ */
  $\beta_q \leftarrow \arg\min_{\zeta,\alpha}(z_q - X^T\beta)^T A_q (z_q - X^T\beta)$ s.t.
    - $\zeta(\tilde{S}_1) \leq \zeta(\tilde{S}_2) \cdots \leq \zeta(\tilde{S}_{\tilde{m}})$
  $q \leftarrow q + 1$
end

Algorithm 6: Coarse Pruning with IRLSG

Data: $(y_i, r_i, Z_i)_{i=1}^n, S = \cup_{j=1}^m S_j = \cup_{h=1}^{|S|} s_h$
Result: $\dot{S}$
while $m - 2 > 0$ **do**
  $[\zeta^T, \alpha^T] \leftarrow IRLSG((y_i, r_i, Z_i)_{i=1}^n, S, \beta_0), AIC_m \leftarrow AIC(\zeta, \alpha), l_m \leftarrow l(\zeta, \alpha), BS_m \leftarrow BS(\zeta, \alpha), \dot{S}^m = S$
  $m \leftarrow m - 1$
  for *j in 2:(m+1)* **do**
    $[\zeta^T, \alpha^T] \leftarrow IRLSG((y_i, r_i, Z_i)_{i=1}^n, S = S_1 \cup S_2 \cdots \cup (S_{j-1} \cup S_j) \cup S_{j+1} \cdots \cup S_{m+1}, \beta_0), AIC_{m,j} \leftarrow AIC(\zeta, \alpha), l_{m,j} \leftarrow l(\zeta, \alpha), BS_{m,j} \leftarrow BS(\zeta, \alpha)$
  end
  $J = \arg\max_j l_{m,j}, l_m = l_{m,J}, BS_m = BS_{m,J}, AIC_m = AIC_{m,J}$
  $\dot{S}_1 \leftarrow S_1, \ldots, \dot{S}_{J-1} \leftarrow S_{J-1} \cup S_J, \dot{S}_J \leftarrow S_{J+1}, \ldots, \dot{S}_m \leftarrow S_{m+1}, \dot{S}^m = \cup_{k=1}^m \dot{S}_k$
  if *the LRT is used* **then**
    $-2(l_m - l_{m+1}) \sim \chi^2(1)$
    if *p-value* $\leq \alpha$ **then**
      $\dot{S} = \dot{S}^{m+1}$
      break
    end
  end
end
if *the BS or AIC is used* **then**
  $\dot{S} = \dot{S}^K, K = \arg\min_m BS_m$ or $K = \arg\min_m AIC_m$
end

Algorithm 6 provides further technical details on the coarse pruning algorithm. The coarse pruning process is carried out only if the OPERA algorithm produces more than two stages. The poset $S = \cup_{j=1}^{m} S_j = \cup_{h=1}^{|S|} s_h$ is formed by the union of all identified stages or groups of risk factors. In this representation, higher stages indicate higher risk, while lower stages indicate lower risk.

When the AIC or BS is used as the stopping rule, the combination with the lowest AIC or BS is selected as the final coarse pruning result. To calculate the AIC, we use the equation:

$$AIC(\zeta, \alpha) = -2l(\zeta, \alpha) + 2df \tag{2.8}$$

Here, $l(\zeta, \alpha)$ represents the log-partial likelihood with the estimated coefficients under the total ordering, and df is the degree of freedom in the model, estimated by the current number of stages. The AIC penalizes the negative log-partial likelihood by a term proportional to the total number of stages.

The BS is essentially the mean squared error (MSE) for binary outcomes. To calculate the BS, we use the equation:

$$BS(\zeta, \alpha) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{p}_i)^2 \tag{2.9}$$

Here, $\hat{p}_i$ is the estimated probability of experiencing the event (i.e., $P(Y_i = 1)$).

2.4 Fine Pruning

Algorithm 4 demonstrates that the coarse pruning procedure takes $\mathcal{O}(m^2)$ operations to obtain the pruned result, where m represents the number of stages identified by Algorithm 1. However, coarse pruning has a limitation: it does not allow the merging of certain groups of risk factors Q within a stage S_j into a neighboring stage S_{j-1} while combining

the remainder into another stage S_{j+1}. To address this limitation, we introduce a fine pruning procedure.

The key to fine pruning is to enforce the merging of $Q \subseteq S_j$ into S_{j-1} and the merging of $S_j \backslash Q$ into S_{j+1}. A brute-force approach to achieving this is to attempt merging each element in S_j into either S_{j-1} or S_{j+1}, which takes $\mathcal{O}(2^{|S_j|})$ operations. This approach is known as an **exhaustive search**. To mitigate the complexity associated with the brute-force approach, we introduce a **quadratic constraint** in the IRLS procedure as an alternative approach.

2.4.1 Exhaustive Search

The OPERA algorithm classifies each group of risk factors into stages according to Algorithm 1. To recombine stages that may have been over-partitioned, coarse pruning is used as outlined in Algorithm 4. Alternatively, fine pruning involves an exhaustive search that considers all possible scenarios satisfying the partial ordering constraints when a stage is removed and split into neighboring stages. The optimal scenario for splitting and merging is determined using the maximum likelihood when each stage is removed, and the candidate stage to be pruned is also determined using the maximum likelihood afterwards. Finally, the LRT or BS or AIC is employed to determine whether a stage should be pruned, and the pruning process is stopped accordingly.

Similar to coarse pruning, the fine pruning process is executed only if the OPERA algorithm yields more than two stages, as demonstrated in Algorithm 7. Line 6 and 7 depict the main distinction between coarse pruning and fine pruning. Coarse pruning focuses on pruning the results at the stage level, whereas fine pruning delves into pruning at the node level. When the LRT is performed between m and $m-1$ stages, if the p-value is no more than α, the pruning step stops. Otherwise, the pruning continues and the total number of stages decreases by 1. The pruning will not cease until the p-value is larger than α. When the BS or AIC is used, the pruning continues until only two stages

remain. The final result is based on the smallest BS or AIC, as indicated in Algorithm 7.

Algorithm 7: Fine Pruning with an Exhaustive Search

From Algorithm 1, m stages are found.
while $m - 2 > 0$ **do**
  Estimate the coefficients for cancer stages given the total ordering among different stages (Algorithm 5), and calculate the corresponding likelihood l_m, the BS BS_m and the AIC AIC_m
  $m \leftarrow m - 1$
  for *j in 2:m* **do**
    Generate a matrix with dimensions of $2^{|S_j|} \times |S_j|$. Each row represents a vector that signifies a potential staging scenario, indicating whether each node in S_j merges into S_{j-1} or S_{j+1}. Remove any rows from the matrix where the partial ordering constraints are violated.
    Iterate through all possible rows in the matrix, estimating the corresponding coefficients that adhere to the total ordering constraint among various stages. Select the row with the highest likelihood $l_{m,j}$.
  end
  Select S_j for pruning if it yields the maximum likelihood $l_m = \max_j l_{m,j}$, and calculate the BS BS_m and the AIC AIC_m
  if *the LRT is used* **then**
    $-2(l_m - l_{m+1}) \sim \chi^2(1)$
    if *p-value* $\leq \alpha$ **then**
      break
    end
  end
end
if *the BS or AIC is used* **then**
  Select the final result with the smallest BS or AIC
end

Algorithm 8 provides further technical insights into our methodology. The log-likelihood, the BS, and the AIC are computed before each iteration of stage pruning. When selecting the optimal scenario for stage pruning, we iterate through all current stages as the outer loop. Within each stage, we explore all possible scenarios of splitting into neighboring stages as the inner loop. To represent these scenarios, we use the matrix $A_{2^{|S_j|} \times |S_j|}$. Additionally, we use $P_{\cdot \times |S_j|}$ to detail the relevant partial ordering constraints, filtering out scenarios that violate the ordering constraint. By multiplying $A_{2^{|S_j|} \times |S_j|}$ by $P^T_{|S_j| \times \cdot}$, we obtain a new matrix W with $2^{|S_j|}$ rows. Rows with negative values in W

indicate that the corresponding rows in A should be removed, before we are left with c scenarios that are then selected based on the log-likelihood for further consideration.

Algorithm 8: Fine Pruning with an Exhaustive Search with IRLSG

Data: $(y_i, r_i, Z_i)_{i=1}^n, S = \cup_{j=1}^m S_j = \cup_{h=1}^{|S|} s_h$
Result: $\dot{S}$
while $m - 2 > 0$ **do**
  $[\zeta^T, \alpha^T] \leftarrow IRLSG((y_i, r_i, Z_i)_{i=1}^n, S, \beta_0), AIC_m \leftarrow AIC(\zeta, \alpha), l_m \leftarrow l(\zeta, \alpha), BS_m \leftarrow BS(\zeta, \alpha), \dot{S}^m = S$
  $m \leftarrow m - 1$
  for *j in 2:m* **do**
    Find the rows of $A_{2^{|S_j|} \times |S_j|} P^T_{|S_j| \times .}$ with negative values
    Remove the corresponding rows to get $A'_{c \times |S_j|}$
    for *k in 1:c* **do**
      $[\zeta^T, \alpha^T] \leftarrow IRLSG((y_i, r_i, Z_i)_{i=1}^n, S, \beta_0), AIC_{m,j,c} \leftarrow AIC(\zeta, \alpha), l_{m,j,c} \leftarrow l(\zeta, \alpha), BS_{m,j,c} \leftarrow BS(\zeta, \alpha)$
    end
  end
  $J, C = \arg\max_{j,c} l_{m,j,c}, l_m = l_{m,J,C}, BS_m = BS_{m,J,C}, AIC_m = AIC_{m,J,C}$
  $\dot{S}_1 \leftarrow S_1, \ldots, \dot{S}_{J-1} \leftarrow S_{J-1} \cup S_{J,C}, \dot{S}_J \leftarrow S_{J+1} \cup (S_J \backslash S_{J,C}), \ldots, \dot{S}_m \leftarrow S_{m+1}, \dot{S}^m = \cup_{p=1}^m \dot{S}_p$
  if *the LRT is used* **then**
    $-2(l_m - l_{m+1}) \sim \chi^2(1)$
    if *p-value* $\leq \alpha$ **then**
      $\dot{S} = \dot{S}^{m+1}$
      break
    end
  end
end
if *the BS or AIC is used* **then**
  $\dot{S} = \dot{S}^K, K = \arg\min_m BS_m$ or $K = \arg\min_m AIC_m$
end

2.4.2 Quadratic Constraint

Definition 2.4.1. Suppose that Algorithm 1 yields m stages, such that $S = \cup_{j=1}^m S_j$. If it is necessary to prune S_j, then we must merge $\tilde{S} \subseteq S_j$ into S_{j-1}, and $S_j \backslash \tilde{S}$ into S_{j+1}. The coefficients associated with the stages obey a total ordering, expressed as $\zeta(S_1) \leq$

$\zeta(S_2) \leq \cdots \leq \zeta(S_m)$. This merging problem can be cast as a quadratic programming constraint, as given in Equation 2.10:

$$\sum_s [(\zeta(s) - \zeta(S_{j-1}))^2 + (\zeta(S_{j+1}) - \zeta(s))^2] \geq \sum_s (\zeta(S_{j+1}) - \zeta(S_{j-1}))^2 \quad \forall s \in S_j, \zeta(S_{j-1}) \leq \zeta(s) \leq \zeta(S_{j+1}) \tag{2.10}$$

Proof. If $a = \zeta(s) - \zeta(S_{j-1}) \geq 0$, $b = \zeta(S_{j+1}) - \zeta(s) \geq 0$, then $g = (a+b)^2 - a^2 - b^2 \geq 0$. Only if $a = 0$ or $b = 0$, then $g = 0$. Thus, if $(\zeta(s) - \zeta(S_{j-1}))^2 + (\zeta(S_{j+1}) - \zeta(s))^2 - (\zeta(S_{j+1}) - \zeta(S_{j-1}))^2 = a^2 + b^2 - (a+b)^2 = -g \geq 0$, then $a = 0$ or $b = 0$ can be concluded. This implies that either $\zeta(s) = \zeta(S_{j-1})$ or $\zeta(s) = \zeta(S_{j+1})$. □

Definition 2.4.2. If S_j needs to be pruned and β^T can be partitioned as $\beta^T = [\zeta^T, \alpha^T]$, then $\zeta^T = [\zeta_1, \zeta_2, \ldots, \zeta_{j-1}, \zeta(s)^T, \zeta_{j+1}, \ldots, \zeta_m]$, where $s \in S_j$, $\zeta_i = \zeta(S_i)$, and $\zeta \in \mathbb{R}^{m-1+|S_j|}$. The quadratic programming constraint in Equation 2.10 can be rewritten as:

$$V_{2|S_j| \times (m-1+|S_j|)} = \begin{bmatrix} 0 & \ldots & -1 & 1 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ . & \ldots & . & . & . & \ldots & . & . & \ldots & . \\ 0 & \ldots & -1 & 0 & 0 & \ldots & 1 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & -1 & 0 & \ldots & 0 & 1 & \ldots & 0 \\ . & \ldots & . & . & . & \ldots & . & . & \ldots & . \\ 0 & \ldots & 0 & 0 & 0 & \ldots & -1 & 1 & \ldots & 0 \end{bmatrix}, \tag{2.11}$$

$$\zeta^T M \zeta = \zeta^T (V^T V - |S_j| v v^T) \zeta = \zeta^T V^T V \zeta - |S_j| \zeta^T v v^T \zeta \geq 0,$$

$$v_{(m-1+|S_j|) \times 1} = [0, 0, \ldots, -1, 0, 0, \ldots, 0, 1, 0, 0, \ldots, 0]^T,$$

$$M = V^T V - |S_j| v v^T.$$

Algorithm 9 presents a methodical approach for fine pruning utilizing a quadratic programming constraint. Prior to the stage pruning, coefficients for the current stages and the corresponding likelihood are estimated within the context of the total ordering among the stages, as outlined in Algorithm 5. The BS, AIC, and log-likelihood are also

computed as part of the pruning steps. The primary distinction between the exhaustive search and the quadratic programming constraint lies in the approach. The latter involves introducing a quadratic constraint to ensure the splitting and merging of all nodes from a stage. In contrast, the former enumerates all possible scenarios. Nevertheless, the scenario with the maximum log-likelihood remains the optimal choice for stage pruning.

Algorithm 9: Fine Pruning with a Quadratic Programming Constraint

From Algorithm 1, m stages are found.
while $m - 2 > 0$ **do**
    Estimate the coefficients for cancer stages given the total ordering among different stages (Algorithm 5), and calculate the corresponding likelihood l_m, the BS BS_m and the AIC AIC_m
    $m \leftarrow m - 1$
    for *j in 2:m* **do**
        Perform IRLS under three different scenarios, including no linear constraint on coefficients, $\zeta_m \leq \max(\zeta_0)$, or $\zeta_m \leq \max(\zeta_0)$ and $\zeta_1 \geq \min(\zeta_0)$
        Under each scenario,
            Introduce a Lagrangian parameter λ in each iteration of IRLS to ensure that the quadratic constraint is satisfied (Algorithm 10)
            Tune the value of λ and select the set of coefficients that maximizes the likelihood (Algorithm 11)
        Choose the coefficients with the highest likelihood during IRLS, regardless of the number of linear constraints
    end
    Select S_j for pruning if it yields the maximum likelihood $l_m = \max_j l_{m,j}$, and calculate the BS BS_m and the AIC AIC_m
    if *the LRT is used* **then**
        $-2(l_m - l_{m+1}) \sim \chi^2(1)$
        if *p-value* $\leq \alpha$ **then**
            break
        end
    end
end
if *the BS or AIC is used* **then**
    Select the final result with the smallest BS or AIC
end

The quadratic programming constraint offers an advantageous and unique approach by avoiding the need to enumerate all possible combinations, which can be exponen-

tially large ($\mathcal{O}(2^{|S_j|})$), when attempting to prune S_j. However, the drawback lies in the non-convex nature of this constraint. In other words, unlike the exhaustive search, the quadratic constraint cannot guarantee to produce the optimal solution with the maximum log-likelihood. When the linear constraints $\zeta_{j-1} \leq \zeta(s) \leq \zeta_{j+1}, s \in S_j$ are satisfied, the condition $\zeta^T M \zeta \leq 0$ holds. Checking whether the difference between $-\zeta^T M \zeta$ and zero is smaller than a tolerance number ϵ ensures the fulfillment of the quadratic constraint and enables successful pruning.

Algorithm 10: One Iteration of IRLS with Linear and Quadratic Constraints during Pruning without Parameter Tuning (OIRLS)

Data: $(y_i, r_i, Z_i)_{i=1}^n, \lambda, S = \tilde{S}_1 \cup \tilde{S}_2 \cup \cdots \cup \tilde{S}_{\tilde{m}}, \beta_{q-1}, \beta_0, c$
Result: $\beta_q = (\zeta_q, \alpha_q), l(\beta_q)$

$Z \leftarrow [Z_1, Z_2, \ldots, Z_n]$, $R \leftarrow [r_1, r_2, \ldots, r_n]$, $X^T \leftarrow [R^T | Z^T]$, $\eta_{q-1} \leftarrow X^T \beta_{q-1}$

$\pi_{q-1} \leftarrow \frac{1}{1+\exp(-\eta_{q-1})}$, $u_{q-1} \leftarrow y_{q-1} - \pi_{q-1}$, $A_{q-1} \leftarrow \text{diag}(\pi_{q-1} \odot (1 - \pi_{q-1}))$

$z_{q-1} \leftarrow \eta_{q-1} + A_{q-1}^{-1} u_{q-1}$; /* $z_i = 0$ if $\frac{\partial l^2}{\partial \eta_i^2} = 0$ */

$\beta_q \leftarrow \arg\min_{\zeta,\alpha} (z_{q-1} - X^T \beta_{q-1})^T A_{q-1} (z_{q-1} - X^T \beta_{q-1}) - \lambda \zeta_{q-1}^T M \zeta_{q-1}$

- $\zeta(\tilde{S}_1) \leq \zeta(\tilde{S}_2) \leq \cdots \leq \zeta(\tilde{S}_{j-1}) \leq \zeta(s) \leq \zeta(\tilde{S}_{j+1}) \leq \cdots \leq \zeta(\tilde{S}_{\tilde{m}}), \forall s \in S_j$
- $a, b \in S_j$, if $a \leq_{S_j} b$, then $\zeta(a) \leq \zeta(b)$
- More linear constraints
  - If $c = 0$, then no additional linear constraints are involved;
  - If $c = 1$, then $\zeta(\tilde{S}_{\tilde{m}}) \leq \max(\zeta_0)$;
  - If $c = 2$, then $\zeta(\tilde{S}_{\tilde{m}}) \leq \max(\zeta_0)$ and $\zeta(S_1) \geq \min(\zeta_0)$

The non-positiveness of M and its associated non-convex constraint introduce complexity to the estimation process during pruning, specifically with the use of IRLS for coefficient estimation. The non-convexity of the optimization problem implies that convergence of IRLS cannot be guaranteed. To enhance estimation accuracy, we initialize ζ_0 with coefficients estimated from a logistic regression model. Additionally, as shown in Algorithm 10, we employ three different scenarios to select the optimal set of coefficients based on the likelihood: (1) no additional linear constraint on coefficients, (2) $\zeta_m \leq \max(\zeta_0)$, or (3) $\zeta_m \leq \max(\zeta_0)$ and $\zeta_1 \geq \min(\zeta_0)$. These strategies help facilitate

estimation and maintain control over the overall estimated coefficients within a certain range.

During each iteration of IRLS, a Lagrangian parameter λ is introduced to ensure the fulfillment of the quadratic constraint. This is accomplished by minimizing $-\zeta^T M\zeta$ until it reaches zero, as illustrated in line 4 of Algorithm 10. The appropriate tuning of the parameter λ depends on the feasibility of satisfying the quadratic constraint $\zeta^T M\zeta = 0$. Initially, λ is assigned a value of 1, as depicted in Algorithm 11. Subsequently, a binary search is performed to determine whether the quadratic constraint can be satisfied. Due to the non-convexity of M, we use the nearest positive definite[Hig88] of the quadratic matrix in the minimization problem in line 4 of Algorithm 10.

Irrespective of whether IRLS converges during the pruning procedure, the incorporation of additional linear constraints, or the number of iterations required to satisfy the quadratic constraint, the set of coefficients with the maximum likelihood is consistently selected. This selection process is demonstrated in line 24-35 of Algorithm 11 and line 4-12 of Algorithm 12. As long as the quadratic constraint is met, the coefficients yielding the highest likelihood are chosen, regardless of the specific IRLS convergence behavior or the inclusion of linear constraints.

Each stage S_i is evaluated as a potential candidate for pruning, and the stage that corresponds to the maximum likelihood is ultimately chosen. Once S_j is selected, either the LRT or the BS or the AIC is employed. In the case of the LRT, a test is conducted between m and $m-1$ stages, and if the resulting p-value is equal to or lower than the predefined Type I error rate α, the pruning procedure is halted. Conversely, if the p-value exceeds α, the pruning process continues, and the total number of stages decreases by 1. Alternatively, when using the BS or AIC, the total number of stages is determined based on the smallest value, as indicated in lines 23-25 of Algorithm 12. In this approach, the stage configuration with the most favorable BS or AIC is selected, and the pruning procedure continues until only two stages are left.

Algorithm 11: One Iteration of IRLS with Linear and Quadratic Constraints during Pruning with Parameter Tuning (OIRLST)

Data: $(y_i, r_i, Z_i)_{i=1}^n, S = \tilde{S}_1 \cup \tilde{S}_2 \cup \cdots \cup \tilde{S}_{\tilde{m}}, \beta_{q-1}, \beta_0, c$
Result: $\beta_q = [\zeta_q, \alpha_q]$

$Z \leftarrow [Z_1, Z_2, \ldots, Z_n]$, $R \leftarrow [r_1, r_2, \ldots, r_n]$, $X^T \leftarrow [R^T | Z^T]$, $\lambda \leftarrow 1$, $t_1 \leftarrow 0$
$\beta_q \leftarrow OIRLS((y_i, r_i, Z_i)_{i=1}^n, \lambda, S, \beta_{q-1}, \beta_0, c)$
while $t_1 < 30$ *and* $|\zeta_q| == \infty$ **do**
  $\beta_q \leftarrow OIRLS((y_i, r_i, Z_i)_{i=1}^n, \lambda, S, \beta_{q-1}, \beta_0, c)$, $\lambda \leftarrow \lambda \times 2$, $t_1 \leftarrow t_1 + 1$
end
if $t_1 = 30$ **then**
  $\lambda \leftarrow 1$, $t_2 \leftarrow 0$
  while $t_2 < 30$ *and* $|\zeta_q| == \infty$ **do**
    $\beta_q \leftarrow OIRLS((y_i, r_i, Z_i)_{i=1}^n, \lambda, S, \beta_{q-1}, \beta_0, c)$, $\lambda \leftarrow \lambda/2$, $t_2 \leftarrow t_2 + 1$
  end
end
$t_3 \leftarrow 0$, $t_4 \leftarrow 0$
if $-\zeta_q^T M \zeta_q \geq \epsilon$ **then**
  while $-\zeta_q^T M \zeta_q \geq \epsilon$ *and* $t_3 < 30$ **do**
    $\beta_q \leftarrow OIRLS((y_i, r_i, Z_i)_{i=1}^n, \lambda, S, \beta_{q-1}, \beta_0, c)$, $\lambda \leftarrow \lambda \times 2$, $t_3 \leftarrow t_3 + 1$
  end
  $\lambda_{max} \leftarrow \lambda/2$, $\lambda_{min} \leftarrow \lambda/4$
else
  while $-\zeta_q^T M \zeta_q < \epsilon$ *and* $t_4 < 30$ **do**
    $\beta_q \leftarrow OIRLS((y_i, r_i, Z_i)_{i=1}^n, \lambda, S, \beta_{q-1}, \beta_0, c)$, $\lambda \leftarrow \lambda/2$, $t_4 \leftarrow t_4 + 1$
  end
  $\lambda_{max} \leftarrow \lambda \times 2$, $\lambda_{min} \leftarrow \lambda$
end
for λ *in* $\{\exp(\log \lambda_{min} + \frac{\log \lambda_{max} - \log \lambda_{min}}{N-1} i)\}_{i=0}^{N-1}$, $N \leftarrow 30$ **do**
  $\beta, l(\beta) \leftarrow OIRLS((y_i, r_i, Z_i)_{i=1}^n, \lambda, S, \beta_{q-1}, \beta_0, c)$
  if $-\zeta^T M \zeta < \epsilon$ **then**
    if $l(\beta_q) == NULL$ **then**
      $\beta_q \leftarrow \beta$, $-l(\beta_q) \leftarrow -l(\beta)$
    else
      if $-l(\beta) < -l(\beta_q)$ **then**
        $\beta_q \leftarrow \beta$, $-l(\beta_q) \leftarrow -l(\beta)$
      end
    end
  end
end

Algorithm 12: Fine Pruning with a Quadratic Programming Constraint with OIRLST

Data: $(y_i, r_i, Z_i)_{i=1}^n, S = \cup_{j=1}^m S_j = \cup_{h=1}^{|S|} s_h$
Result: $\dot{S}$

while $m - 2 > 0$ **do**
  $[\zeta^T, \alpha^T] \leftarrow IRLSG((y_i, r_i, Z_i)_{i=1}^n, S, \beta_0), AIC_m \leftarrow AIC(\zeta, \alpha), l_m \leftarrow l(\zeta, \alpha), BS_m \leftarrow BS(\zeta, \alpha), \dot{S}^m = S$
  $m \leftarrow m - 1$
  for $j \leftarrow 2 : m$ **do**
    for $c \leftarrow 0 : 2$ **do**
      $q \leftarrow 1, \beta_{m,j,c,0} \leftarrow \beta_0$
      $\beta_{m,j,c,q}, l_{m,j,c,q} \leftarrow OIRLST((y_i, r_i, Z_i)_{i=1}^n, S, \beta_{m,j,c,q-1}, \beta_0, c)$
      while $q <= 10$ **do**
        $\beta_{m,j,c,q+1}, l_{m,j,c,q+1} \leftarrow OIRLST((y_i, r_i, Z_i)_{i=1}^n, S, \beta_{m,j,c,q}, \beta_0, c)$, $q \leftarrow q + 1$
      end
    end
  end
  $J, C, Q = \arg\max_{j,c,q} l_{m,j,c,q}, l_m = l_{m,J,C,Q}, BS_m = BS_{m,J,C,Q}, AIC_m = AIC_{m,J,C,Q}$
  $\dot{S}_1 \leftarrow S_1, \ldots, \dot{S}_{J-1} \leftarrow S_{J-1} \cup S_{J,C,Q}, \dot{S}_J \leftarrow S_{J+1} \cup (S_J \backslash S_{J,C,Q}), \ldots, \dot{S}_m \leftarrow S_{m+1}, \dot{S}^m = \cup_{p=1}^m \dot{S}_p$
  if *the LRT is used* **then**
    $-2(l_m - l_{m+1}) \sim \chi^2(1)$
    if *p-value* $\leq \alpha$ **then**
      $\dot{S} = \dot{S}^{m+1}$
      break
    end
  end
end
if *the BS or AIC is used* **then**
  $\dot{S} = \dot{S}^K, K = \arg\min_m BS_m$ or $K = \arg\min_m AIC_m$
end

2.5 Simulation

2.5.1 Edge Misclassification Rate

Refer to Figure 2.1, which depicts all combinations of risk factors in a tree-like structure. In this structure, each node represents a unique combination, and each edge represents an ordering relationship. One simplistic approach to assess the performance of our algorithms is to assign a stage label to each node, compare the classification with the ground truth, and calculate the proportion of nodes that are incorrectly classified known as the node misclassification rate. However, this metric is not robust as it heavily relies on the total number of stages and the accuracy of early-stage classification. Therefore, we propose the use of the edge misclassification rate as a more reliable measure. The edge misclassification rate quantifies the proportion of incorrectly classified edges.

Mathematically, edge misclassification rate can be denoted as follows

$$1 - \frac{\sum_i^{|E|} \mathbf{1}(e_i^T s - e_i^T t = 0)}{|E|} \tag{2.12}$$

In Equation (2.12), $|E|$ represents the total number of edges, and e_i is a $|S|$-dimensional vector that represents a partial ordering relationship, where $(S, \leq)$ is a poset comprising all combinations of risk factors. For example, if $p, q \in (S, \leq)$ and $p \leq_S q$, then $e_i^T = [0, 0, ..., 0, -1, 0, ..., 0, 1, 0, ..., 0, 0]$, where -1 corresponds to p and 1 corresponds to q. Furthermore, s is a $|S|$-dimensional vector representing the prediction for cancer staging, and t is a $|S|$-dimensional vector representing the ground truth for cancer staging.

2.5.2 Setup

To assess the performance of OPERA and its related methods, we selected lasso tree [LWC13] as the alternative approach for comparison. Although lasso tree was initially designed for survival outcomes with two risk factors, we extended it to handle binary

outcomes and multiple risk factors so that our algorithms could be compared effectively.

For the purpose of simulation, logistic regression models, as shown in Table 2.1, were used to define the underlying probabilities (p_i) of experiencing an outcome for each patient (i), where $0.1 \leq p_i \leq 0.9$. The binary outcome Y_i was simulated based on the binomial distribution with a size of 1 and a probability of p_i for each patient. The assignment of patients to risk categories followed a uniform distribution, ensuring an equal probability of belonging to each category.

Model	Assumptions
$logit(p_i) = \sum_{j=1}^{5} \beta_j \mathbf{1}(X_i \in (S_j, \leq))$	$Y_i \sim BIN(1, p_i)$ $X_i \sim UNIF\{(S, \leq)\}$ $0.1 \leq p_i \leq 0.9$ $i = 1, 2, 3, ..., n$ $\beta = -2, -1, 0, 1, 2$

Table 2.1: The model and assumptions to simulate binary data

The underlying coefficients βs for risk categories defined by two risk factors were pre-specified as illustrated in Figure 2.3. The figure on the left Figure 2.3a represents a scenario that favors lasso tree, as lasso tree is only capable of handling neighboring categories within the same stage. Conversely, the figure on the right Figure 2.3b depicts a scenario that favors OPERA, where non-neighboring categories can belong to the same stage, such as the second stage and the third stage.

In the binary outcome scenario, the sample size n can vary between 1600 and 3200 for two-risk-factor scenarios, and between 2700 and 5400 for three-risk-factor scenarios. For each simulation scenario, a total of 500 simulations were conducted. The two-risk-factor scenarios involved two different 4-level ordinal risk factors, namely A and B, as depicted in Figure 2.3. Alternatively, the three-risk-factor scenarios considered three different 3-level risk factors, denoted as A, B, and C, as shown in Figure 2.4. It is important to note that no additional covariates needed to be adjusted in the simulations.

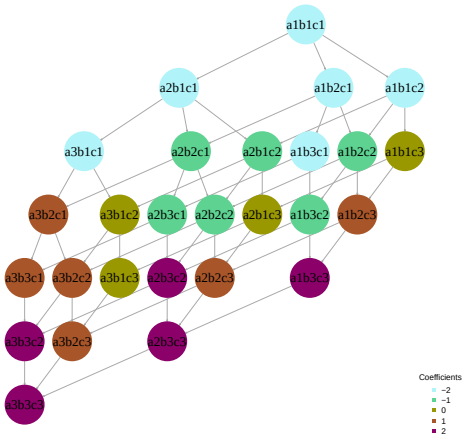
	b1	b2	b3	b4
a1	-2	-2	-2	-1
a2	-1	-1	-1	-1
a3	0	1	1	2
a4	0	1	2	2

(a) Simulation setup favoring lasso tree

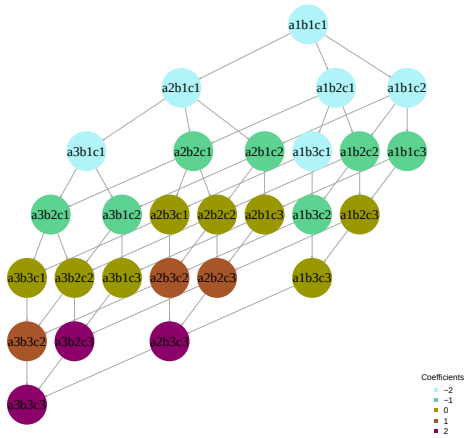
	b1	b2	b3	b4
a1	-2	-2	-2	-1
a2	-1	-1	-1	0
a3	0	1	1	2
a4	0	1	2	2

(b) Simulation setup favoring opera

Figure 2.3: Simulation setup for coefficients βs with two risk factors

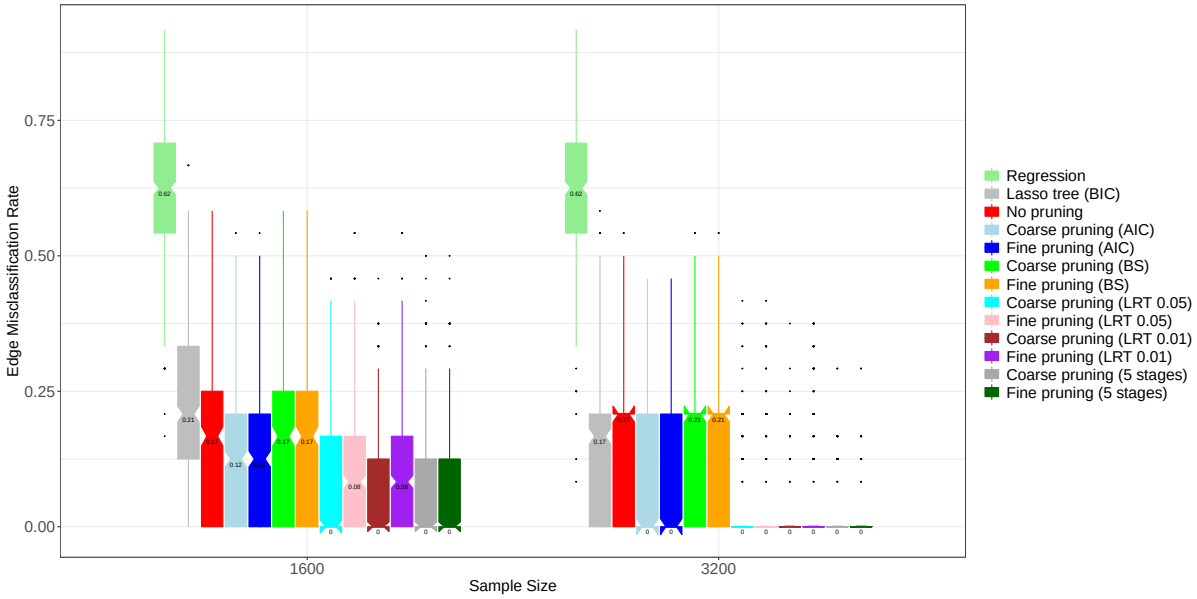


(a) Simulation setup favoring lasso tree

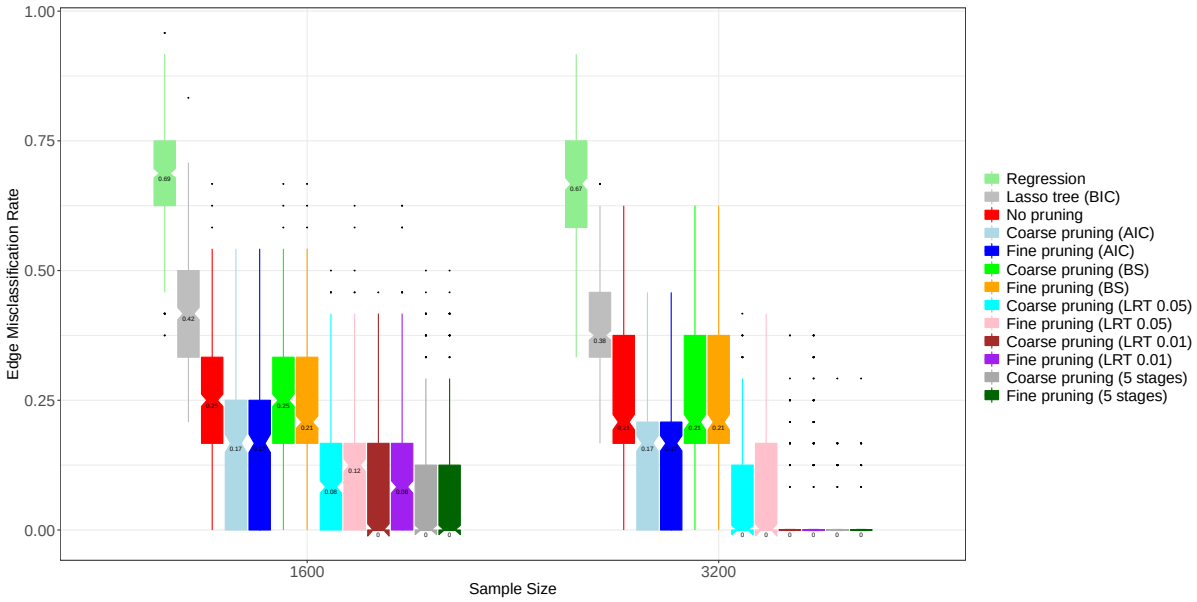


(b) Simulation setup favoring opera

Figure 2.4: Simulation setup for coefficients βs with three risk factors

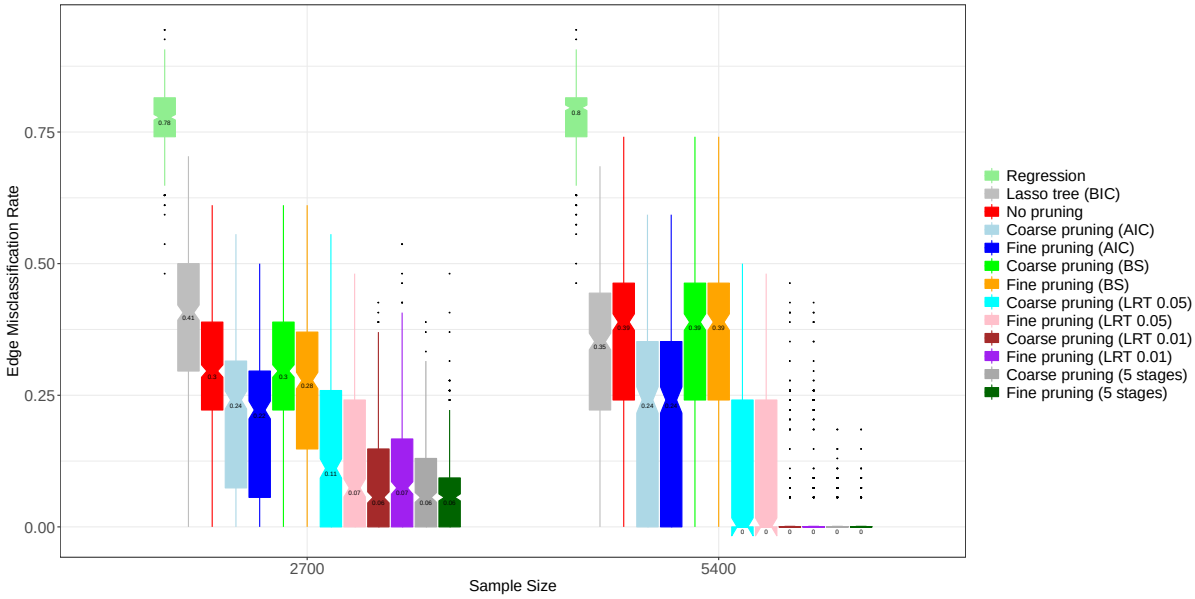


(a) Simulation scenario favoring lasso tree

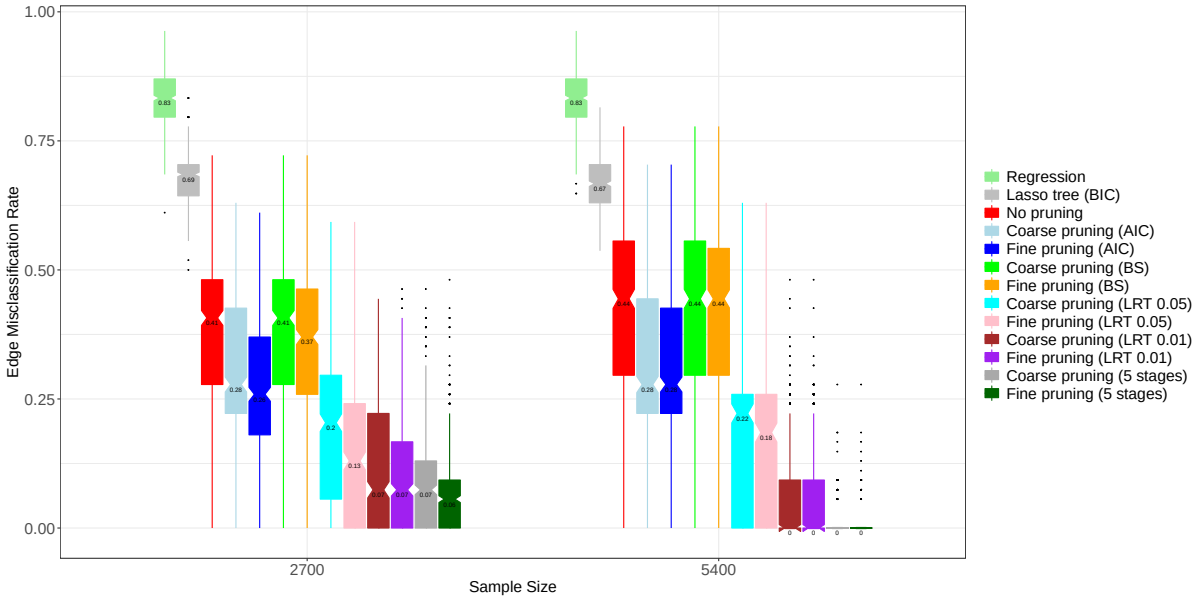


(b) Simulation scenario favoring opera

Figure 2.5: The edge misclassification rate for different methods with two risk factors



(a) Simulation scenario favoring lasso tree



(b) Simulation scenario favoring opera

Figure 2.6: The edge misclassification rate for different methods with three risk factors

2.5.3 Results

The results for various simulation scenarios, including those with two risk factors and three risk factors, are displayed in Figures 2.5 and 2.6, respectively. We also consider various underlying true staging patterns, including the neighbouring staging patterns that favor lasso tree and the non-neighboring staging patterns that favor OPERA. In each simulation scenario, we compared multiple approaches, including the use of a logistic regression model with a partial ordering constraint and the grouping of variables with the same coefficients (after rounding them to the nearest two decimal places) into stages (referred to as *regression*), the application of lasso tree using BIC (referred to as *lasso tree*), the use of OPERA without pruning (referred to as *no pruning*), the implementation of coarse pruning (referred to as *coarse pruning*), and the adoption of fine pruning using exhaustive search (referred to as *fine pruning*). The stopping rule for pruning relies on the criterion used to determine whether a stage should undergo pruning. This criterion can be based on the LRT, where the Type I error rate can be set at either 0.05 or 0.01, the BS, the AIC, or pruning until the true number of stages is reached – in our case, 5 stages.

In three-risk-factor scenarios favoring lasso tree, as illustrated in Figure 2.6a, the regression method yields the poorest performance in the median edge misclassification rate. In contrast, coarse pruning using LRT with $\alpha = 0.01$ demonstrates the best performance among all the criteria without knowing the true number of stages. Also, the result is closely comparable to pruning based on the predefined number of stages with the median edge misclassification rate equal to 0.06 when the sample size equals 2700 and 0 when the sample size equals 5400. This approach leads to an improvement of 0.72 in reducing the median edge misclassification rate compared to regression, and a further enhancement of 0.35 compared to the use of the lasso tree for a sample size of 2700. Similarly, an improvement of 0.8 compared to using regression and 0.35 compared to using lasso tree are observed for a sample size of 5400. Among the other pruning

criteria, using the LRT ($\alpha = 0.05$) yields the best results in the median misclassification rate, followed by the AIC, and then the BS. Fine pruning steps only uses the exhaustive search method here. Fine pruning consistently outperforms or at least performs on par with coarse pruning across various criteria, except for a slightly worse performance (0.01 increase in the median edge misclassification rate) for a sample size of 2700 using LRT ($\alpha = 0.01$). However, with a larger sample size, fine pruning matches the performance of coarse pruning, regardless of the stopping criteria used. The enhancement from OPERA without pruning to pruning using the BS is extremely small (the difference in the median misclassification rate is no more than 0.02), indicating that the BS is not an effective criterion for pruning. While OPERA outperforms lasso tree with a smaller sample size (0.3 vs. 0.41), it fails to do so with a larger sample size (0.39 vs. 0.35), due to the issue of over-partitioning. Nevertheless, pruning substantially enhances performance with a larger sample size by mitigating the over-partitioning issue.

In three-risk-factor scenarios favoring OPERA, as illustrated in Figure 2.6b, similar conclusions can be drawn. The regression method still yields the poorest performance, while coarse pruning using LRT with $\alpha = 0.01$ still demonstrates the best performance in the median misclassification rate, closely comparable to pruning based on the predefined number of stages (0.07 for a sample size of 2700 while 0 for a sample size of 5400). When the sample size is 2700, this approach results in an improvement of 0.76 in reducing the median edge misclassification rate compared to using regression and 0.62 compared to using lasso tree. Similarly, with a sample size of 5400, improvements of 0.83 compared to using regression and 0.67 compared to using lasso tree are observed. Among the other pruning criteria, using the LRT ($\alpha = 0.05$) still yields the best results, followed by the AIC, and then the BS. Fine pruning steps still only use exhaustive search. Fine pruning consistently outperforms or at least performs on par with coarse pruning across different criteria by demonstrating a smaller median misclassification rate or a smaller interquartile range (IQR). The improvement from OPERA without pruning to pruning using the BS

is still extremely small (0 for coarse pruning while 0.04 for fine pruning), indicating that the BS is not an effective criterion for pruning. In both sample sizes, OPERA outperforms lasso tree, as the simulation scenarios favor OPERA with non-neighboring staging patterns.

In two-risk factor scenarios favoring lasso tree, as depicted in Figure 2.5a, lasso tree outperforms OPERA for a sample size of 3200 (0.17 vs 0.21), while the opposite can be observed for a sample size of 1600 (0.21 vs 0.17). It is consistent with the previous results in Figure 2.6a that lasso tree can outperform OPERA only for a larger sample size. Coarse pruning using the LRT ($\alpha = 0.01$) still demonstrates the best performance in the median edge misclassification rate (0 for both sample sizes), comparable to pruning with the predefined number of stages. On the other hand, regression still exhibits the worst performance. Fine pruning does not exhibit better performance than coarse pruning across different stopping rules. In two-risk factor scenarios favoring OPERA, as shown in Figure 2.5b, OPERA without pruning outperforms lasso tree across different sample sizes, contrasting the scenarios favoring lasso tree. Coarse pruning using the LRT ($\alpha = 0.01$) still shows the best performance in the median edge misclassification rate (0 for both sample sizes), comparable to pruning with the predefined number of stages. Fine pruning still does not show a better performance than coarse pruning across different stopping rules, except using the BS for a sample size of 1600.

Across all scenarios, pruning with BS consistently fails to improve accuracy, whereas pruning with LRT ($\alpha = 0.01$) consistently performs the best among all the criteria without knowing the true number of stages, closely comparable to pruning using the predefined number of stages. Coarse pruning achieves performance as good as fine pruning.

2.5.4 Discussion

Given the demonstrated improvements of pruning methods over OPERA without pruning, further exploration of these methods can be undertaken using different initial steps.

These steps involve applying the lasso tree with AIC as the criterion for determining optimal groupings, using the lasso tree with BIC as the default criterion, and utilizing OPERA without pruning. As depicted in Figures 2.7, 2.8, 2.9, and 2.10, while the three initial methods yield different results without pruning, the use of pruning enables all methods to perform well, with low median edge misclassification rates across different simulation scenarios, including different sample sizes, different staging patterns, and different numbers of risk factors.

To further explore which pruning method performs the best across different initial methods, we calculate the mean edge misclassification for each simulation scenario across 500 simulations and select the top 3 pruning methods with the lowest mean edge misclassification rates. The results, including the estimated standard deviation (SD) and 95% confidence intervals, are displayed in Table 2.2 - 2.5. The estimated SD can be calculated using the sample standard deviation, while the confidence interval can be calculated using the mean edge misclassification rate plus or minus $Z_{0.975} = 1.96$ times the sample standard error. In three-risk-factor simulation scenarios, either with neighboring or non-neighboring staging patterns, LRT ($\alpha = 0.01$) is consistently the best stopping rule with the lowest mean edge misclassification rates, as shown in Table 2.2 - 2.3. In two-risk-factor simulation scenarios, either with neighboring or non-neighboring staging patterns, coarse pruning with LRT ($\alpha = 0.01$) remains the best stopping rule with the lowest mean edge misclassification rates, as shown in Table 2.4 - 2.5.

The next question pertains to whether coarse pruning can be as effective as fine pruning with LRT ($\alpha = 0.01$) as the stopping rule. If it is not the best, how much less effective is it? As shown in Table 2.6, we compare the mean misclassification rates among coarse pruning, fine pruning using exhaustive search, and fine pruning using quadratic constraint with LRT ($\alpha = 0.01$) as the stopping rule. Coarse pruning consistently demonstrates a smaller mean misclassification rate across all simulation scenarios, compared to fine pruning using quadratic constraint, given all the negative differences between these

two pruning approaches. Compared to fine pruning with exhaustive search, coarse pruning still shows a smaller mean edge misclassification rate across the majority of simulation scenarios. In cases where fine pruning using exhaustive search performs better, the difference is significantly small or even not significant (p-value ≥ 0.05) after we use the paired sample t-test to analyze the difference. The largest significant difference is 2%, which means only around 1 more edge out of 54 edges is classified correctly compared with coarse pruning. Thus, we still believe coarse pruning demonstrates comparable performance and is preferable due to its lower computational cost.

To evaluate the performance of fine pruning methods with exhaustive search compared to those with quadratic programming constraints, each fine pruning method using each criterion is examined, as illustrated in Figures 2.11, 2.12, 2.13, and 2.14. Across all scenarios, utilizing quadratic programming constraints achieves a low median edge misclassification rate, comparable to using exhaustive search. To further investigate how much using quadratic constraints leads to inferior performance compared to using exhaustive search, we use the paired sample t-test to compare these two approaches. As shown in Table 2.6, with LRT ($\alpha = 0.01$) as the stopping rule, using exhaustive search only demonstrates a small yet significant improvement over using quadratic constraints, with 5.8% as the largest difference in the mean edge misclassification rate. This is roughly equivalent to 1 edges out of 24 edges. In other words, when the computational cost is high with using exhaustive search, using quadratic constraints can be an alternative with comparable performance. Note that, due to 500 simulations for each scenario, even a small difference can be significant.

Given coarse pruning with LRT ($\alpha = 0.01$) as the best pruning approach, different initial methods also have a slight impact on the final result. While it may not be evident from Figures 2.7 - 2.10 based on the median edge misclassification rate, we can still observe that using the lasso tree leads to a better performance in terms of the mean edge misclassification rate when the sample size is smaller, with 2.8% as the largest difference.

However, all three approaches perform very similarly well when the sample size is larger, with less than 1% as the largest difference. In real data analysis, we recommend using the lasso tree as the initial method, not only due to its slightly better performance but also its lower computational cost as shown in Table 2.7, in comparison with OPERA.

Table 2.2: The mean edge misclassification rates for the top 3 pruning methods with different initial methods in three-risk-factor simulation scenarios with non-neighboring staging patterns (Mean = Estimated Mean; SD = Estimated Standard Deviation; Lower = 95% Confidence Interval Lower Limit; Upper = 95% Confidence Interval Upper Limit; quad = using quadratic constraint; ex = using exhaustive search)

Sample Size	Initial Method	Pruning Method	Mean	SD	Lower	Upper
2700	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	0.091	0.108	0.082	0.101
2700	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.101	0.115	0.091	0.112
2700	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	0.136	0.125	0.125	0.147
2700	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	0.094	0.109	0.084	0.103
2700	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.111	0.118	0.100	0.121
2700	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	0.139	0.124	0.128	0.149
2700	OPERA	fine pruning ex (LRT 0.01)	0.100	0.109	0.090	0.109
2700	OPERA	coarse pruning (LRT 0.01)	0.120	0.119	0.109	0.130
2700	OPERA	fine pruning quad (LRT 0.01)	0.137	0.117	0.127	0.147
5400	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	0.063	0.106	0.053	0.072
5400	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.068	0.114	0.058	0.078
5400	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	0.098	0.114	0.088	0.108
5400	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	0.061	0.105	0.051	0.070
5400	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.062	0.109	0.052	0.072
5400	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	0.097	0.115	0.087	0.107
5400	OPERA	fine pruning ex (LRT 0.01)	0.060	0.103	0.051	0.069
5400	OPERA	coarse pruning (LRT 0.01)	0.062	0.109	0.053	0.072
5400	OPERA	fine pruning quad (LRT 0.01)	0.097	0.112	0.087	0.106

Table 2.3: The mean edge misclassification rates for the top 3 pruning methods with different initial methods in three-risk-factor simulation scenarios with neighboring staging patterns

Sample Size	Initial Method	Pruning Method	Mean	SD	Lower	Upper
2700	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.067	0.095	0.059	0.076
2700	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	0.087	0.116	0.077	0.097
2700	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	0.094	0.138	0.082	0.107
2700	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.068	0.096	0.060	0.077
2700	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	0.092	0.119	0.082	0.103
2700	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	0.097	0.139	0.085	0.109
2700	OPERA	coarse pruning (LRT 0.01)	0.095	0.101	0.086	0.104
2700	OPERA	fine pruning ex (LRT 0.01)	0.104	0.114	0.094	0.114
2700	OPERA	fine pruning quad (LRT 0.01)	0.115	0.136	0.103	0.127
5400	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	0.040	0.092	0.032	0.048
5400	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.043	0.095	0.035	0.051
5400	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	0.047	0.108	0.038	0.057
5400	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.036	0.087	0.029	0.044
5400	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	0.037	0.089	0.029	0.045
5400	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	0.043	0.107	0.034	0.053
5400	OPERA	coarse pruning (LRT 0.01)	0.041	0.089	0.033	0.049
5400	OPERA	fine pruning ex (LRT 0.01)	0.041	0.089	0.033	0.048
5400	OPERA	fine pruning quad (LRT 0.01)	0.046	0.103	0.037	0.055

Table 2.4: The mean edge misclassification rates for the top 3 pruning methods with different initial methods in two-risk-factor simulation scenarios with non-neighboring staging patterns

Sample Size	Initial Method	Pruning Method	Mean	SD	Lower	Upper
1600	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.067	0.093	0.059	0.075
1600	Lasso tree (AIC)	coarse pruning (LRT 0.05)	0.083	0.110	0.074	0.093
1600	Lasso tree (AIC)	fine pruning ex (LRT 0.05)	0.089	0.116	0.079	0.099
1600	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.072	0.097	0.064	0.081
1600	Lasso tree (BIC)	coarse pruning (LRT 0.05)	0.090	0.114	0.080	0.100
1600	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	0.098	0.124	0.087	0.108
1600	OPERA	coarse pruning (LRT 0.01)	0.082	0.097	0.073	0.090
1600	OPERA	coarse pruning (LRT 0.05)	0.098	0.112	0.089	0.108
1600	OPERA	fine pruning ex (LRT 0.01)	0.105	0.124	0.094	0.116
3200	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.015	0.052	0.011	0.020
3200	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	0.026	0.075	0.019	0.032
3200	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	0.052	0.118	0.042	0.063
3200	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.016	0.052	0.011	0.020
3200	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	0.026	0.075	0.019	0.032
3200	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	0.053	0.118	0.043	0.063
3200	OPERA	coarse pruning (LRT 0.01)	0.018	0.057	0.013	0.022
3200	OPERA	fine pruning ex (LRT 0.01)	0.028	0.078	0.021	0.035
3200	OPERA	fine pruning quad (LRT 0.01)	0.054	0.118	0.044	0.065

Table 2.5: The mean edge misclassification rates for the top 3 pruning methods with different initial methods in two-risk-factor simulation scenarios with neighboring staging patterns

Sample Size	Initial Method	Pruning Method	Mean	SD	Lower	Upper
1600	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.061	0.091	0.053	0.069
1600	Lasso tree (AIC)	coarse pruning (LRT 0.05)	0.072	0.102	0.063	0.081
1600	Lasso tree (AIC)	fine pruning ex (LRT 0.05)	0.089	0.119	0.079	0.100
1600	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.069	0.100	0.061	0.078
1600	Lasso tree (BIC)	coarse pruning (LRT 0.05)	0.080	0.110	0.071	0.090
1600	Lasso tree (BIC)	fine pruning ex (LRT 0.05)	0.101	0.126	0.090	0.112
1600	OPERA	coarse pruning (LRT 0.01)	0.075	0.096	0.067	0.084
1600	OPERA	coarse pruning (LRT 0.05)	0.086	0.109	0.077	0.096
1600	OPERA	fine pruning ex (LRT 0.05)	0.103	0.121	0.092	0.114
3200	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.012	0.045	0.008	0.016
3200	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	0.028	0.081	0.021	0.035
3200	Lasso tree (AIC)	coarse pruning (LRT 0.05)	0.045	0.084	0.037	0.052
3200	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.012	0.045	0.008	0.016
3200	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	0.028	0.080	0.021	0.035
3200	Lasso tree (BIC)	coarse pruning (LRT 0.05)	0.040	0.081	0.033	0.048
3200	OPERA	coarse pruning (LRT 0.01)	0.015	0.050	0.011	0.019
3200	OPERA	fine pruning ex (LRT 0.01)	0.031	0.083	0.023	0.038
3200	OPERA	coarse pruning (LRT 0.05)	0.041	0.085	0.034	0.049

Table 2.6: The pairwise comparisons in the mean misclassification rate among coarse pruning, fine pruning using exhaustive search and fine pruning using quadratic constraint with LRT ($\alpha = 0.01$) (Each cell displays the difference with the corresponding p-value in parentheses)

	Sample Size	Initial Method	Coarse vs Ex	Coarse vs Quad	Ex vs Quad
Three Risk Factors & Non-neighboring	2700	OPERA	**0.020** (< 0.001)	-0.017 (< 0.001)	-0.037 (< 0.001)
	2700	Lasso tree (AIC)	0.010 (0.01)	-0.035 (< 0.001)	-0.045 (< 0.001)
	2700	Lasso tree (BIC)	0.017 (< 0.001)	-0.028 (< 0.001)	-0.045 (< 0.001)
	5400	OPERA	0.002 (0.302)	-0.034 (< 0.001)	-0.037 (< 0.001)
	5400	Lasso tree (AIC)	0.005 (0.062)	-0.030 (< 0.001)	-0.035 (< 0.001)
	5400	Lasso tree (BIC)	0.001 (0.684)	-0.035 (< 0.001)	-0.036 (< 0.001)
Three Risk Factors & Neighboring	2700	OPERA	-0.009 (0.023)	-0.021 (< 0.001)	-0.011 (0.047)
	2700	Lasso tree (AIC)	-0.020 (< 0.001)	-0.027 (< 0.001)	-0.007 (0.248)
	2700	Lasso tree (BIC)	-0.024 (< 0.001)	-0.029 (< 0.001)	-0.004 (0.488)
	5400	OPERA	0.000 (0.896)	-0.005 (0.181)	-0.005 (0.166)
	5400	Lasso tree (AIC)	0.003 (0.259)	-0.004 (0.255)	-0.007 (0.050)
	5400	Lasso tree (BIC)	0.000 (0.869)	-0.007 (0.071)	-0.007 (0.089)
Two Risk Factors & Non-neighboring	1600	OPERA	-0.024 (< 0.001)	-0.068 (< 0.001)	-0.045 (< 0.001)
	1600	Lasso tree (AIC)	-0.023 (< 0.001)	-0.081 (< 0.001)	**-0.058** (< 0.001)
	1600	Lasso tree (BIC)	-0.025 (< 0.001)	-0.082 (< 0.001)	-0.056 (< 0.001)
	3200	OPERA	-0.010 (< 0.001)	-0.037 (< 0.001)	-0.027 (< 0.001)
	3200	Lasso tree (AIC)	-0.010 (< 0.001)	-0.037 (< 0.001)	-0.027 (< 0.001)
	3200	Lasso tree (BIC)	-0.010 (< 0.001)	-0.037 (< 0.001)	-0.027 (< 0.001)
Two Risk Factors & Neighboring	1600	OPERA	-0.030 (< 0.001)	-0.047 (< 0.001)	-0.017 (< 0.001)
	1600	Lasso tree (AIC)	-0.034 (< 0.001)	-0.060 (< 0.001)	-0.026 (< 0.001)
	1600	Lasso tree (BIC)	-0.037 (< 0.001)	-0.060 (< 0.001)	-0.024 (< 0.001)
	3200	OPERA	-0.016 (< 0.001)	-0.032 (< 0.001)	-0.016 (< 0.001)
	3200	Lasso tree (AIC)	-0.016 (< 0.001)	-0.032 (< 0.001)	-0.016 (< 0.001)
	3200	Lasso tree (BIC)	-0.016 (< 0.001)	-0.032 (< 0.001)	-0.017 (< 0.001)

Table 2.7: The average time for each initial method along with coarse pruning with LRT(0.01)

	Sample Size	Initial Method	Pruning Method	Average (s)	Total (s)
	2700	Lasso tree (AIC)	coarse pruning	373	
	2700	Lasso tree (AIC)	no pruning	177	550
	2700	Lasso tree (BIC)	coarse pruning	332	
	2700	Lasso tree (BIC)	no pruning	201	533
	2700	OPERA	coarse pruning	151	
Three Risk Factors	2700	OPERA	no pruning	503	654
& Non-neighboring	5400	Lasso tree (AIC)	coarse pruning	1546	
	5400	Lasso tree (AIC)	no pruning	700	2246
	5400	Lasso tree (BIC)	coarse pruning	1172	
	5400	Lasso tree (BIC)	no pruning	698	1870
	5400	OPERA	coarse pruning	397	
	5400	OPERA	no pruning	1953	2350
	2700	Lasso tree (AIC)	coarse pruning	168	
	2700	Lasso tree (AIC)	no pruning	192	360
	2700	Lasso tree (BIC)	coarse pruning	109	
	2700	Lasso tree (BIC)	no pruning	210	319
	2700	OPERA	coarse pruning	63	
Three Risk Factors	2700	OPERA	no pruning	469	531
& Neighboring	5400	Lasso tree (AIC)	coarse pruning	487	
	5400	Lasso tree (AIC)	no pruning	661	1148
	5400	Lasso tree (BIC)	coarse pruning	268	
	5400	Lasso tree (BIC)	no pruning	709	978
	5400	OPERA	coarse pruning	262	
	5400	OPERA	no pruning	1568	1830

Table 2.7: The average time for each initial method along with coarse pruning with LRT(0.01)

	Sample Size	Initial Method	Pruning Method	Average (s)	Total (s)
	1600	Lasso tree (AIC)	coarse pruning	50	
	1600	Lasso tree (AIC)	no pruning	39	89
	1600	Lasso tree (BIC)	coarse pruning	39	
	1600	Lasso tree (BIC)	no pruning	40	79
	1600	OPERA	coarse pruning	20	
Two Risk Factors &	1600	OPERA	no pruning	89	110
Non-neighboring	3200	Lasso tree (AIC)	coarse pruning	129	
	3200	Lasso tree (AIC)	no pruning	107	236
	3200	Lasso tree (BIC)	coarse pruning	133	
	3200	Lasso tree (BIC)	no pruning	129	263
	3200	OPERA	coarse pruning	89	
	3200	OPERA	no pruning	392	481
	1600	Lasso tree (AIC)	coarse pruning	34	
	1600	Lasso tree (AIC)	no pruning	42	76
	1600	Lasso tree (BIC)	coarse pruning	36	
	1600	Lasso tree (BIC)	no pruning	45	81
	1600	OPERA	coarse pruning	44	
Two Risk Factors &	1600	OPERA	no pruning	151	195
Neighboring	3200	Lasso tree (AIC)	coarse pruning	106	
	3200	Lasso tree (AIC)	no pruning	125	230
	3200	Lasso tree (BIC)	coarse pruning	74	
	3200	Lasso tree (BIC)	no pruning	124	198
	3200	OPERA	coarse pruning	73	
	3200	OPERA	no pruning	344	417

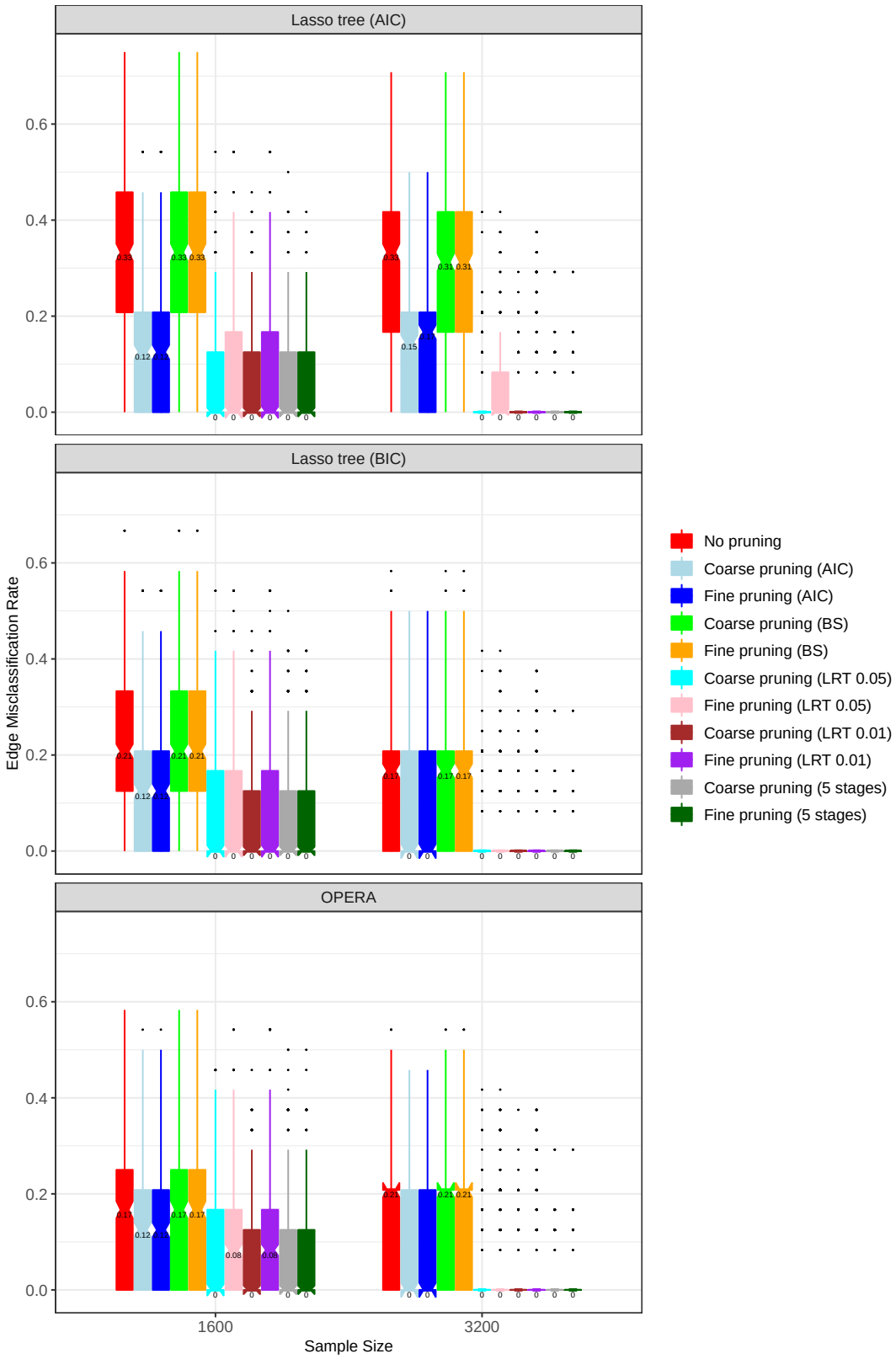


Figure 2.7: The edge misclassification rate for simulation scenarios with two risk factors favoring lasso tree

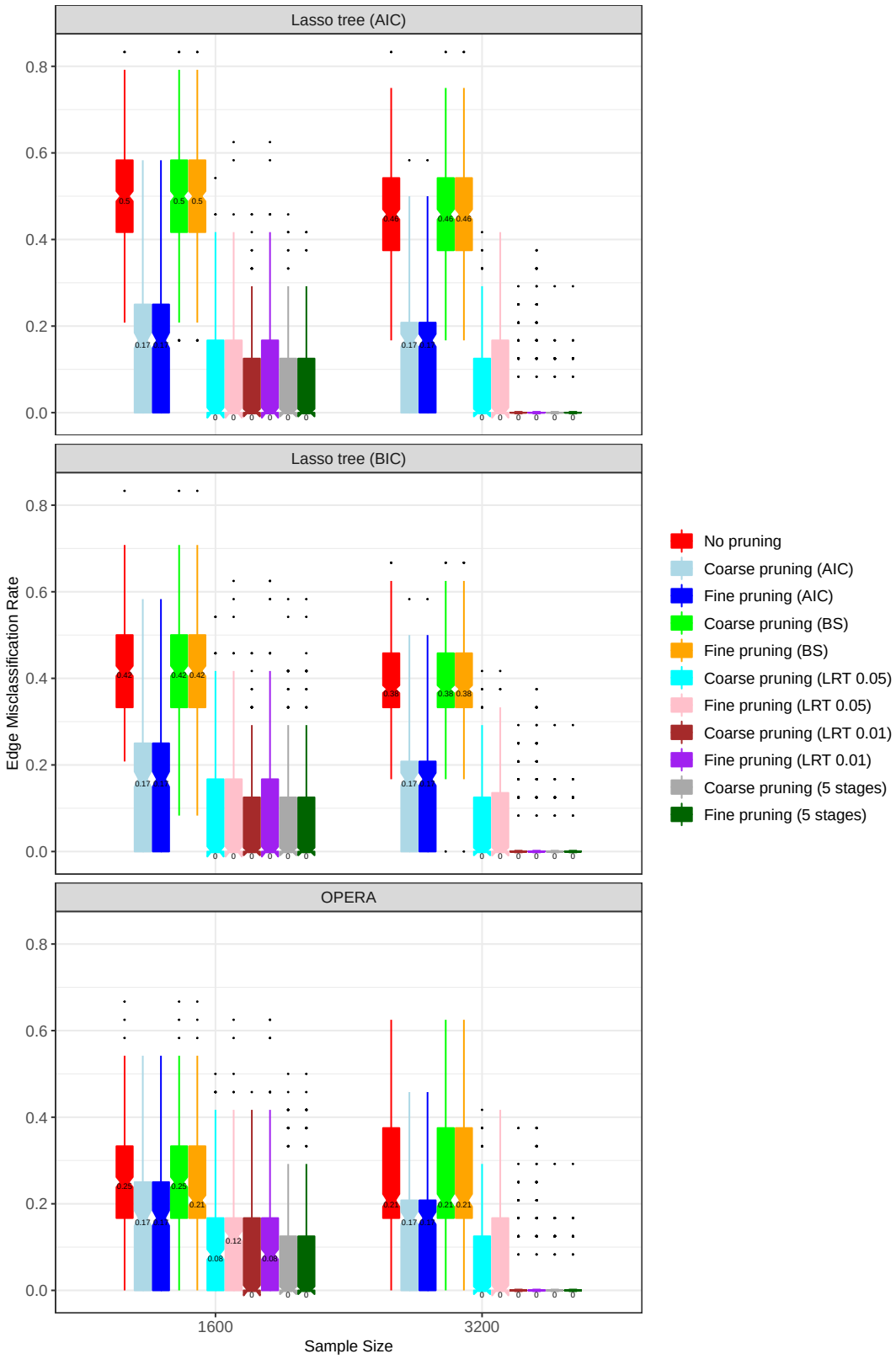


Figure 2.8: The edge misclassification rate for simulation scenarios with two risk factors favoring opera

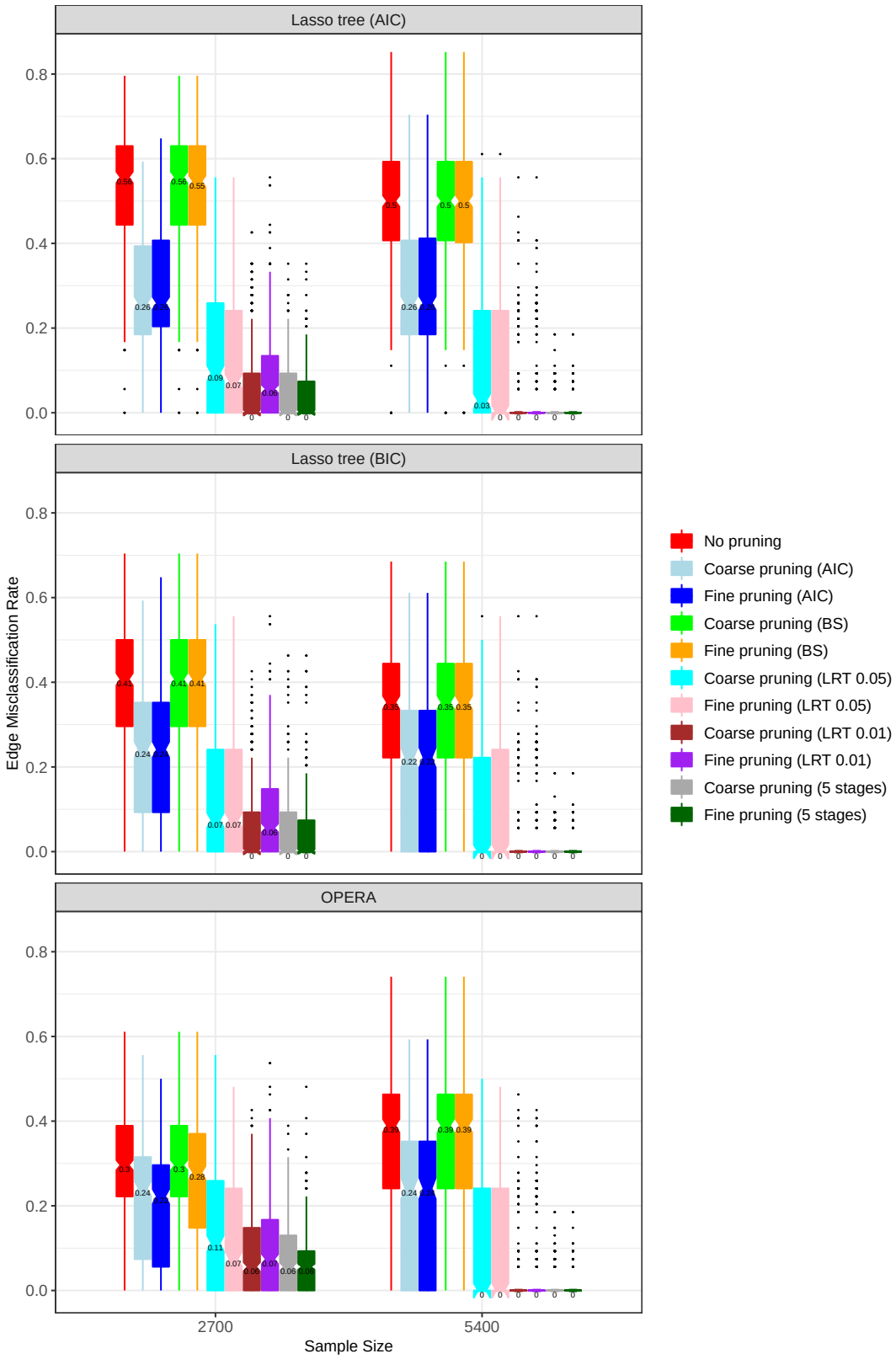


Figure 2.9: The edge misclassification rate for simulation scenarios with three risk factors favoring lasso tree

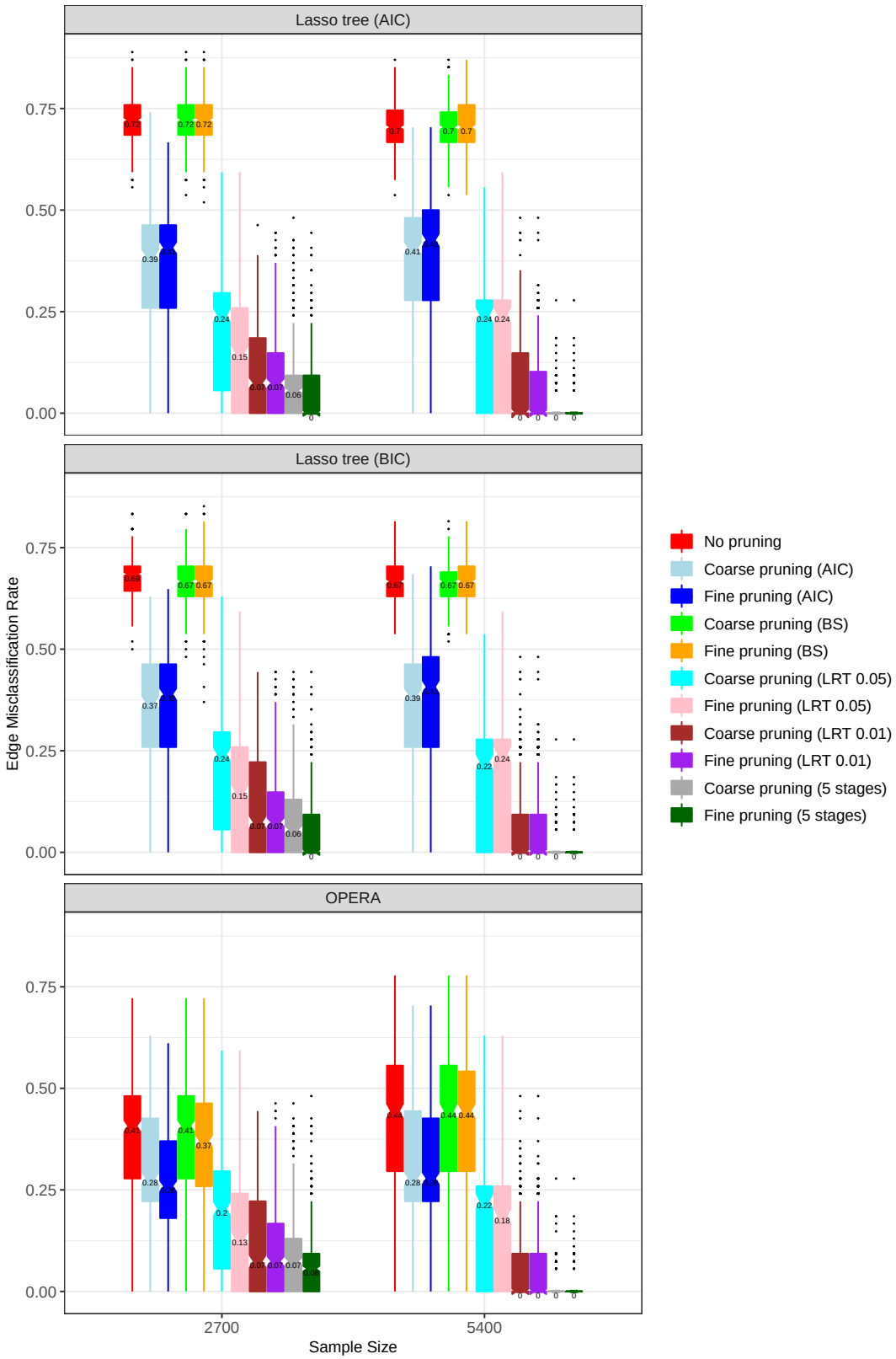


Figure 2.10: The edge misclassification rate for simulation scenarios with three risk factors favoring opera

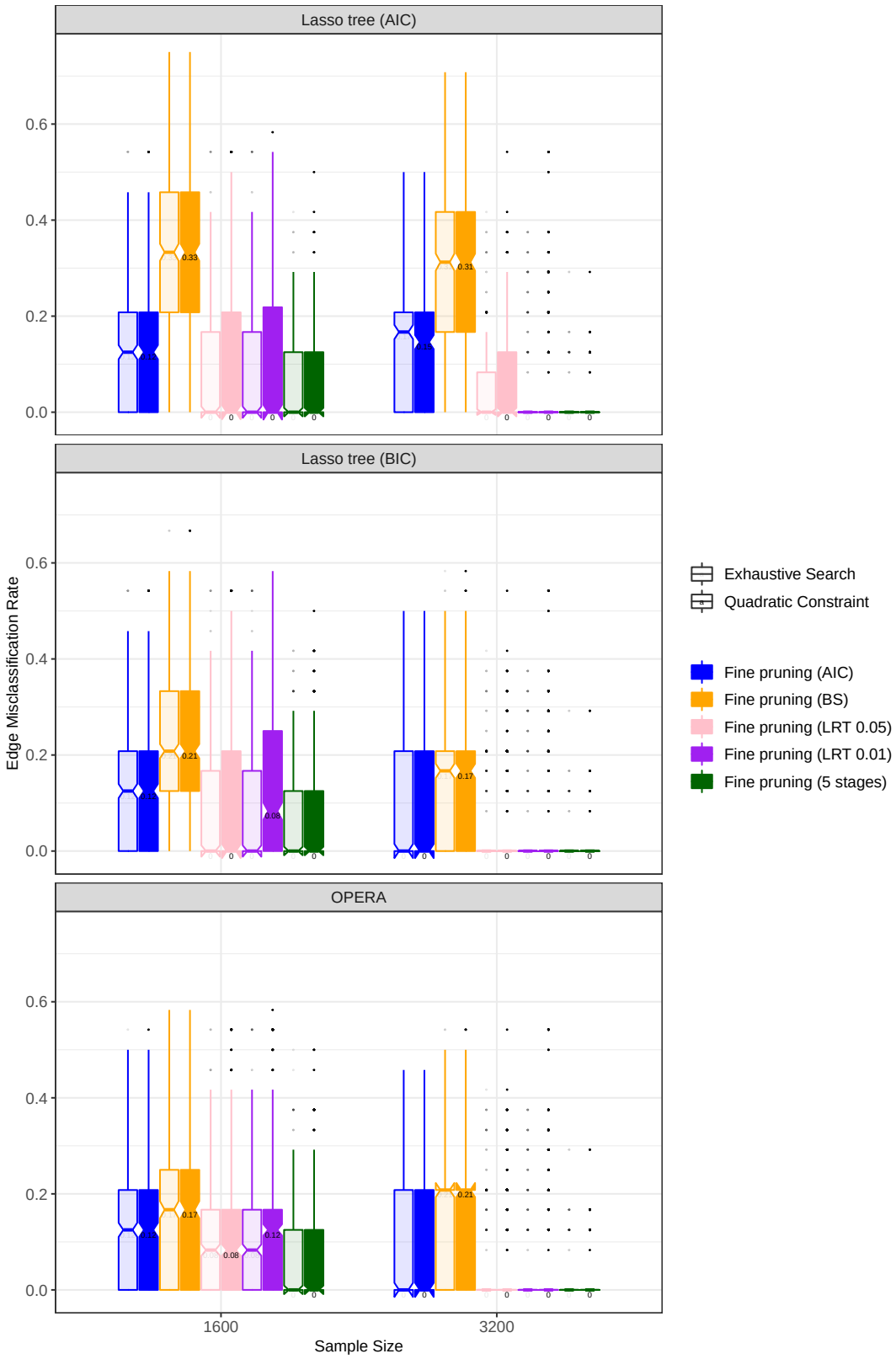


Figure 2.11: The edge misclassification rate for fine pruning methods with two-risk-factor simulation scenarios favoring lasso tree

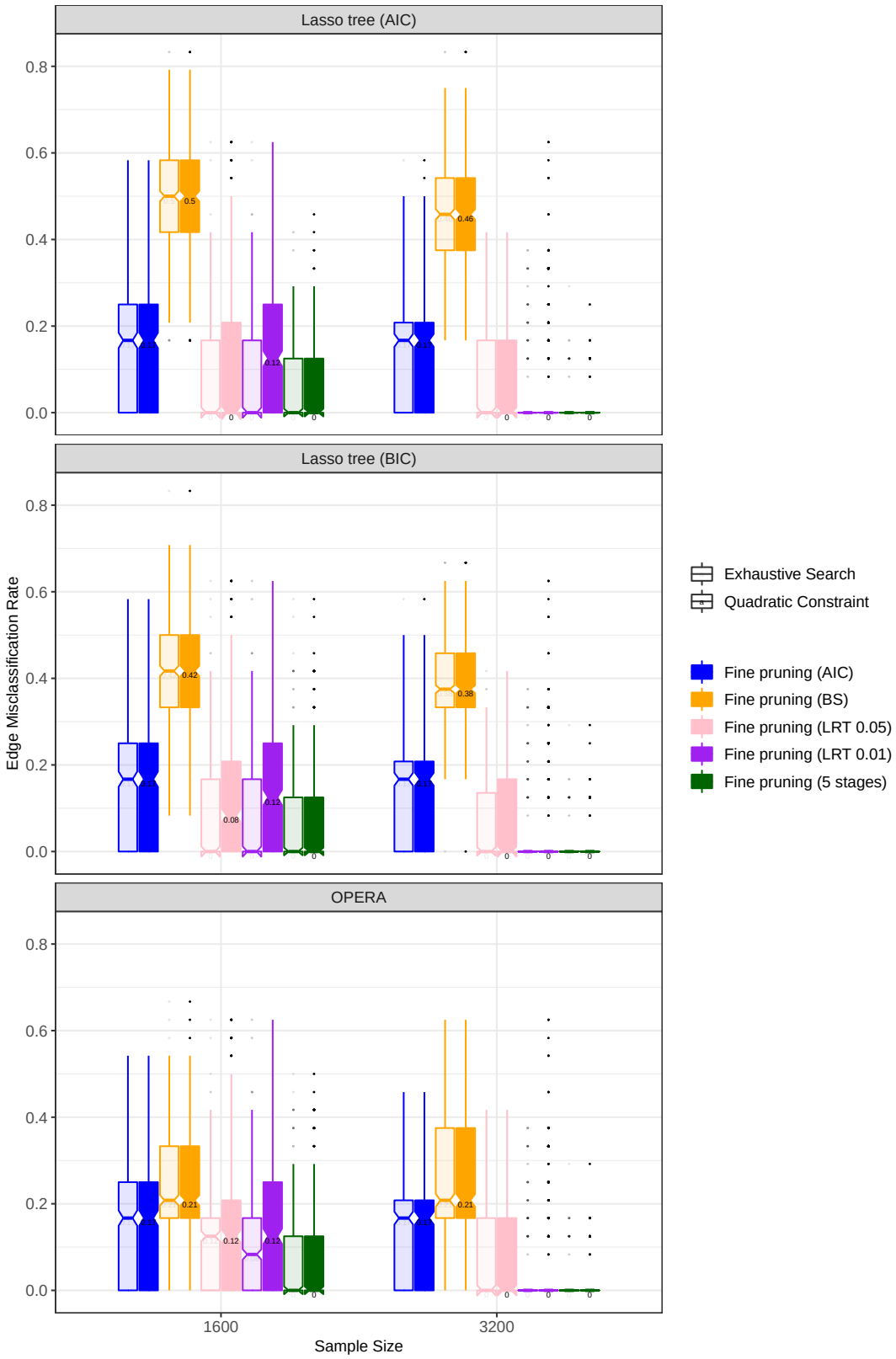


Figure 2.12: The edge misclassification rate for fine pruning methods with two-risk-factor simulation scenarios favoring opera

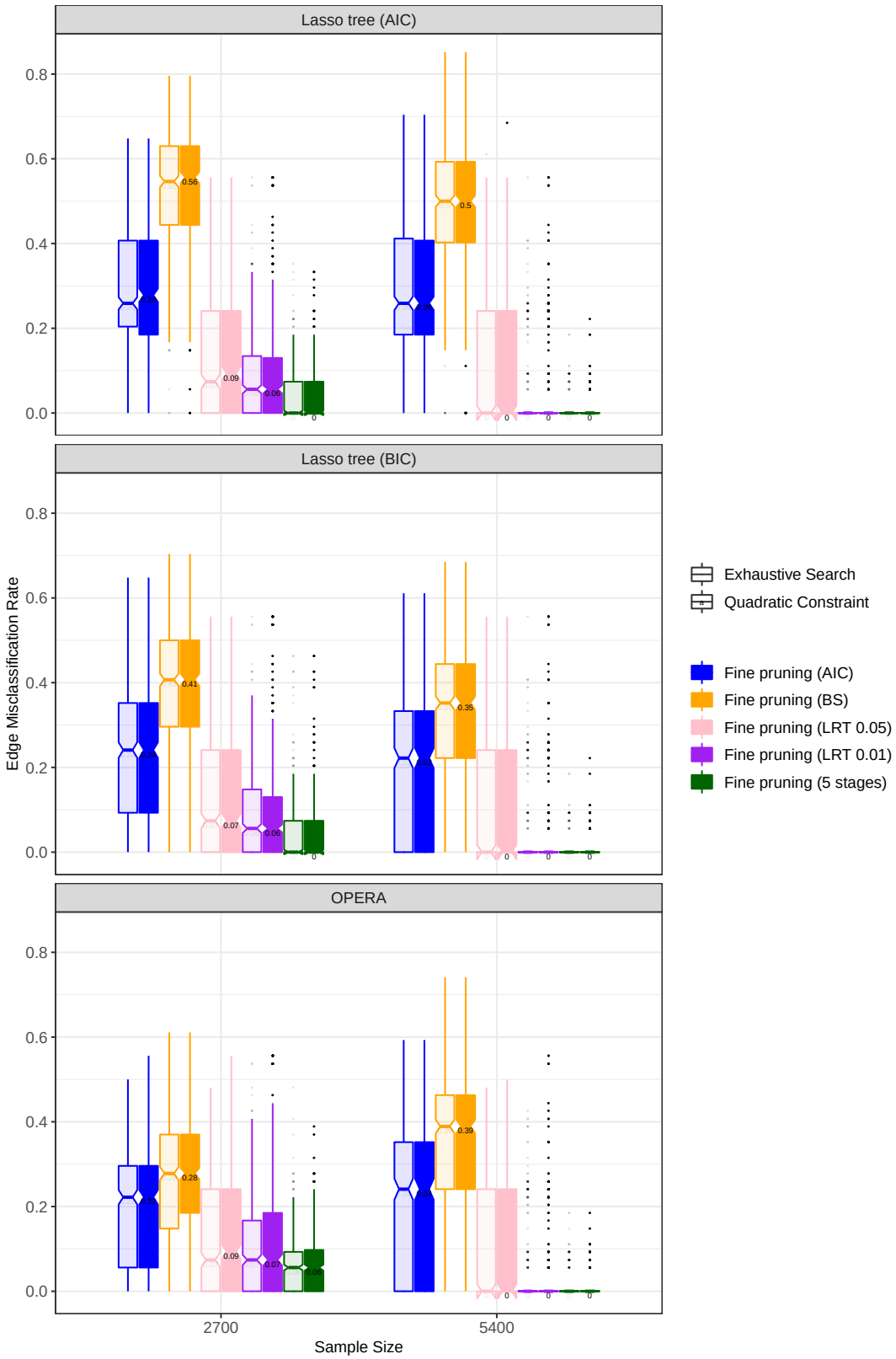


Figure 2.13: The edge misclassification rate for fine pruning methods with three-risk-factor simulation scenarios favoring lasso tree

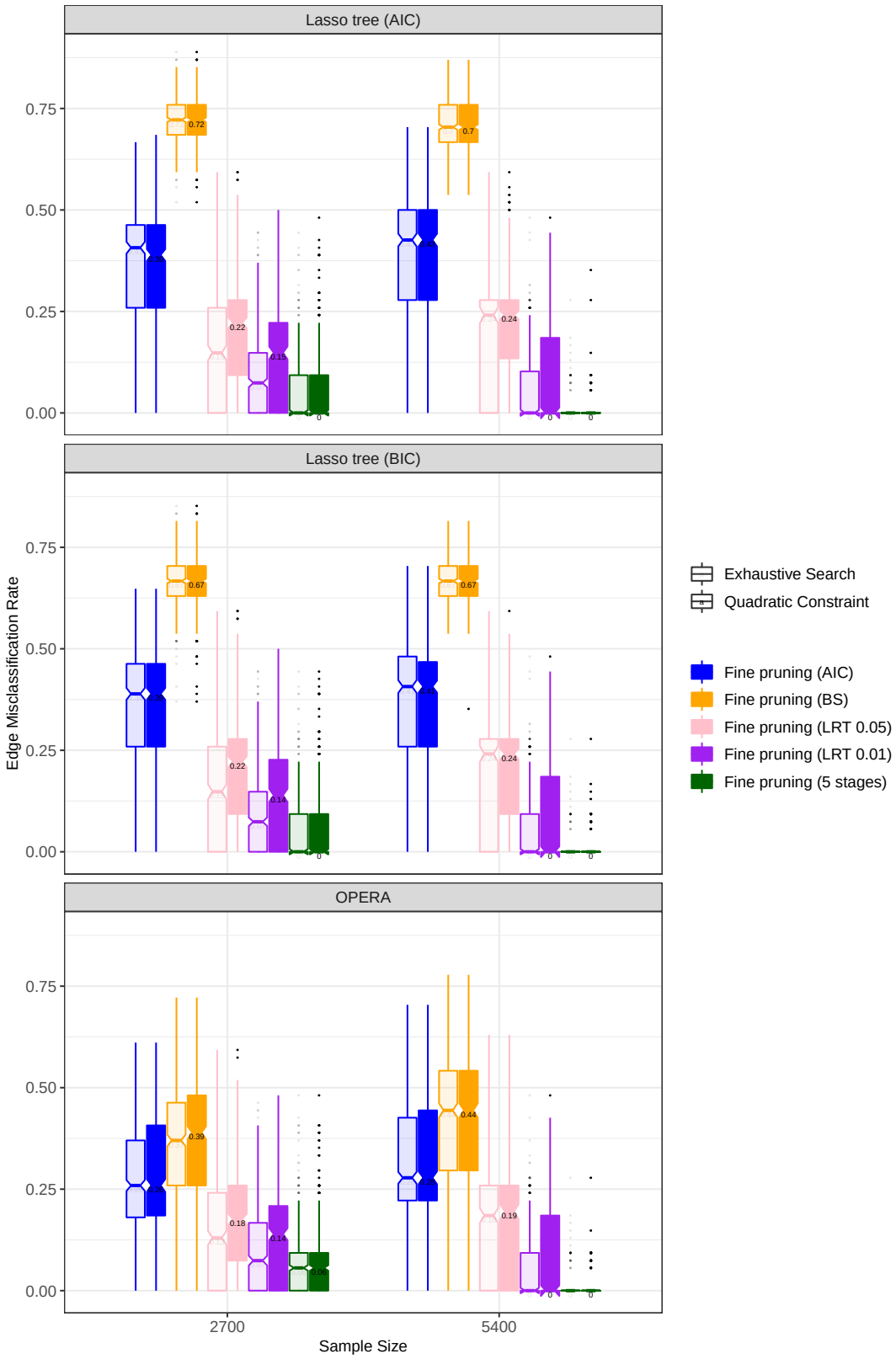


Figure 2.14: The edge misclassification rate for fine pruning methods with three-risk-factor simulation scenarios favoring opera

Chapter 3

Survival Outcome

This chapter provides a comprehensive overview of OPERA and its application in analyzing survival outcomes that involve multiple risk factors. Similar to binary outcomes, a pruning step is integrated to improve an excessively detailed staging system and counteract overfitting. The primary difference between OPERA applied to binary outcomes and survival outcomes resides in the formulation of the (partial) likelihood function and its derivatives utilized in the optimization process. Hence, this chapter begins with an introduction to the partial likelihood function, followed by simulation studies aimed at assessing the performance of OPERA when employed with survival data.

3.1 The Partial Likelihood for Survival Data

When presented with survival data $(y_i, \delta_i, r_i, Z_i)_{i=1}^n$, where each subject i has observed data including the survival outcome y_i with δ_i indicating censorship status, the risk category r_i indicating the subject's category, and covariates $Z_i \in \mathbb{R}^p$, the logarithm of the partial likelihood can be expressed as follows under a Cox proportional hazards model:

$$l(\xi, \alpha) = \sum_{i \in \{i:\delta_i=1\}} [(\xi(r_i) + Z_i^T \alpha) - \log \sum_{j \in \{j:y_i \leq y_j\}} \exp(\xi(r_j) + Z_j^T \alpha)] \tag{3.1}$$

To establish a fixed reference level, an intercept $\mu \in \mathbb{R}$ is introduced, where $\gamma(r) = \xi(r) - \mu$. If $\gamma(r) = 0$, the element r can be amalgamated into the minimum element (if it exists) l with $\gamma(l) = 0$. Using the equation $\xi(r) = \gamma(r) + \mu$, the lasso-type modeling for survival outcomes can be defined as follows:

$$\begin{aligned}\underset{\mu,\gamma,\alpha,\gamma(l)=0}{\arg\min} -l(\mu,\gamma,\alpha) + \lambda\sum_{r\in S}|\gamma(r)| = \underset{\mu,\gamma,\alpha,\gamma(l)=0}{\arg\min}\{-\sum_{i:\delta_i=1}[(\mu+\gamma(r_i)+Z_i^T\alpha)- \\ \log\sum_{j\in R_i=\{j:y_i\le y_j\}}\exp(\mu+\gamma(r_j)+Z_j^T\alpha)] + \lambda\sum_{r\in S}|\gamma(r)|\}\end{aligned} \tag{3.2}$$

In cases where there is no unique minimum, similar to when there is a unique minimum element, the reference level is selected from the set of minimal elements. Based on $\xi(r) = \gamma(r) + \mu$, the lasso-type modeling for survival outcomes can be defined as follows:

$$\begin{aligned}\underset{\mu,\gamma,\alpha}{\arg\min} -l(\mu,\gamma,\alpha) + \lambda\sum_{r\in S}|\gamma(r)| = \underset{\mu,\gamma,\alpha}{\arg\min}\{-\sum_{i:\delta_i=1}[(\mu+\gamma(r_i)+Z_i^T\alpha)- \\ \log\sum_{j\in R_i=\{j:y_i\le y_j\}}\exp(\mu+\gamma(r_j)+Z_j^T\alpha)] + \lambda\sum_{r\in S}|\gamma(r)|\}\end{aligned} \tag{3.3}$$

3.2 Optimization

To solve the $L1$ penalized Cox model[LWC13], an iterative procedure can be employed, similar to the $L1$ penalized logistic regression model. This procedure consists of expressing the standard Newton-Raphson update as an IRLS step, followed by replacing the weighted least squares step with a constrained weighted least squares procedure. The necessary derivatives for computation are as follows:

$$\begin{aligned} u &= \frac{\partial l}{\partial \eta} \\ u_i &= \delta_i - \exp(\eta_i)\sum_{k\in C_i}\frac{1}{\sum_{j\in R_k}\exp(\eta_j)} \\ C_i &= \{k : i \in R_k\} \\ R_k &= \{l : y_k \le y_l\} \end{aligned} \tag{3.4}$$

$$\begin{aligned} A &= \frac{-\partial^2 l}{\partial\eta\eta^T} \\ \frac{\partial^2 l}{\partial\eta_i^2} &= u_i - \delta_i + \exp(2\eta_i)\sum_{k\in C_i}\frac{1}{\{\sum_{j\in R_k}\exp(\eta_j)\}^2} \\ z &= \eta + A^{-1}u \end{aligned} \tag{3.5}$$

This computation requires $\mathcal{O}(n^3)$ operations, where A is a full matrix. To expedite the computation, A can be substituted with a diagonal matrix D having diagonal entries equal to $-\frac{\partial l^2}{\partial \eta_i^2}$[Has17].

3.3 Simulation

3.3.1 Setup

To evaluate the performance of OPERA on survival outcomes and compare it with related methods, we chose lasso tree as the alternative approach. Although lasso tree was originally developed for survival outcomes with only two risk factors, we extended its application to handle survival outcomes with multiple risk factors. This allowed for an effective comparison of our algorithms.

For simulation purposes, we utilized Cox proportional hazards models, as depicted in Table 3.1, to generate the failure time and censoring time, both of which followed an exponential distribution. The parameter λ_j was chosen to achieve the desired proportion of censored patients, denoted as δ_p. The survival outcome Y_i was simulated based on the shorter time between the failure time and censoring time for each patient. To avoid cases with no censoring in advanced stages, the mean of the exponential distribution used to simulate the censoring time decreased as the stage advanced. The assignment of patients to risk categories followed a uniform distribution, ensuring an equal probability of belonging to each category.

Model	Assumptions
$\log(\lambda_i) = \sum_{j=1}^{5} \beta_j \mathbf{1}(X_i \in S_j)$	$T_i \sim EXP(\lambda_i)$ $C_i \sim EXP(\exp(\lambda_j) \times (1 - \frac{1}{s_i+1}))$ $Y_i = \min\{T_i, C_i\}$ $\delta_i = \mathbf{1}(T_i \leq C_i)$ $\delta_p = 1 - \frac{1}{n}\sum_i \delta_i$ $X_i \sim UNIF\{S\}$ $i = 1, 2, 3, ..., n$ $s_i \in \{1, 2, 3, 4, 5\}$ $\beta = 0, 1, 2, 3, 4$

Table 3.1: The model and assumptions to simulate survival data

The coefficients for risk categories defined by two risk factors, denoted as βs, were predetermined and displayed in Figure 3.1. The left figure, Figure 3.1a, illustrates a scenario that benefits lasso tree, as it can effectively handle neighboring categories within the same stage. On the other hand, the right figure, Figure 3.1b, portrays a scenario that favors OPERA, as it allows non-neighboring categories to belong to the same stage, such as the second stage.

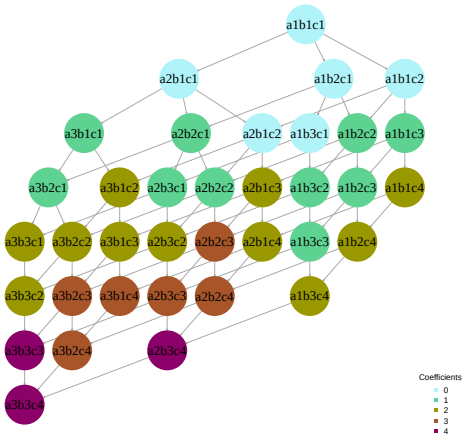
	b1	b2	b3	b4
a1	0	0	0	1
a2	1	1	1	1
a3	2	3	3	4
a4	2	3	4	4

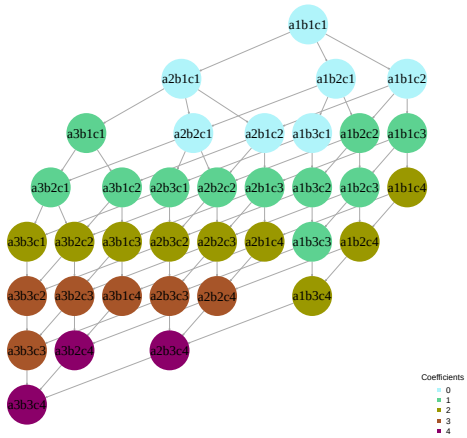
(a) Simulation setup favoring lasso tree

	b1	b2	b3	b4
a1	0	0	0	1
a2	1	1	1	2
a3	2	3	3	4
a4	2	3	4	4

(b) Simulation setup favoring opera

Figure 3.1: Simulation setup for coefficients βs with two risk factors





(a) Simulation setup favoring lasso tree

(b) Simulation setup favoring opera

Figure 3.2: Simulation setup for coefficients βs with three risk factors

In the survival outcome scenarios, the sample size (n) ranged from 800 to 1600 for the two-risk-factor scenarios, and from 1800 to 3600 for the three-risk-factor scenarios. The censoring proportions varied between 0.5 and 0.8. A total of 500 simulations were conducted for each scenario.

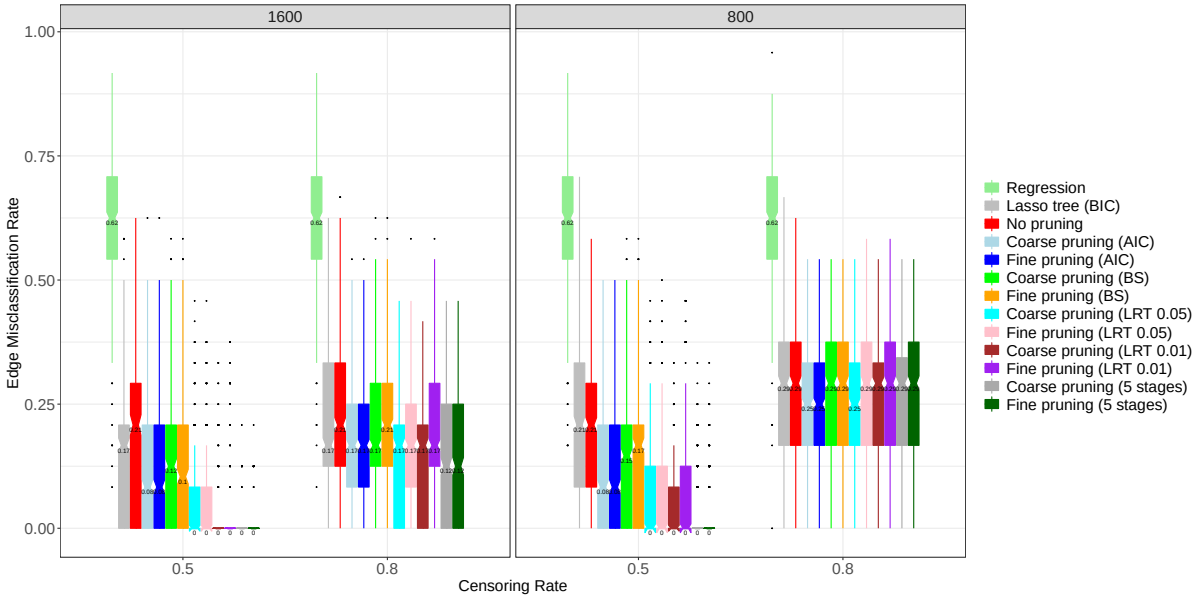
For the two-risk-factor scenarios, two 4-level ordinal risk factors, namely A and B, were considered, as illustrated in Figure 3.1. Conversely, the three-risk-factor scenarios involved three different 3-level risk factors, denoted as A, B, and C, as depicted in Figure 3.2. Each stage is represented by a different color. It is important to note that no additional covariates needed to be adjusted in the simulations.

3.3.2 Results

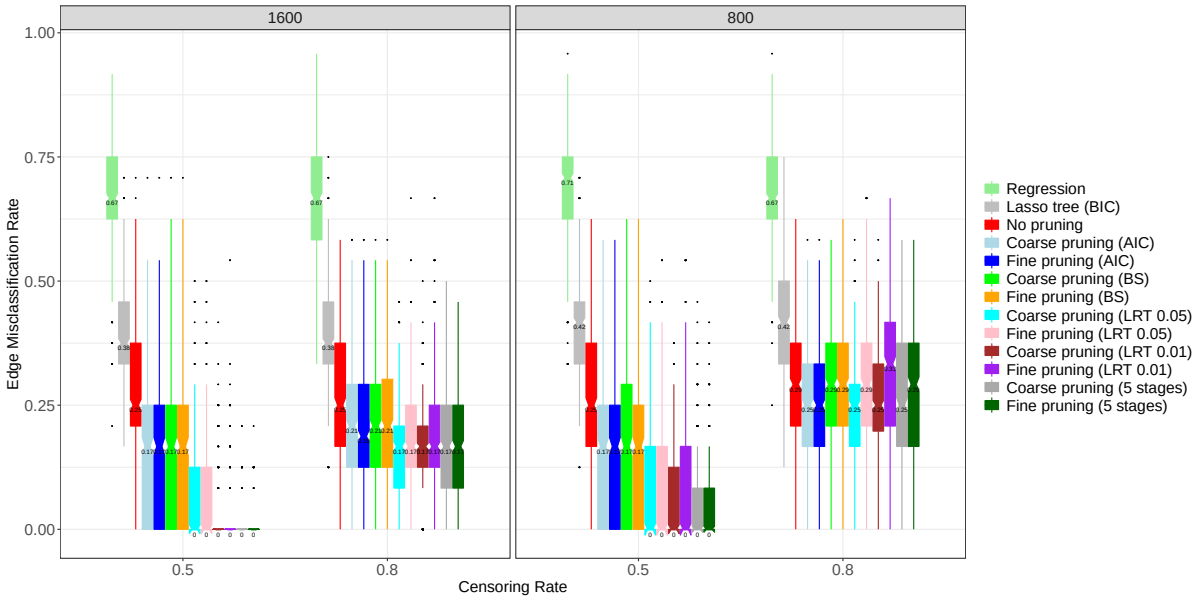
The results for different simulation scenarios with survival outcomes, including two risk factors and three risk factors, are presented in Figure 3.3 and 3.4, respectively. We still compare the same set of approaches with the same set of stopping rules. The only difference lies in the Brier Score (BS). Instead of using the mean squared error, we use the integrated Brier Score (IBS) [Gra+99].

In three-risk-factor scenarios favoring the lasso tree, as shown in Figure 3.4a, the regression method performs the worst in terms of the median edge misclassification rate, while pruning using LRT ($\alpha = 0.01$) demonstrates the best performance, comparable to pruning based on the predefined number of stages across different sample sizes and censoring proportions. Fine pruning with LRT ($\alpha = 0.01$) yields notable improvements in reducing the median edge misclassification rate: 0.33 for OPERA without pruning and 0.56 for the lasso tree when the sample size is 1800 and the censoring rate is 0.5. For a censoring rate of 0.8, the improvements are 0.09 for OPERA without pruning and 0.33 for the lasso tree. The accuracy decreases as the censoring proportion increases due to a lack of patients experiencing events. Additionally, when the sample size is 3600 and the censoring rate is 0.5, there is an improvement of 0.39 for OPERA without pruning and 0.6 for the lasso tree, while for a censoring rate of 0.8, there is an improvement of 0.16 for OPERA without pruning and 0.46 for the lasso tree. The overall accuracy increases as the sample size increases. Coarse pruning with LRT ($\alpha = 0.01$) also performs well across different simulation scenarios with no more than a 0.03 increase in the median edge misclassification compared to fine pruning with LRT ($\alpha = 0.01$). Among the other pruning criteria, using LRT ($\alpha = 0.05$) yields the best results, followed by AIC and then BS. However, the discrepancy is less obvious as the censoring proportion increases. IBS proves to be a more effective criterion for pruning with survival data compared to BS used for binary data. OPERA outperforms the lasso tree in all four scenarios, and pruning consistently improves performance by mitigating the over-partitioning issue, regardless of which stopping rule is used.

In three-risk-factor scenarios favoring the OPERA tree, as depicted in Figure 3.4b, similar conclusions

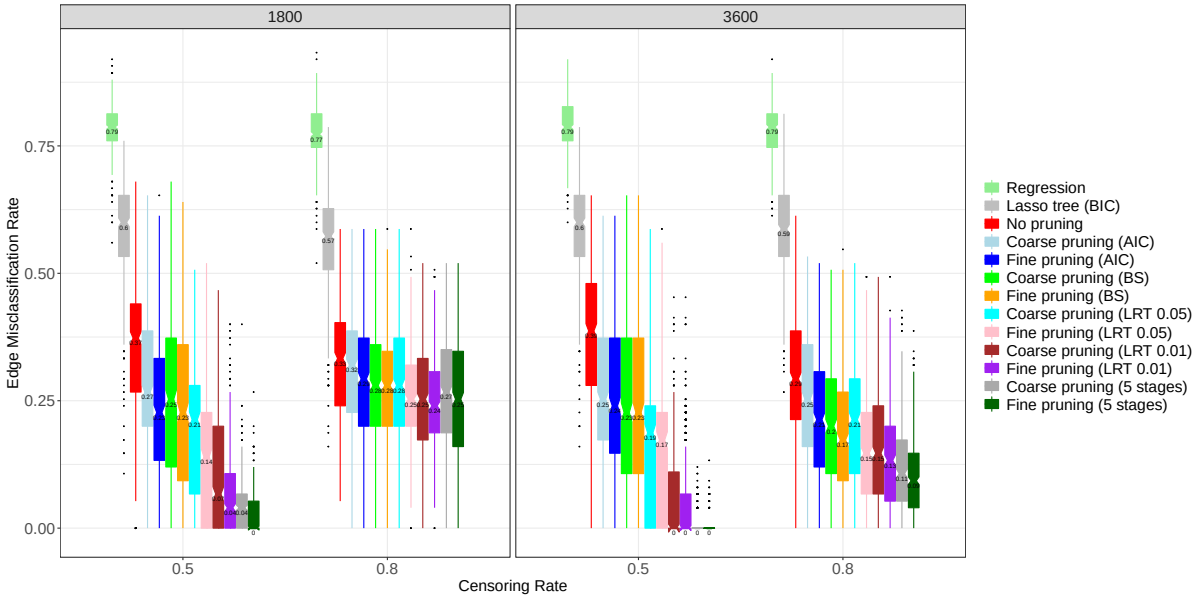


(a) Simulation scenario favoring lasso tree

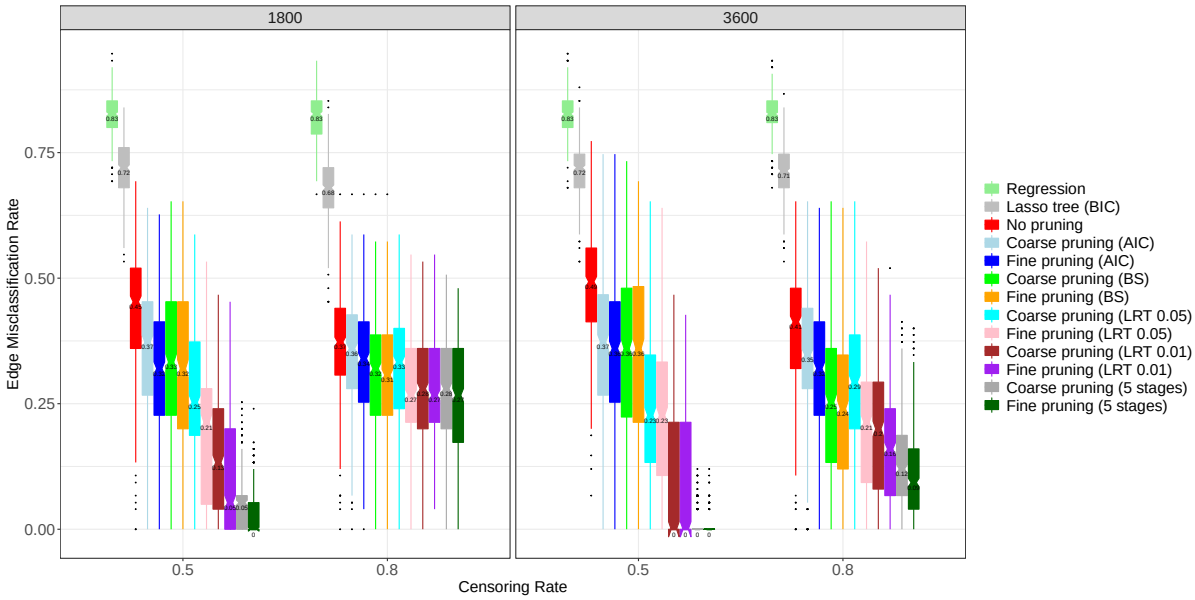


(b) Simulation scenario favoring opera

Figure 3.3: The edge misclassification rate for different methods with two risk factors



(a) Simulation scenario favoring lasso tree



(b) Simulation scenario favoring opera

Figure 3.4: The edge misclassification rate for different methods with three risk factors

can be drawn. The regression method still performs the worst, while pruning using LRT ($\alpha = 0.01$) still demonstrates comparable performance to pruning based on the predefined number of stages across all four scenarios. When the sample size is 1800 and the censoring proportion is 0.5, fine pruning using LRT ($\alpha = 0.01$) results in an improvement of 0.4 in reducing the median edge misclassification rate compared to OPERA without pruning, and an improvement of 0.67 compared to the lasso tree. For a censoring rate of 0.8, there is an improvement of 0.1 compared to OPERA without pruning and 0.41 compared to the lasso tree. Similarly, when the sample size is 3600 and the censoring proportion is 0.5, there is an improvement of 0.49 using OPERA without pruning and 0.72 using the lasso tree, while for a censoring rate of 0.8, there is an improvement of 0.25 compared to OPERA without pruning and 0.55 compared to the lasso tree. Coarse pruning with LRT ($\alpha = 0.01$) also performs well across different simulation scenarios with no more than a 0.08 increase in the median edge misclassification compared to fine pruning with LRT ($\alpha = 0.01$). Before pruning using LRT ($\alpha = 0.01$) is applied, the overall misclassification rates are higher compared to three-risk-factor scenarios favoring the lasso tree, and the lasso tree performs worse than in the three-risk-factor scenarios favoring the lasso tree, as three-risk-factor scenarios favoring OPERA are more complicated and involve non-neighboring patterns. A larger sample size and a smaller censoring rate still correspond to higher accuracy.

In two-risk factor scenarios favoring the lasso tree, as depicted in Figure 3.3a, the lasso tree and OPERA demonstrate very close performance, with the difference in the median edge misclassification rate being no more than 0.04. Pruning using LRT ($\alpha = 0.01$) still leads to the best performance among various stopping rules when the true number of stages is unknown, except when the sample size equals 800 and the censoring rate equals 0.8. However, in that specific scenario, the median edge misclassification rate is only 0.04 higher than the lowest, which corresponds to roughly 3 edges out of 75. Coarse pruning with LRT ($\alpha = 0.01$) demonstrates the same median edge misclassification rate as fine pruning with LRT ($\alpha = 0.01$) across all simulation scenarios with a relatively smaller interquartile range (IQR). In two-risk factor scenarios favoring OPERA, as shown in Figure 3.3b, OPERA without pruning outperforms the lasso tree in the median edge misclassification rate in all scenarios, as the lasso tree fails to identify non-neighboring staging patterns. Coarse pruning with LRT ($\alpha = 0.01$) still demonstrates the best performance across all simulation scenarios.

Across all scenarios, pruning using LRT ($\alpha = 0.01$) consistently performs the best, comparable to pruning using the true number of stages. Coarse pruning can achieve as good performance as fine pruning with no more than a 0.08 increase in the median edge misclassification when LRT ($\alpha = 0.01$) is used as the stopping rule.

3.3.3 Discussion

Different initial steps, including utilizing the lasso tree with AIC as the criterion for determining optimal groupings, the lasso tree with BIC as the default criterion, and OPERA without pruning, are still considered to explore how initial methods can affect the performance. As depicted in Figures 3.5, 3.6, 3.7, and 3.8, while the three initial methods yield different results without pruning, the introduction of pruning enables all methods to perform closely well, with comparable median edge misclassification rates.

To further explore which pruning method performs the best across different initial methods, we calculate the mean edge misclassification for each simulation scenario across 500 simulations and select the top 3 pruning methods with the lowest mean edge misclassification rates. The results, including the estimated standard deviation (SD) and 95% confidence intervals, are displayed in Table 3.2 - 3.5. In three-risk-factor simulation scenarios, pruning using LRT ($\alpha = 0.01$) consistently performs the best in terms of the mean edge misclassification rate. In two-risk-factor simulation scenarios, pruning using LRT ($\alpha = 0.01$) consistently demonstrates the top three lowest mean edge misclassification rates, except when the sample size is 800 and the censoring rate is 0.8. However, in that specific simulation scenario, we observe very close performance between coarse pruning using LRT ($\alpha = 0.01$) and the best approach, with a difference no greater than 0.04 in the median misclassification rate.

To investigate whether coarse pruning using LRT ($\alpha = 0.01$) can demonstrate performance on par with fine pruning using LRT ($\alpha = 0.01$), we conduct the paired sample t-test between coarse pruning and fine pruning, as shown in Table 3.6. The largest difference is 5.5%, indicating that roughly 4 more edges out of 75 edges are correctly classified. Thus, coarse pruning using LRT ($\alpha = 0.01$) can perform as well as fine pruning using LRT ($\alpha = 0.01$) in the majority of simulation scenarios, with only a slight increase in edge misclassification rate in a few cases.

To evaluate the performance of fine pruning methods with exhaustive search compared to those with a quadratic programming constraint, each fine pruning method using each criterion is examined, as illustrated in Figures 3.9 - 3.12. Across all simulation scenarios, utilizing a quadratic programming constraint achieves a similarly low median edge misclassification rate as that of exhaustive search, with 0.04 as the largest difference in the median edge misclassification rate. In terms of the mean edge misclassification rate, the largest difference, as shown in Table 3.6, between the two approaches is 1.9%, indicating that roughly 1.5 more edges out of 75 edges are correctly classified. Thus, across all scenarios, a quadratic constraint can deliver performance on par with exhaustive search.

All three initial approaches perform closely well after pruning, with 3.5% as the largest difference in

the mean edge misclassification rate. In real data analysis, we recommend using OPERA as the initial method, not only due to its good performance but also its lower computational cost as shown in Table 3.7, in comparison with the lasso tree.

Table 3.2: The mean edge misclassification rates for the top 3 pruning methods with different initial methods in three-risk-factor simulation scenarios with non-neighboring staging patterns (Mean = Estimated Mean; SD = Estimated Standard Deviation; Lower = 95% Confidence Interval Lower Limit; Upper = 95% Confidence Interval Upper Limit; quad = using quadratic constraint; ex = using exhaustive search)

Sample Size	Censoring Rate	Initial Method	Pruning Method	Mean	SD	Lower	Upper
1800	0.5	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	0.098	0.117	0.088	0.108
1800	0.5	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	0.110	0.121	0.099	0.120
1800	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.134	0.130	0.123	0.145
1800	0.5	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	0.101	0.116	0.091	0.111
1800	0.5	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	0.112	0.121	0.102	0.123
1800	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.136	0.130	0.124	0.147
1800	0.5	OPERA	fine pruning ex (LRT 0.01)	0.094	0.111	0.084	0.103
1800	0.5	OPERA	fine pruning quad (LRT 0.01)	0.113	0.120	0.103	0.124
1800	0.5	OPERA	coarse pruning (LRT 0.01)	0.148	0.126	0.137	0.159
1800	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.250	0.110	0.240	0.259
1800	0.8	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	0.259	0.108	0.250	0.269
1800	0.8	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	0.263	0.117	0.252	0.273
1800	0.8	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	0.297	0.115	0.287	0.307
1800	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.299	0.119	0.289	0.309
1800	0.8	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	0.306	0.118	0.295	0.316
1800	0.8	OPERA	coarse pruning (LRT 0.01)	0.274	0.102	0.265	0.283
1800	0.8	OPERA	fine pruning ex (LRT 0.01)	0.280	0.103	0.271	0.289
1800	0.8	OPERA	fine pruning ex (LRT 0.05)	0.280	0.103	0.271	0.289
3600	0.5	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	0.106	0.122	0.095	0.117
3600	0.5	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	0.108	0.123	0.097	0.119
3600	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.113	0.124	0.103	0.124
3600	0.5	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	0.106	0.123	0.096	0.117
3600	0.5	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	0.107	0.124	0.096	0.118
3600	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.110	0.122	0.099	0.121
3600	0.5	OPERA	fine pruning ex (LRT 0.01)	0.090	0.117	0.079	0.100

Table 3.2: The mean edge misclassification rates for the top 3 pruning methods with different initial methods in three-risk-factor simulation scenarios with non-neighboring staging patterns (Mean = Estimated Mean; SD = Estimated Standard Deviation; Lower = 95% Confidence Interval Lower Limit; Upper = 95% Confidence Interval Upper Limit; quad = using quadratic constraint; ex = using exhaustive search)

Sample Size	Censoring Rate	Initial Method	Pruning Method	Mean	SD	Lower	Upper
3600	0.5	OPERA	fine pruning quad (LRT 0.01)	0.090	0.117	0.080	0.101
3600	0.5	OPERA	coarse pruning (LRT 0.01)	0.102	0.123	0.091	0.112
3600	0.8	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	0.159	0.111	0.150	0.169
3600	0.8	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	0.174	0.124	0.163	0.185
3600	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.186	0.134	0.174	0.197
3600	0.8	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	0.162	0.113	0.152	0.172
3600	0.8	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	0.177	0.126	0.166	0.188
3600	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.193	0.131	0.182	0.205
3600	0.8	OPERA	fine pruning ex (LRT 0.01)	0.160	0.107	0.150	0.169
3600	0.8	OPERA	fine pruning quad (LRT 0.01)	0.177	0.119	0.167	0.188
3600	0.8	OPERA	coarse pruning (LRT 0.01)	0.194	0.128	0.183	0.205

Table 3.3: The mean edge misclassification rates for the top 3 pruning methods with different initial methods in three-risk-factor simulation scenarios with neighboring staging patterns

Sample Size	Censoring Rate	Initial Method	Pruning Method	Mean	SD	Lower	Upper
1800	0.5	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	0.080	0.103	0.071	0.089
1800	0.5	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	0.087	0.105	0.078	0.096
1800	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.100	0.112	0.090	0.110
1800	0.5	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	0.093	0.104	0.084	0.102
1800	0.5	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	0.097	0.104	0.088	0.106
1800	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.111	0.112	0.101	0.120
1800	0.5	OPERA	fine pruning ex (LRT 0.01)	0.070	0.089	0.063	0.078
1800	0.5	OPERA	fine pruning quad (LRT 0.01)	0.080	0.097	0.072	0.089
1800	0.5	OPERA	coarse pruning (LRT 0.01)	0.103	0.110	0.094	0.113
1800	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.210	0.111	0.200	0.220
1800	0.8	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	0.232	0.106	0.223	0.241
1800	0.8	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	0.233	0.114	0.223	0.243
1800	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.242	0.117	0.232	0.252
1800	0.8	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	0.263	0.109	0.254	0.273
1800	0.8	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	0.267	0.119	0.256	0.277
1800	0.8	OPERA	fine pruning ex (LRT 0.01)	0.248	0.099	0.239	0.257
1800	0.8	OPERA	coarse pruning (LRT 0.01)	0.255	0.104	0.246	0.264
1800	0.8	OPERA	fine pruning ex (LRT 0.05)	0.257	0.101	0.248	0.266
3600	0.5	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	0.072	0.106	0.063	0.082
3600	0.5	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	0.072	0.105	0.063	0.081
3600	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.076	0.107	0.066	0.085
3600	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.071	0.104	0.062	0.080
3600	0.5	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	0.071	0.104	0.061	0.080
3600	0.5	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	0.072	0.105	0.063	0.081
3600	0.5	OPERA	fine pruning ex (LRT 0.01)	0.054	0.093	0.046	0.063
3600	0.5	OPERA	fine pruning quad (LRT 0.01)	0.055	0.093	0.047	0.064
3600	0.5	OPERA	coarse pruning (LRT 0.01)	0.060	0.097	0.052	0.069

Table 3.3: The mean edge misclassification rates for the top 3 pruning methods with different initial methods in three-risk-factor simulation scenarios with neighboring staging patterns

Sample Size	Censoring Rate	Initial Method	Pruning Method	Mean	SD	Lower	Upper
3600	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.134	0.109	0.124	0.143
3600	0.8	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	0.136	0.099	0.127	0.145
3600	0.8	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	0.136	0.106	0.126	0.145
3600	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.136	0.108	0.127	0.146
3600	0.8	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	0.142	0.105	0.132	0.151
3600	0.8	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	0.143	0.100	0.135	0.152
3600	0.8	OPERA	fine pruning ex (LRT 0.01)	0.135	0.094	0.127	0.143
3600	0.8	OPERA	fine pruning quad (LRT 0.01)	0.140	0.101	0.131	0.149
3600	0.8	OPERA	coarse pruning (LRT 0.01)	0.155	0.106	0.146	0.165

Table 3.4: The mean edge misclassification rates for the top 3 pruning methods with different initial methods in two-risk-factor simulation scenarios with non-neighboring staging patterns

Sample Size	Censoring Rate	Initial Method	Pruning Method	Mean	SD	Lower	Upper
1600	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.019	0.060	0.014	0.024
1600	0.5	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	0.025	0.071	0.019	0.031
1600	0.5	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	0.026	0.073	0.019	0.032
1600	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.018	0.059	0.013	0.024
1600	0.5	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	0.024	0.071	0.018	0.030
1600	0.5	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	0.025	0.073	0.019	0.031
1600	0.5	OPERA	coarse pruning (LRT 0.01)	0.020	0.060	0.015	0.025
1600	0.5	OPERA	fine pruning ex (LRT 0.01)	0.026	0.074	0.020	0.033
1600	0.5	OPERA	fine pruning quad (LRT 0.01)	0.026	0.072	0.020	0.032
1600	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.05)	0.146	0.105	0.137	0.155
1600	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.153	0.094	0.145	0.162
1600	0.8	Lasso tree (AIC)	fine pruning ex (LRT 0.05)	0.162	0.119	0.152	0.173
1600	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.167	0.090	0.159	0.174
1600	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.05)	0.168	0.100	0.160	0.177
1600	0.8	Lasso tree (BIC)	fine pruning ex (LRT 0.05)	0.176	0.115	0.166	0.186
1600	0.8	OPERA	coarse pruning (LRT 0.01)	0.157	0.096	0.149	0.165
1600	0.8	OPERA	coarse pruning (LRT 0.05)	0.159	0.107	0.149	0.168
1600	0.8	OPERA	fine pruning ex (LRT 0.05)	0.178	0.120	0.167	0.188
800	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.051	0.081	0.043	0.058
800	0.5	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	0.084	0.123	0.073	0.095
800	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.05)	0.085	0.114	0.075	0.095
800	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.058	0.086	0.050	0.065
800	0.5	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	0.089	0.123	0.078	0.099
800	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.05)	0.090	0.113	0.080	0.100
800	0.5	OPERA	coarse pruning (LRT 0.01)	0.051	0.082	0.044	0.058
800	0.5	OPERA	fine pruning ex (LRT 0.01)	0.085	0.122	0.075	0.096
800	0.5	OPERA	coarse pruning (LRT 0.05)	0.088	0.116	0.078	0.098

Table 3.4: The mean edge misclassification rates for the top 3 pruning methods with different initial methods in two-risk-factor simulation scenarios with non-neighboring staging patterns

Sample Size	Censoring Rate	Initial Method	Pruning Method	Mean	SD	Lower	Upper
800	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.05)	0.227	0.107	0.218	0.237
800	0.8	Lasso tree (AIC)	fine pruning ex (AIC)	0.235	0.124	0.224	0.246
800	0.8	Lasso tree (AIC)	fine pruning quad (AIC)	0.237	0.125	0.226	0.248
800	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.05)	0.247	0.107	0.237	0.256
800	0.8	Lasso tree (BIC)	fine pruning ex (AIC)	0.255	0.119	0.245	0.266
800	0.8	Lasso tree (BIC)	fine pruning quad (AIC)	0.257	0.119	0.247	0.268
800	0.8	OPERA	coarse pruning (LRT 0.05)	0.244	0.105	0.235	0.253
800	0.8	OPERA	fine pruning ex (AIC)	0.260	0.113	0.250	0.269
800	0.8	OPERA	fine pruning quad (AIC)	0.261	0.115	0.251	0.271

Table 3.5: The mean edge misclassification rates for the top 3 pruning methods with different initial methods in two-risk-factor simulation scenarios with neighboring staging patterns

Sample Size	Censoring Rate	Initial Method	Pruning Method	Mean	SD	Lower	Upper
1600	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.016	0.055	0.012	0.021
1600	0.5	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	0.024	0.074	0.018	0.030
1600	0.5	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	0.026	0.075	0.020	0.033
1600	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.016	0.054	0.011	0.021
1600	0.5	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	0.023	0.074	0.017	0.030
1600	0.5	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	0.026	0.075	0.019	0.032
1600	0.5	OPERA	coarse pruning (LRT 0.01)	0.016	0.055	0.012	0.021
1600	0.5	OPERA	fine pruning quad (LRT 0.01)	0.024	0.074	0.017	0.030
1600	0.5	OPERA	fine pruning ex (LRT 0.01)	0.026	0.075	0.020	0.033
1600	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.05)	0.122	0.104	0.113	0.131
1600	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.138	0.099	0.129	0.146
1600	0.8	Lasso tree (AIC)	fine pruning quad (LRT 0.05)	0.146	0.122	0.136	0.157
1600	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.05)	0.136	0.101	0.127	0.145
1600	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.143	0.095	0.135	0.152
1600	0.8	Lasso tree (BIC)	fine pruning ex (AIC)	0.161	0.112	0.152	0.171
1600	0.8	OPERA	coarse pruning (LRT 0.01)	0.140	0.101	0.131	0.149
1600	0.8	OPERA	coarse pruning (LRT 0.05)	0.141	0.107	0.132	0.150
1600	0.8	OPERA	fine pruning quad (LRT 0.05)	0.173	0.123	0.162	0.183
800	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	0.036	0.068	0.031	0.042
800	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.05)	0.065	0.100	0.056	0.074
800	0.5	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	0.066	0.115	0.056	0.076
800	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	0.042	0.073	0.035	0.048
800	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.05)	0.067	0.099	0.058	0.075
800	0.5	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	0.070	0.117	0.060	0.081
800	0.5	OPERA	coarse pruning (LRT 0.01)	0.037	0.069	0.031	0.043
800	0.5	OPERA	coarse pruning (LRT 0.05)	0.065	0.099	0.057	0.074
800	0.5	OPERA	fine pruning quad (LRT 0.01)	0.067	0.115	0.057	0.077

Table 3.5: The mean edge misclassification rates for the top 3 pruning methods with different initial methods in two-risk-factor simulation scenarios with neighboring staging patterns

Sample Size	Censoring Rate	Initial Method	Pruning Method	Mean	SD	Lower	Upper
800	0.8	Lasso tree (AIC)	fine pruning quad (AIC)	0.212	0.121	0.202	0.223
800	0.8	Lasso tree (AIC)	fine pruning ex (AIC)	0.213	0.120	0.202	0.223
800	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.05)	0.214	0.110	0.204	0.223
800	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.05)	0.218	0.107	0.209	0.228
800	0.8	Lasso tree (BIC)	coarse pruning (AIC)	0.219	0.114	0.209	0.229
800	0.8	Lasso tree (BIC)	fine pruning ex (AIC)	0.219	0.109	0.210	0.229
800	0.8	Lasso tree (BIC)	fine pruning quad (AIC)	0.220	0.110	0.210	0.229
800	0.8	OPERA	coarse pruning (LRT 0.05)	0.247	0.107	0.237	0.256
800	0.8	OPERA	fine pruning quad (AIC)	0.253	0.112	0.244	0.263
800	0.8	OPERA	fine pruning ex (AIC)	0.254	0.112	0.244	0.264

Table 3.6: The pairwise comparisons in the mean misclassification rate among coarse pruning, fine pruning using exhaustive search and fine pruning using quadratic constraint with LRT ($\alpha = 0.01$) (Each cell displays the difference with the corresponding p-value in parentheses)

	Sample Size	Censor -ing Rate	Initial Method	Coarse vs Ex	Coarse vs Quad	Ex vs Quad
Three Risk Factors & Non-neigh -boring	1800	0.5	OPERA	**0.055** (< 0.001)	0.035 (< 0.001)	**-0.019** (< 0.001)
	1800	0.5	Lasso tree (AIC)	0.036 (< 0.001)	0.024 (< 0.001)	-0.012 (< 0.001)
	1800	0.5	Lasso tree (BIC)	0.035 (< 0.001)	0.023 (< 0.001)	-0.011 (< 0.001)
	1800	0.8	OPERA	-0.006 (0.246)	-0.014 (0.002)	-0.009 (0.035)
	1800	0.8	Lasso tree (AIC)	-0.010 (0.047)	-0.013 (0.005)	-0.003 (0.450)
	1800	0.8	Lasso tree (BIC)	-0.007 (0.171)	0.002 (0.647)	0.009 (0.031)
	3600	0.5	OPERA	0.012 (< 0.001)	0.011 (< 0.001)	-0.001 (0.498)
	3600	0.5	Lasso tree (AIC)	0.006 (0.015)	0.007 (< 0.001)	0.002 (0.329)
	3600	0.5	Lasso tree (BIC)	0.003 (0.204)	0.004 (0.057)	0.001 (0.633)
	3600	0.8	OPERA	0.034 (< 0.001)	0.017 (< 0.001)	-0.018 (< 0.001)
	3600	0.8	Lasso tree (AIC)	0.026 (< 0.001)	0.012 (0.009)	-0.014 (< 0.001)
	3600	0.8	Lasso tree (BIC)	0.031 (< 0.001)	0.016 (0.001)	-0.015 (< 0.001)
Three Risk Factors & Neighbor -ing	1800	0.5	OPERA	0.033 (< 0.001)	0.023 (< 0.001)	-0.010 (0.001)
	1800	0.5	Lasso tree (AIC)	0.020 (< 0.001)	0.013 (< 0.001)	-0.007 (0.001)
	1800	0.5	Lasso tree (BIC)	0.018 (< 0.001)	0.014 (< 0.001)	-0.004 (0.071)
	1800	0.8	OPERA	0.007 (0.068)	-0.007 (0.122)	-0.014 (< 0.001)
	1800	0.8	Lasso tree (AIC)	-0.022 (< 0.001)	-0.023 (< 0.001)	-0.001 (0.702)
	1800	0.8	Lasso tree (BIC)	-0.021 (< 0.001)	-0.025 (< 0.001)	-0.004 (0.293)
	3600	0.5	OPERA	0.006 (< 0.001)	0.005 (0.002)	-0.001 (0.100)
	3600	0.5	Lasso tree (AIC)	0.003 (0.048)	0.004 (0.018)	0.000 (0.559)
	3600	0.5	Lasso tree (BIC)	-0.001 (0.627)	0.000 (0.922)	0.001 (0.182)
	3600	0.8	OPERA	0.020 (< 0.001)	0.015 (< 0.001)	-0.005 (0.111)
	3600	0.8	Lasso tree (AIC)	-0.002 (0.537)	-0.002 (0.538)	0.000 (0.928)
	3600	0.8	Lasso tree (BIC)	-0.007 (0.091)	-0.005 (0.166)	0.002 (0.473)

Table 3.6: The pairwise comparisons in the mean misclassification rate among coarse pruning, fine pruning using exhaustive search and fine pruning using quadratic constraint with LRT ($\alpha = 0.01$) (Each cell displays the difference with the corresponding p-value in parentheses)

	Sample Size	Censor -ing Rate	Initial Method	Coarse vs Ex	Coarse vs Quad	Ex vs Quad
Two Risk Factors & Non-neighbor -ing	800	0.5	OPERA	-0.034 (< 0.001)	-0.037 (< 0.001)	-0.002 (0.405)
	800	0.5	Lasso tree (AIC)	-0.034 (< 0.001)	-0.036 (< 0.001)	-0.002 (0.407)
	800	0.5	Lasso tree (BIC)	-0.031 (< 0.001)	-0.033 (< 0.001)	-0.002 (0.361)
	800	0.8	OPERA	-0.062 (< 0.001)	-0.065 (< 0.001)	-0.003 (0.383)
	800	0.8	Lasso tree (AIC)	-0.065 (< 0.001)	-0.065 (< 0.001)	0.000 (0.904)
	800	0.8	Lasso tree (BIC)	-0.060 (< 0.001)	-0.059 (< 0.001)	0.001 (0.768)
	1600	0.5	OPERA	-0.007 (0.003)	-0.006 (0.002)	0.001 (0.631)
	1600	0.5	Lasso tree (AIC)	-0.007 (0.003)	-0.006 (0.002)	0.001 (0.631)
	1600	0.5	Lasso tree (BIC)	-0.007 (0.002)	-0.006 (0.002)	0.001 (0.631)
	1600	0.8	OPERA	-0.039 (< 0.001)	-0.042 (< 0.001)	-0.003 (0.157)
	1600	0.8	Lasso tree (AIC)	-0.037 (< 0.001)	-0.040 (< 0.001)	-0.003 (0.175)
	1600	0.8	Lasso tree (BIC)	-0.029 (< 0.001)	-0.032 (< 0.001)	-0.003 (0.183)
Two Risk Factors & Neighbor -ing	800	0.5	OPERA	-0.035 (< 0.001)	-0.030 (< 0.001)	0.005 (0.087)
	800	0.5	Lasso tree (AIC)	-0.033 (< 0.001)	-0.030 (< 0.001)	0.003 (0.252)
	800	0.5	Lasso tree (BIC)	-0.032 (< 0.001)	-0.029 (< 0.001)	0.004 (0.171)
	800	0.8	OPERA	-0.030 (< 0.001)	-0.030 (< 0.001)	0.001 (0.798)
	800	0.8	Lasso tree (AIC)	-0.040 (< 0.001)	-0.036 (< 0.001)	0.004 (0.089)
	800	0.8	Lasso tree (BIC)	-0.039 (< 0.001)	-0.036 (< 0.001)	0.003 (0.140)
	1600	0.5	OPERA	-0.010 (< 0.001)	-0.008 (0.004)	0.002 (0.097)
	1600	0.5	Lasso tree (AIC)	-0.010 (< 0.001)	-0.008 (0.004)	0.002 (0.097)
	1600	0.5	Lasso tree (BIC)	-0.010 (< 0.001)	-0.007 (0.004)	0.002 (0.097)
	1600	0.8	OPERA	-0.049 (< 0.001)	-0.044 (< 0.001)	0.005 (0.010)
	1600	0.8	Lasso tree (AIC)	-0.044 (< 0.001)	-0.038 (< 0.001)	0.006 (0.002)
	1600	0.8	Lasso tree (BIC)	-0.046 (< 0.001)	-0.041 (< 0.001)	0.004 (0.038)

Table 3.7: The average time for each initial method along with coarse pruning with LRT(0.01)

	Sample Size	Censoring Rate	Initial Method	Pruning Method	Average (s)	Total (s)
	1800	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	312	
	1800	0.5	Lasso tree (AIC)	no pruning	8407	8720
	1800	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	321	
Three	1800	0.5	Lasso tree (BIC)	no pruning	10004	10326
Risk	1800	0.5	OPERA	coarse pruning (LRT 0.01)	91	
Factors &	1800	0.5	OPERA	no pruning	237	328
Non-	1800	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	294	
neighbor	1800	0.8	Lasso tree (AIC)	no pruning	3256	3550
-ing	1800	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	197	
	1800	0.8	Lasso tree (BIC)	no pruning	2964	3161
	1800	0.8	OPERA	coarse pruning (LRT 0.01)	87	
	1800	0.8	OPERA	no pruning	92	178
	3600	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	596	
	3600	0.5	Lasso tree (AIC)	no pruning	37046	37641
	3600	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	571	
	3600	0.5	Lasso tree (BIC)	no pruning	55601	56171
	3600	0.5	OPERA	coarse pruning (LRT 0.01)	332	
	3600	0.5	OPERA	no pruning	842	1174
	3600	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	581	
	3600	0.8	Lasso tree (AIC)	no pruning	16034	16615
	3600	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	822	
	3600	0.8	Lasso tree (BIC)	no pruning	35263	36085
	3600	0.8	OPERA	coarse pruning (LRT 0.01)	291	
	3600	0.8	OPERA	no pruning	531	822

Table 3.7: The average time for each initial method along with coarse pruning with LRT(0.01)

	Sample Size	Censoring Rate	Initial Method	Pruning Method	Average (s)	Total (s)
Three Risk Factors & Neighbor -ing	1800	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	222	
	1800	0.5	Lasso tree (AIC)	no pruning	7577	7799
	1800	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	185	
	1800	0.5	Lasso tree (BIC)	no pruning	7849	8034
	1800	0.5	OPERA	coarse pruning (LRT 0.01)	100	
	1800	0.5	OPERA	no pruning	144	244
	1800	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	237	
	1800	0.8	Lasso tree (AIC)	no pruning	3704	3941
	1800	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	159	
	1800	0.8	Lasso tree (BIC)	no pruning	3646	3805
	1800	0.8	OPERA	coarse pruning (LRT 0.01)	69	
	1800	0.8	OPERA	no pruning	96	165
	3600	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	398	
	3600	0.5	Lasso tree (AIC)	no pruning	36461	36858
	3600	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	364	
	3600	0.5	Lasso tree (BIC)	no pruning	52995	53359
	3600	0.5	OPERA	coarse pruning (LRT 0.01)	355	
	3600	0.5	OPERA	no pruning	781	1136
	3600	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	449	
	3600	0.8	Lasso tree (AIC)	no pruning	18592	19041
	3600	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	530	
	3600	0.8	Lasso tree (BIC)	no pruning	35340	35870
	3600	0.8	OPERA	coarse pruning (LRT 0.01)	237	
	3600	0.8	OPERA	no pruning	508	745

Table 3.7: The average time for each initial method along with coarse pruning with LRT(0.01)

	Sample Size	Censoring Rate	Initial Method	Pruning Method	Average (s)	Total (s)
Two Risk Factors & Non-neighbor-ing	1600	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	99	
	1600	0.5	Lasso tree (AIC)	no pruning	4410	4509
	1600	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	97	
	1600	0.5	Lasso tree (BIC)	no pruning	4652	4749
	1600	0.5	OPERA	coarse pruning (LRT 0.01)	230	
	1600	0.5	OPERA	no pruning	93	324
	1600	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	94	
	1600	0.8	Lasso tree (AIC)	no pruning	1122	1215
	1600	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	75	
	1600	0.8	Lasso tree (BIC)	no pruning	1123	1198
	1600	0.8	OPERA	coarse pruning (LRT 0.01)	64	
	1600	0.8	OPERA	no pruning	88	153
	800	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	32	
	800	0.5	Lasso tree (AIC)	no pruning	135	167
	800	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	28	
	800	0.5	Lasso tree (BIC)	no pruning	140	169
	800	0.5	OPERA	coarse pruning (LRT 0.01)	24	
	800	0.5	OPERA	no pruning	13	37
	800	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	30	
	800	0.8	Lasso tree (AIC)	no pruning	52	82
	800	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	22	
	800	0.8	Lasso tree (BIC)	no pruning	67	90
	800	0.8	OPERA	coarse pruning (LRT 0.01)	18	
	800	0.8	OPERA	no pruning	22	40

Table 3.7: The average time for each initial method along with coarse pruning with LRT(0.01)

	Sample Size	Censoring Rate	Initial Method	Pruning Method	Average (s)	Total (s)
	1600	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	81	
	1600	0.5	Lasso tree (AIC)	no pruning	4259	4340
	1600	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	71	
	1600	0.5	Lasso tree (BIC)	no pruning	4422	4492
Two Risk	1600	0.5	OPERA	coarse pruning (LRT 0.01)	167	
Factors &	1600	0.5	OPERA	no pruning	74	242
Neighbor	1600	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	83	
-ing	1600	0.8	Lasso tree (AIC)	no pruning	1415	1498
	1600	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	70	
	1600	0.8	Lasso tree (BIC)	no pruning	1495	1565
	1600	0.8	OPERA	coarse pruning (LRT 0.01)	139	
	1600	0.8	OPERA	no pruning	412	550
	800	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	24	
	800	0.5	Lasso tree (AIC)	no pruning	129	153
	800	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	21	
	800	0.5	Lasso tree (BIC)	no pruning	128	149
	800	0.5	OPERA	coarse pruning (LRT 0.01)	21	
	800	0.5	OPERA	no pruning	195	215
	800	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	23	
	800	0.8	Lasso tree (AIC)	no pruning	51	74
	800	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	75	
	800	0.8	Lasso tree (BIC)	no pruning	74	149
	800	0.8	OPERA	coarse pruning (LRT 0.01)	14	
	800	0.8	OPERA	no pruning	6	20

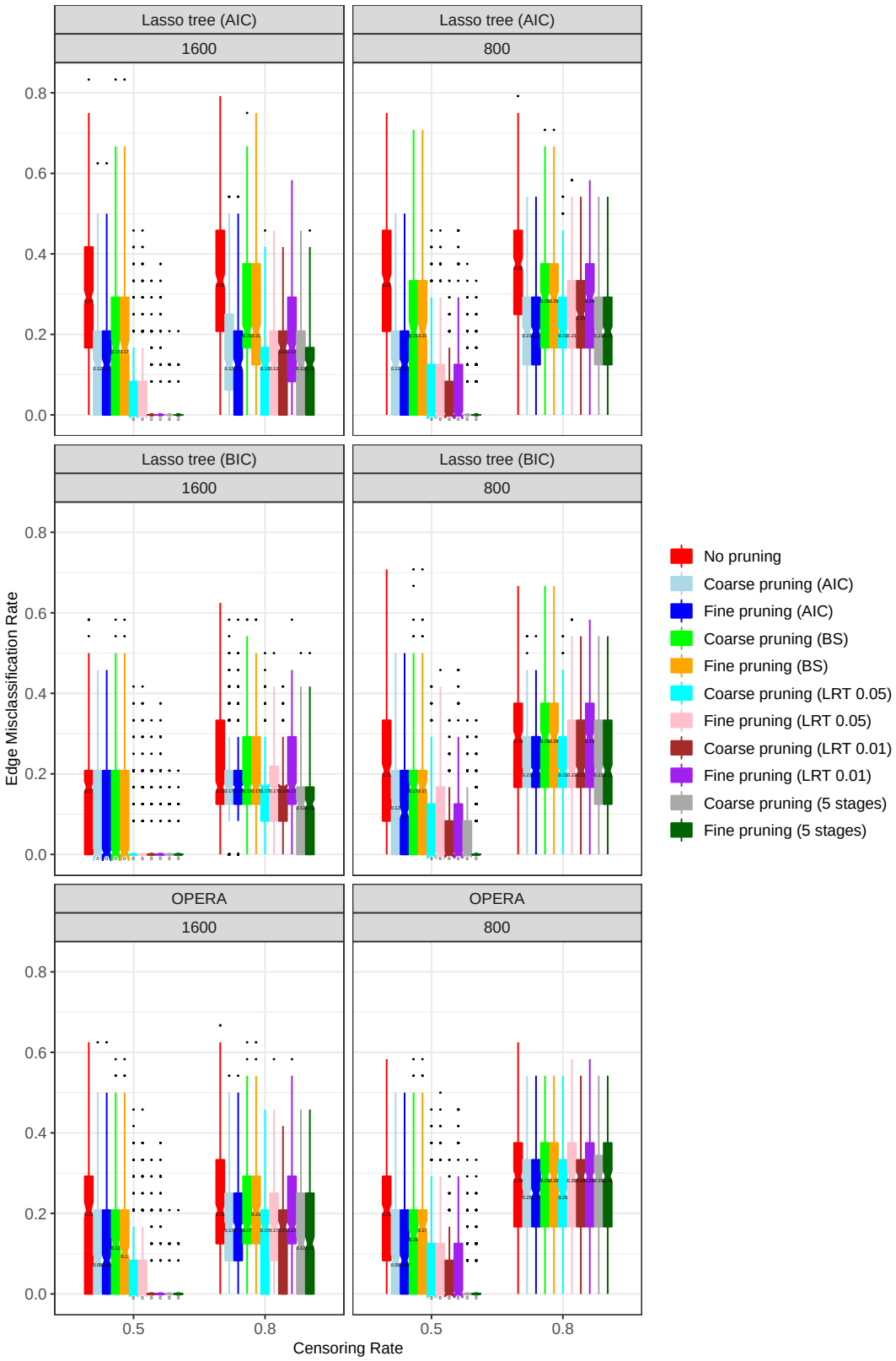


Figure 3.5: The edge misclassification rate for different pruning methods with two-risk-factor simulation scenarios favoring lasso tree

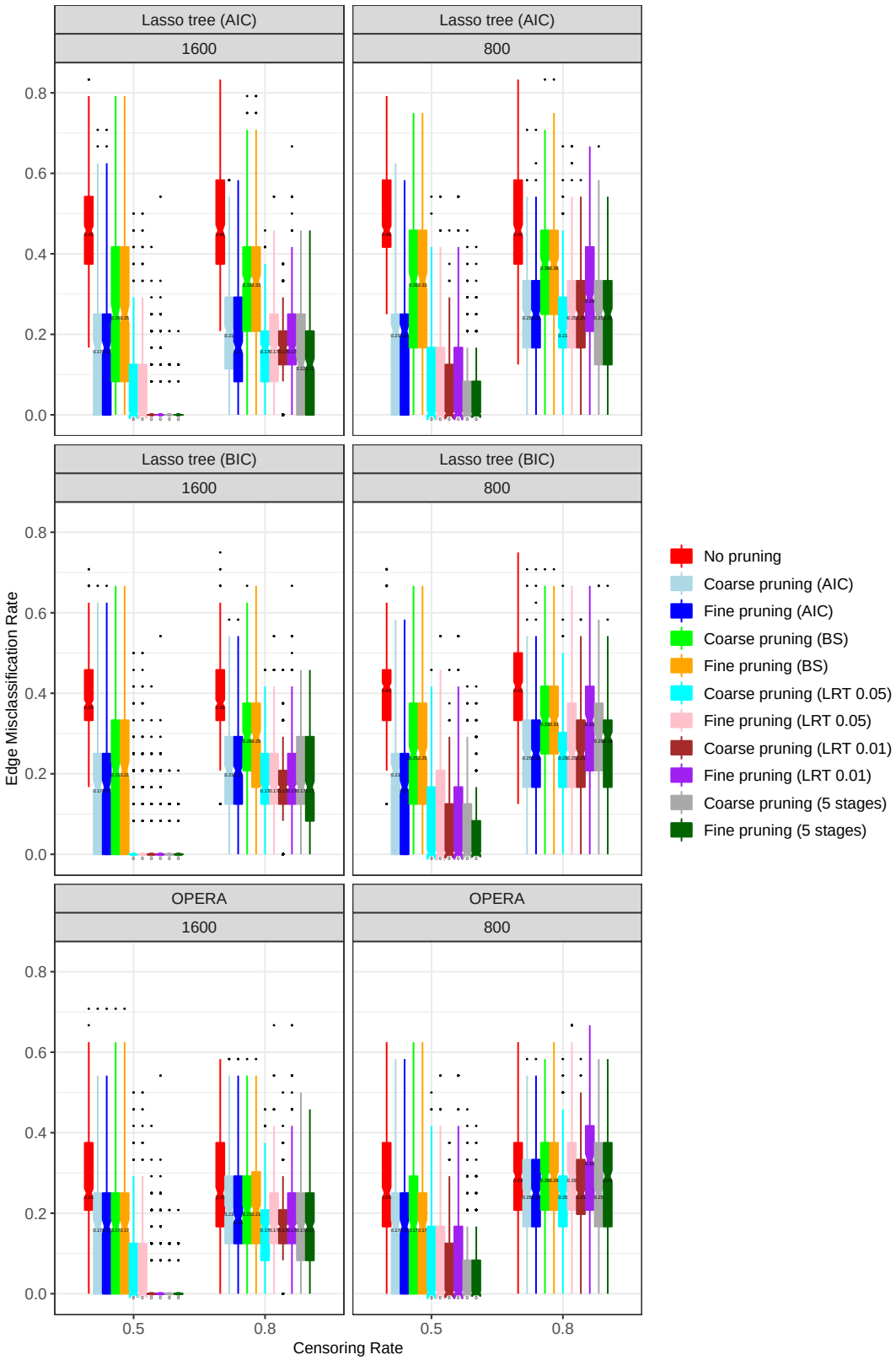


Figure 3.6: The edge misclassification rate for different pruning methods with two-risk-factor simulation scenarios favoring opera

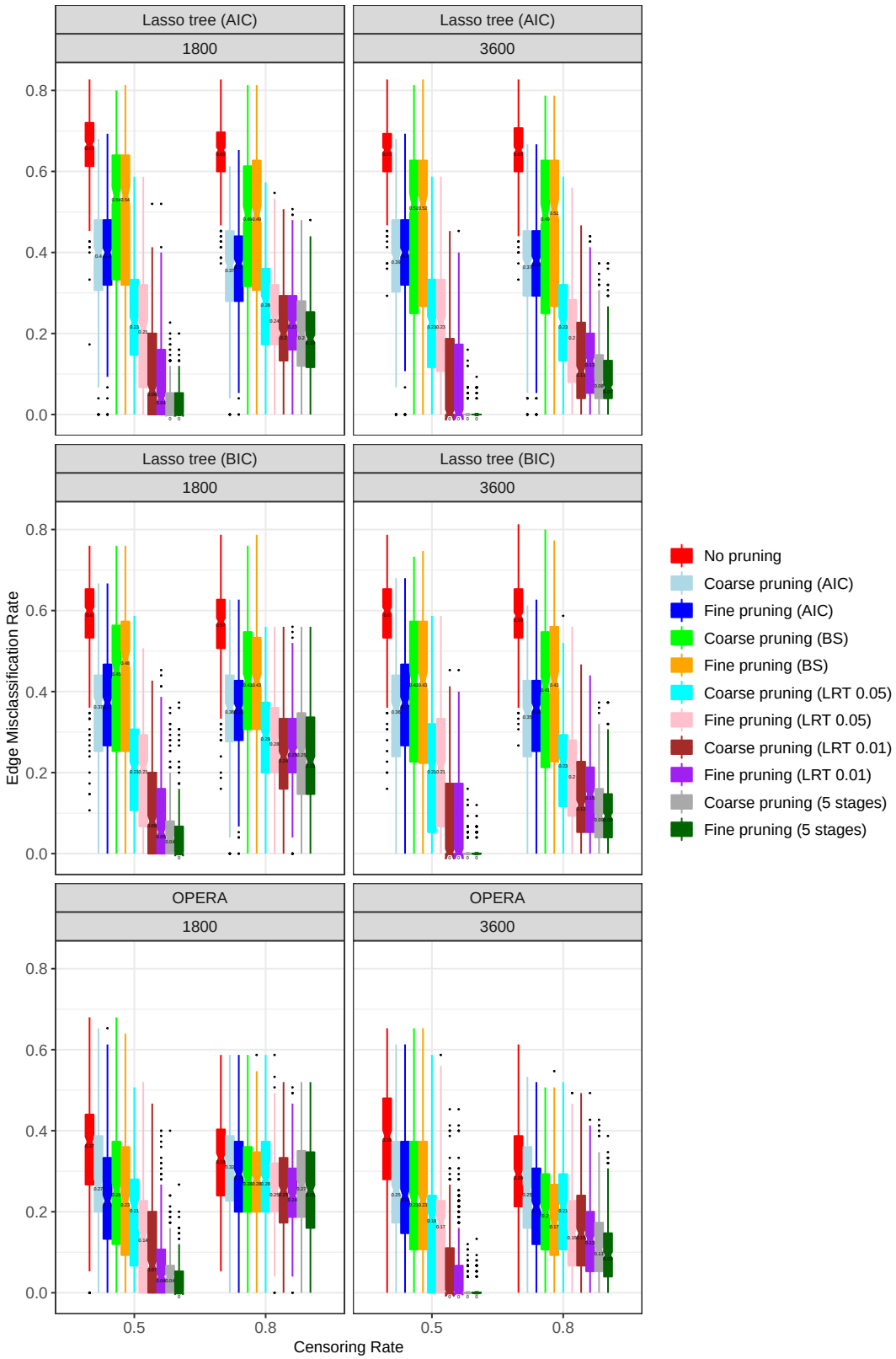


Figure 3.7: The edge misclassification rate for different pruning methods with three-risk-factor simulation scenarios favoring lasso tree

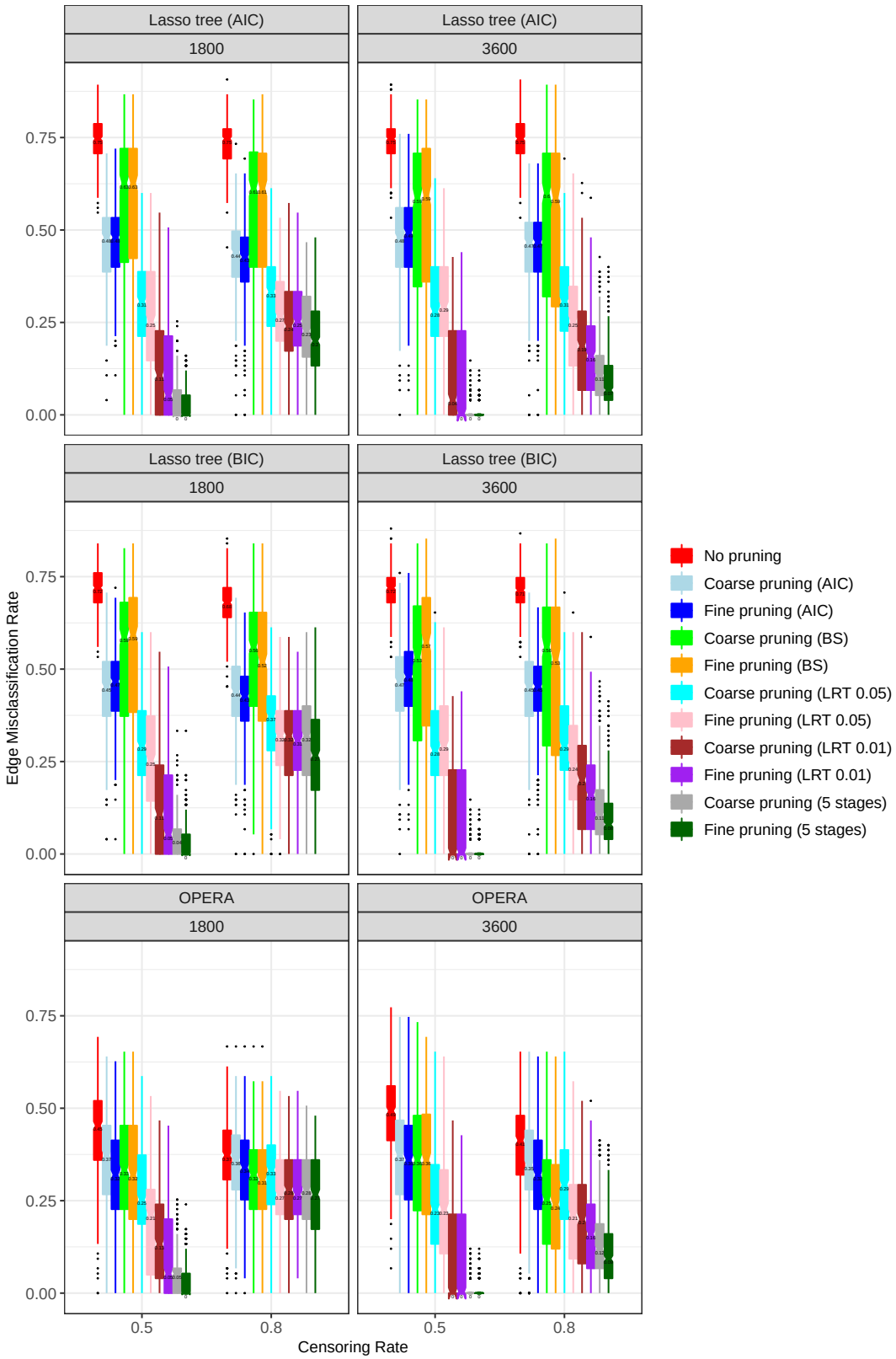


Figure 3.8: The edge misclassification rate for different pruning methods with three-risk-factor simulation scenarios favoring opera

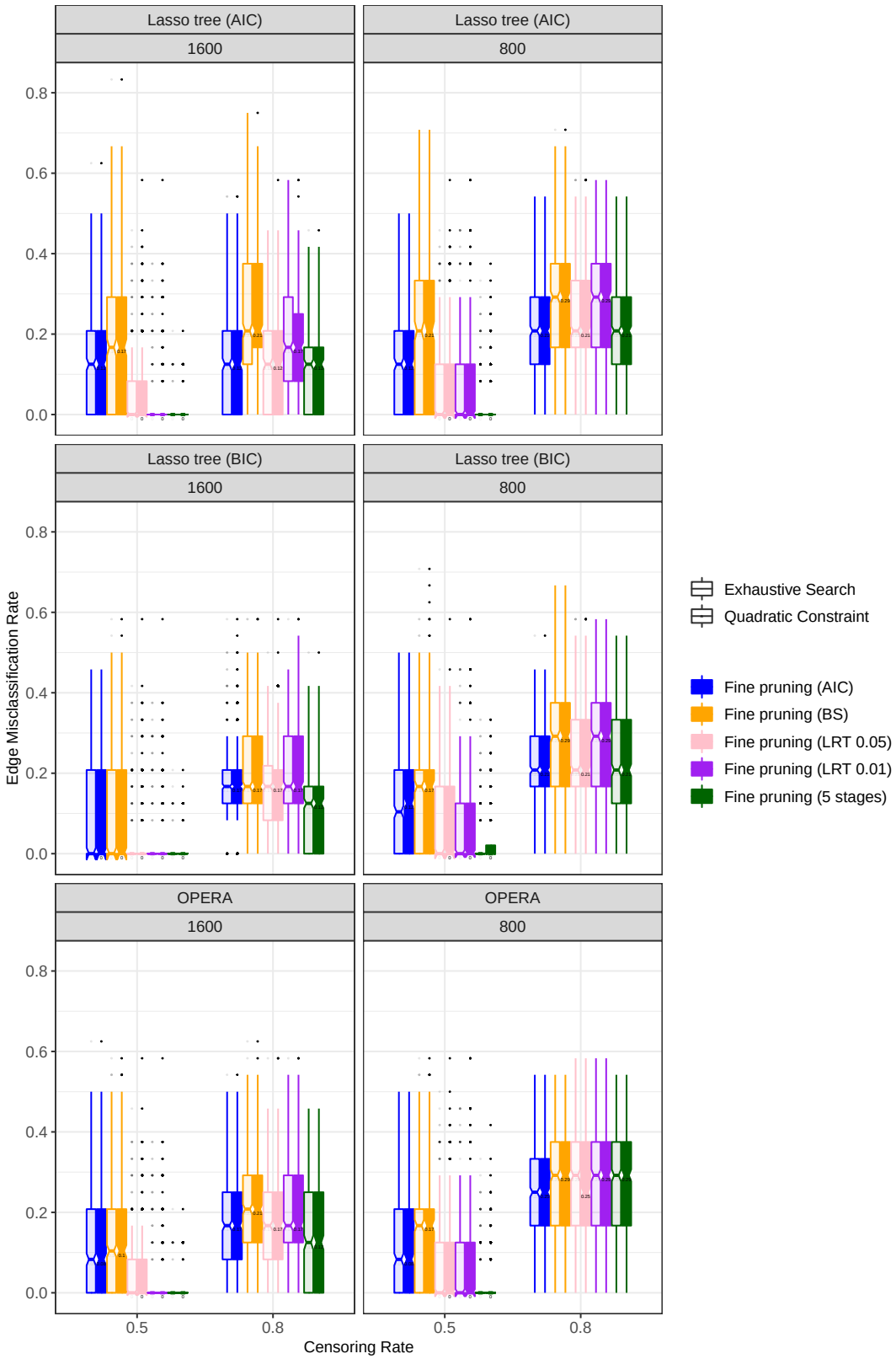


Figure 3.9: The edge misclassification rate for fine pruning methods with two-risk-factor simulation scenarios favoring lasso tree

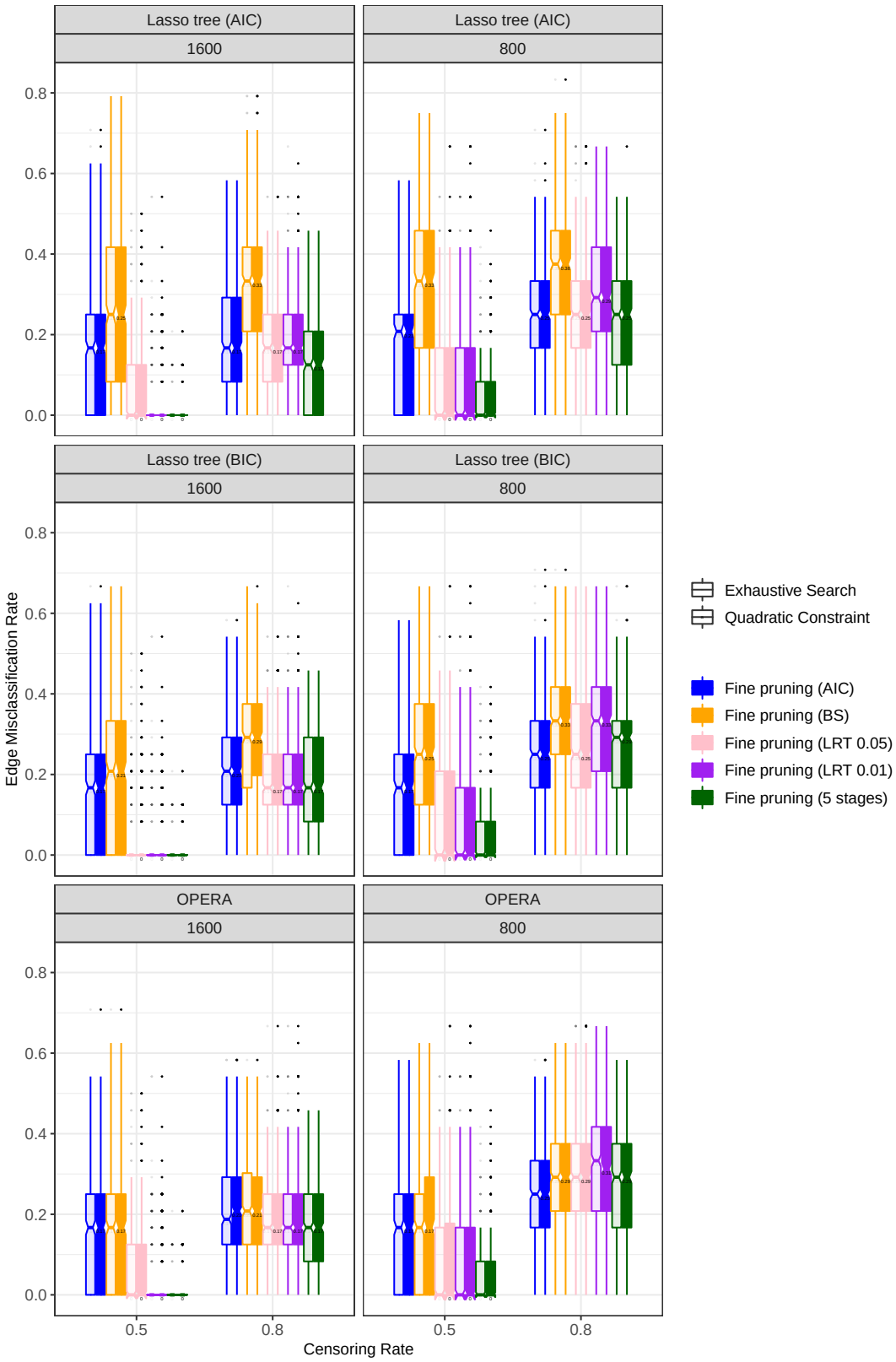


Figure 3.10: The edge misclassification rate for fine pruning methods with two-risk-factor simulation scenarios favoring opera

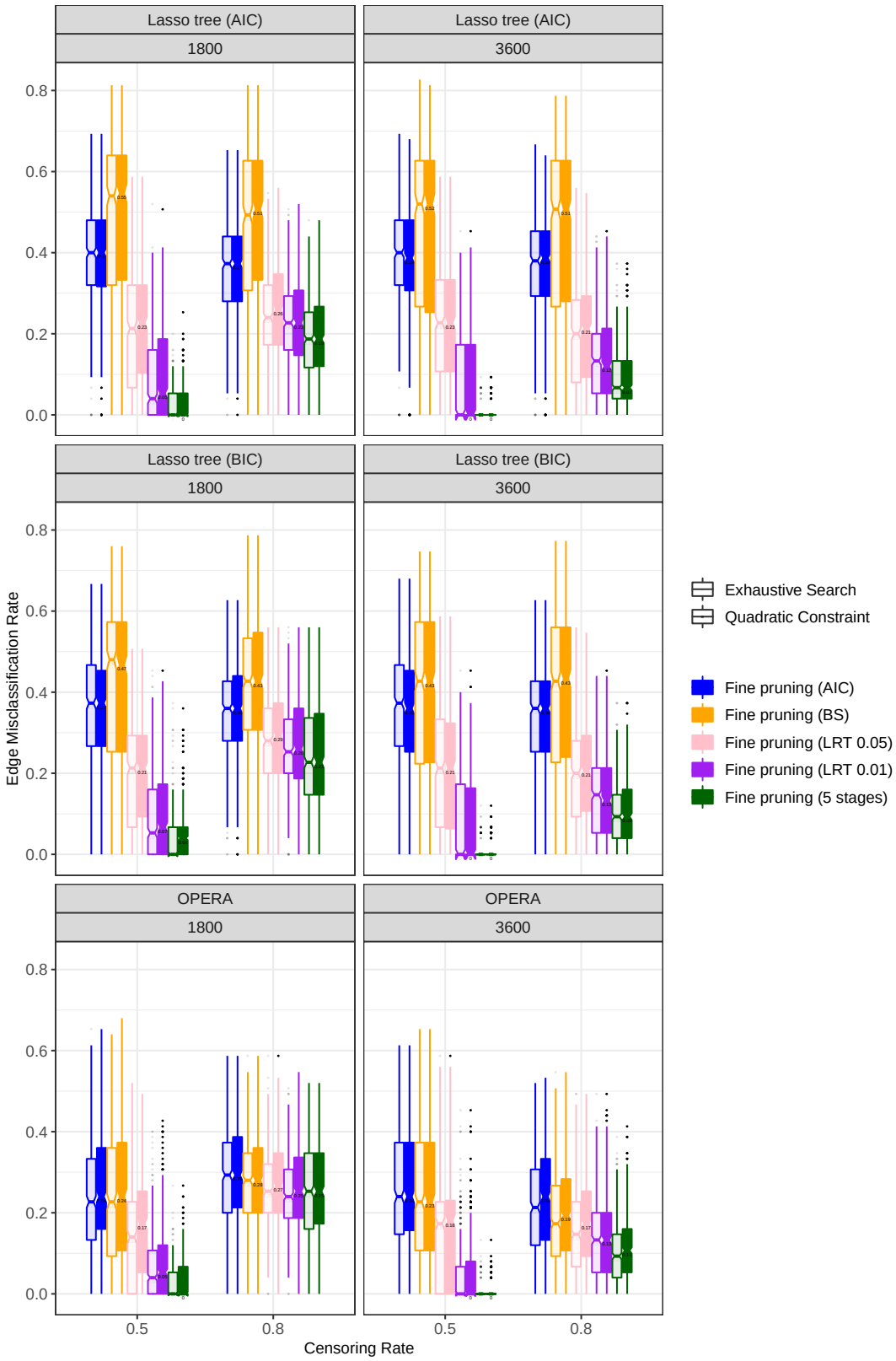


Figure 3.11: The edge misclassification rate for fine pruning methods with three-risk-factor simulation scenarios favoring lasso tree

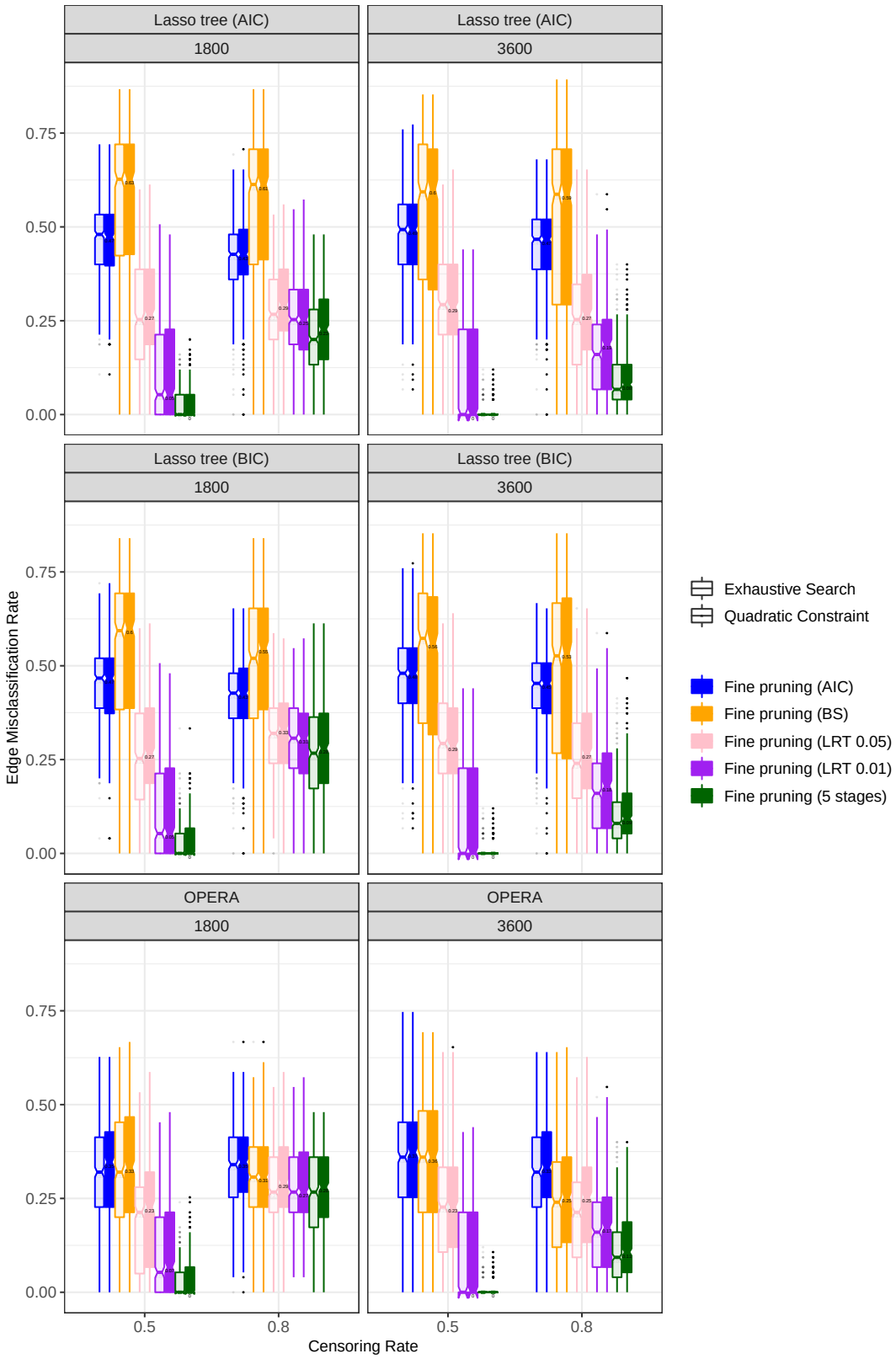


Figure 3.12: The edge misclassification rate for fine pruning methods with three-risk-factor simulation scenarios favoring opera

Chapter 4

Continuous Risk Factor

Continuous risk factors can be utilized in cancer staging when they are categorized as ordinal risk factors. If the continuous risk factor is subdivided into multiple levels, it should be possible to merge certain levels that exhibit the same staging pattern. In Figure 4.1, for example, variable a represents a continuous risk factor that is categorized into an ordinal variable consisting of six levels. However, $a1$ and $a2$ show the same staging pattern. The same applies to $a3$ and $a4$, or $a5$ and $a6$. Consequently, $a1$ and $a2$ can be combined as a single level, along with merging $a3$ and $a4$, as well as $a5$ and $a6$. This application proves highly useful when dealing with continuous risk factors. By initially dividing it into numerous levels and using this bottom-up approach, we can merge certain levels based on the same staging pattern to determine an improved threshold that simplifies the staging system, similar to pruning. For example, variable a can be age, a common continuous risk factor. Initially, we divide it into six levels from <20, $20\leq-<30$, $30\leq-<40$, $40\leq-<50$, $50\leq-<60$, ≥ 60. Upon staging, we identify the thresholds 30 and 50.

4.1 Setup

We conducted simulation studies to demonstrate the efficacy of our methods in achieving the goal of identifying thresholds and corresponding subgroups for continuous risk factors in order to simplify the cancer staging system. Our simulation studies encompass both non-neighboring patterns, as depicted in Figure 4.1, and neighboring patterns, as depicted in Figure 4.2. This distinction arises from the fact that the lasso tree can only perform staging on neighboring patterns, while OPERA can perform staging on both non-neighboring and neighboring patterns. We consider three risk factors, denoted as a, categorized into six levels: $a1$ and $a2$ form a subgroup with the same staging pattern, as do $a3$ and $a4$, and $a5$ and

$a6$. The probabilities for a patient to be in $a1, a2$, $a3, a4$, $a5$, and $a6$ are 0.1, 0.1, 0.3, 0.3, 0.1, 0.1, respectively. For the other two risk factors, each patient follows a uniform distribution to be placed in each level. Regarding a binary outcome, the coefficients assigned to stages 1 to 5 are -2, -1, 0, 1, and 2, respectively. In contrast, for a survival outcome, the coefficients assigned to stages 1 to 5 are 0, 1, 2, 3, and 4. Additional information regarding the simulation setup can be found in Chapter 2, Table 2.1, and Chapter 3, Table 3.1.

Furthermore, our simulation studies take into account both survival outcomes and binary outcomes. Regarding survival outcomes, we consider two different sample sizes: 1200 and 2400, along with two censoring rates: 0.5 and 0.8. For binary outcomes, we consider two sample sizes: 2400 and 4800.

For each simulation scenario, we compare three initial methods: lasso tree using BIC, lasso tree using AIC, and OPERA. We also consider four pruning stopping rules: (integrated) Brier score, AIC, LRT with α equal to either 0.05 or 0.01, and retaining only 5 stages. In addition, we compare three pruning methods: fine pruning using exhaustive search, fine pruning using quadratic constraint, and coarse pruning.

To evaluate each method, we utilize the probability of correctly identifying one, two, or three subgroups for the continuous risk factor. This estimation is derived from the proportion of total simulations where exactly one, two, or three subgroups are accurately identified. It is important to note that the subgroups in question are denoted as $a1\&a2$, $a3\&a4$, and $a5\&a6$.

When evaluating our methods, we assume that the true subgroups are unknown. In order to count a subgroup as discovered, it is essential that it exhibits the same staging pattern, which must differ from the staging patterns defined by the adjacent levels. For instance, if we identify $a3\&a4$ as having the same staging pattern, it is crucial to verify that this pattern is distinct from the pattern defined by either $a2$ or $a5$. Only then can we consider $a3\&a4$ as a successfully discovered subgroup.

4.2 Results

The true subgroup discovery rate refers to the estimated probability of correctly identifying one, two, or three subgroups for the continuous risk factor a. Figure 4.3 - 4.6 present the main results, comparing all methods in various simulation scenarios with either survival outcomes or binary outcomes, considering either neighboring staging patterns or non-neighboring staging patterns. All fine pruning methods are based on exhaustive search. We also consider three different initial methods, including using lasso tree (AIC), lasso tree (BIC), and OPERA.

In Figures 4.3 to 4.4, focusing on survival outcomes, regardless of the censoring rate or sample size

or the initial method, coarse pruning with LRT ($\alpha = 0.01$) demonstrates the best performance. It yields the highest estimated probability of correctly identifying all three subgroups, as well as the highest estimated probability of identifying at least one subgroup correctly, without knowing the true number of stages. Notably, when the censoring rate is 0.8, it surpasses the performance of the method with the true number of stages provided. In Figure 4.3, specifically focusing on neighboring staging patterns, coarse pruning consistently outperforms fine pruning with LRT as the stopping rule across different sample sizes and censoring rates. However, as the censoring rate decreases, the difference between the two pruning approaches becomes smaller. In Figure 4.4, which focuses on non-neighboring staging patterns, the same conclusion can be drawn, with the exception that fine pruning slightly outperforms coarse pruning in terms of slightly higher estimated probabilities of identifying all three subgroups when the sample size is 2400 and the censoring rate is 0.5. However, even in this specific scenario, the estimated probabilities of identifying at least one subgroup remain the same when comparing these two approaches. In the discussion, we will further explore the extent to which fine pruning can outperform coarse pruning.

Moving on to Figures 4.5 to 4.6, which focus on binary outcomes, LRT ($\alpha = 0.01$) continues to emerge as the optimal stopping rule for pruning without knowing the true number of stages. When the sample size is 2400 and only neighboring patterns are considered, coarse pruning with LRT ($\alpha = 0.01$) performs better. On the other hand, when the sample size is 4800, fine pruning with LRT ($\alpha = 0.01$) yields slightly superior results in terms of slightly higher estimated probabilities of identifying all three subgroups, given that the estimated probabilities of identifying at least one subgroup remain the same when comparing these two approaches. In the case of non-neighboring patterns, coarse pruning generally performs better. However, when the sample size is 2400 and OPERA is used as the initial method, there is a 0.01 increase in the estimated probability of identifying at least one subgroup. In the discussion, we will further explore the extent to which fine pruning can outperform coarse pruning.

4.3 Discussion

In summary, pruning (LRT $\alpha = 0.01$) consistently exhibits the best performance in correctly identifying at least one subgroup and all three subgroups, thereby simplifying the staging system. This remains true regardless of the outcome type, sample size, censoring rate, and initial method. As shown in Tables 4.1, 4.2, and 4.3, we also calculate the average number of discovered subgroups, which can range from 0 to 3, and select the top three pruning methods for each initial method in simulation scenarios with survival or binary outcomes and non-neighboring or neighboring staging patterns. Pruning (LRT $\alpha = 0.01$) consistently demonstrates the top three highest average number of discovered subgroups and performs

the best in the majority of cases.

To further investigate the difference in performance between coarse and fine pruning (LRT $\alpha = 0.01$), we conduct the paired sample t-test to compare the average number of discovered subgroups between the two approaches, as shown in Tables 4.4 and 4.5. For survival outcomes, the largest difference in the average number of discovered subgroups is 0.044. This indicates that in the worst case, using coarse pruning leads to a decrease of 0.044 in the average number of discovered subgroups. For binary outcomes, the largest difference is 0.036. Considering its comparable performance with fine pruning and its lower computational cost, *coarse pruning* is recommended.

Pruning can improve the initial results, enabling all methods to achieve similar levels of performance with no more than 0.28 difference in the average number of discovered subgroups, regardless of the initial method used, as shown in Tables 4.1, 4.2, and 4.3. Considering the higher computational cost associated with lasso tree methods, as indicated in Table 4.6, OPERA is suggested as the preferred initial method for survival outcomes. Meanwhile, lasso tree methods are recommended for binary outcomes due to their lower computational cost, as presented in Table 4.7.

Regarding fine pruning using the quadratic constraint or exhaustive search, we conducted the paired sample t-test to compare the average number of discovered subgroups between the two approaches with LRT ($\alpha = 0.01$), as shown in Tables 4.4 and 4.5. For survival outcomes, the largest difference in the average number of discovered subgroups is 0.106. This indicates that in the worst case, using the quadratic constraint for fine pruning leads to a decrease of 0.106 in the average number of discovered subgroups. For binary outcomes, the largest difference is 0.084. As shown in Figures 4.7-4.10, the largest difference in the true subgroup discovery rate of finding at least one subgroup is no more than 0.03. In conclusion, fine pruning using the quadratic constraint can be considered an effective alternative to fine pruning using exhaustive search.

Table 4.1: The average number of discovered subgroups for the top 3 pruning methods with different initial methods in simulation scenarios with non-neighboring staging patterns and survival outcome (Mean = Estimated Mean; SD = Estimated Standard Deviation; Lower = 95% Confidence Interval Lower Limit; Upper = 95% Confidence Interval Upper Limit; quad = using quadratic constraint; ex = using exhaustive search)

Sample Size	Censor -ing Rate	Initial Method	Pruning Method	Mean	SD	Lower	Upper
1200	0.5	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	2.452	0.676	2.393	2.511
1200	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	2.442	0.690	2.382	2.502
1200	0.5	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	2.404	0.694	2.343	2.465
1200	0.5	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	2.456	0.673	2.397	2.515
1200	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	2.452	0.690	2.391	2.513
1200	0.5	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	2.412	0.689	2.352	2.472
1200	0.5	OPERA	fine pruning quad (LRT 0.01)	2.372	0.700	2.311	2.433
1200	0.5	OPERA	fine pruning ex (LRT 0.01)	2.356	0.706	2.294	2.418
1200	0.5	OPERA	coarse pruning (LRT 0.01)	2.354	0.739	2.289	2.419
1200	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	1.860	0.826	1.788	1.932
1200	0.8	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	1.762	0.883	1.685	1.839
1200	0.8	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	1.714	0.909	1.634	1.794
1200	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	1.786	0.840	1.712	1.860
1200	0.8	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	1.716	0.877	1.639	1.793
1200	0.8	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	1.672	0.920	1.591	1.753
1200	0.8	OPERA	coarse pruning (LRT 0.01)	1.704	0.866	1.628	1.780
1200	0.8	OPERA	fine pruning ex (LRT 0.05)	1.678	0.865	1.602	1.754
1200	0.8	OPERA	fine pruning quad (LRT 0.01)	1.664	0.897	1.585	1.743
2400	0.5	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	2.756	0.541	2.709	2.803
2400	0.5	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	2.734	0.573	2.684	2.784
2400	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	2.712	0.591	2.660	2.764
2400	0.5	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	2.756	0.541	2.709	2.803
2400	0.5	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	2.734	0.573	2.684	2.784
2400	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	2.714	0.597	2.662	2.766
2400	0.5	OPERA	fine pruning ex (LRT 0.01)	2.748	0.549	2.700	2.796
2400	0.5	OPERA	fine pruning quad (LRT 0.01)	2.728	0.585	2.677	2.779

Table 4.1: The average number of discovered subgroups for the top 3 pruning methods with different initial methods in simulation scenarios with non-neighboring staging patterns and survival outcome (Mean = Estimated Mean; SD = Estimated Standard Deviation; Lower = 95% Confidence Interval Lower Limit; Upper = 95% Confidence Interval Upper Limit; quad = using quadratic constraint; ex = using exhaustive search)

Sample Size	Censoring Rate	Initial Method	Pruning Method	Mean	SD	Lower	Upper
2400	0.5	OPERA	coarse pruning (LRT 0.01)	2.714	0.607	2.661	2.767
2400	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	2.028	0.775	1.960	2.096
2400	0.8	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	2.020	0.783	1.951	2.089
2400	0.8	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	1.984	0.813	1.913	2.055
2400	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	2.004	0.768	1.937	2.071
2400	0.8	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	2.004	0.768	1.937	2.071
2400	0.8	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	1.966	0.804	1.896	2.036
2400	0.8	OPERA	fine pruning quad (LRT 0.01)	1.964	0.825	1.892	2.036
2400	0.8	OPERA	coarse pruning (LRT 0.01)	1.940	0.813	1.869	2.011
2400	0.8	OPERA	fine pruning ex (LRT 0.01)	1.932	0.837	1.859	2.005

Table 4.2: The average number of discovered subgroups for the top 3 pruning methods with different initial methods in simulation scenarios with neighboring staging patterns and survival outcome

Sample Size	Censor -ing Rate	Initial Method	Pruning Method	Mean	SD	Lower	Upper
1200	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	2.424	0.833	2.351	2.497
1200	0.5	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	2.368	0.889	2.290	2.446
1200	0.5	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	2.344	0.907	2.264	2.424
1200	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	2.440	0.817	2.368	2.512
1200	0.5	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	2.352	0.891	2.274	2.430
1200	0.5	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	2.336	0.910	2.256	2.416
1200	0.5	OPERA	coarse pruning (LRT 0.01)	2.318	0.850	2.244	2.392
1200	0.5	OPERA	fine pruning quad (LRT 0.01)	2.296	0.868	2.220	2.372
1200	0.5	OPERA	fine pruning ex (LRT 0.05)	2.260	0.868	2.184	2.336
1200	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.05)	1.428	1.001	1.340	1.516
1200	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	1.420	0.997	1.333	1.507
1200	0.8	Lasso tree (AIC)	fine pruning ex (AIC)	1.388	0.977	1.302	1.474
1200	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.05)	1.436	1.004	1.348	1.524
1200	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	1.424	0.999	1.336	1.512
1200	0.8	Lasso tree (BIC)	coarse pruning (AIC)	1.394	0.972	1.309	1.479
1200	0.8	OPERA	coarse pruning (LRT 0.01)	1.148	0.894	1.070	1.226
1200	0.8	OPERA	coarse pruning (LRT 0.05)	1.142	0.912	1.062	1.222
1200	0.8	OPERA	no pruning	1.132	0.895	1.054	1.210
2400	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	2.770	0.571	2.720	2.820
2400	0.5	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	2.742	0.603	2.689	2.795
2400	0.5	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	2.728	0.625	2.673	2.783
2400	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	2.784	0.549	2.736	2.832
2400	0.5	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	2.744	0.603	2.691	2.797
2400	0.5	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	2.732	0.624	2.677	2.787
2400	0.5	OPERA	coarse pruning (LRT 0.01)	2.774	0.505	2.730	2.818
2400	0.5	OPERA	fine pruning quad (LRT 0.01)	2.766	0.525	2.720	2.812
2400	0.5	OPERA	fine pruning ex (LRT 0.01)	2.738	0.574	2.688	2.788

Table 4.2: The average number of discovered subgroups for the top 3 pruning methods with different initial methods in simulation scenarios with neighboring staging patterns and survival outcome

Sample Size	Censor -ing Rate	Initial Method	Pruning Method	Mean	SD	Lower	Upper
2400	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	1.910	0.971	1.825	1.995
2400	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.05)	1.836	0.983	1.750	1.922
2400	0.8	Lasso tree (AIC)	fine pruning ex (LRT 0.05)	1.732	0.991	1.645	1.819
2400	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	1.904	0.953	1.820	1.988
2400	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.05)	1.834	0.963	1.750	1.918
2400	0.8	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	1.736	1.004	1.648	1.824
2400	0.8	OPERA	coarse pruning (LRT 0.01)	1.710	0.994	1.623	1.797
2400	0.8	OPERA	fine pruning ex (LRT 0.05)	1.676	0.984	1.590	1.762
2400	0.8	OPERA	fine pruning ex (AIC)	1.672	0.973	1.587	1.757
2400	0.8	OPERA	coarse pruning (LRT 0.05)	1.670	0.994	1.583	1.757

Table 4.3: The average number of discovered subgroups for the top 3 pruning methods with different initial methods in simulation scenarios with binary outcome

	Sample Size	Initial Method	Pruning Method	Mean	SD	Lower	Upper
Non-neighbor-ing	2400	Lasso tree (AIC)	coarse pruning (LRT 0.01)	2.440	0.660	2.382	2.498
	2400	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	2.342	0.700	2.281	2.403
	2400	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	2.334	0.751	2.268	2.400
	2400	Lasso tree (BIC)	coarse pruning (LRT 0.01)	2.478	0.650	2.421	2.535
	2400	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	2.354	0.697	2.293	2.415
	2400	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	2.352	0.757	2.286	2.418
	2400	OPERA	fine pruning ex (LRT 0.01)	2.248	0.701	2.187	2.309
	2400	OPERA	coarse pruning (LRT 0.01)	2.224	0.703	2.162	2.286
	2400	OPERA	fine pruning ex (LRT 0.05)	2.184	0.772	2.116	2.252
	4800	Lasso tree (AIC)	coarse pruning (LRT 0.01)	2.740	0.552	2.692	2.788
	4800	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	2.696	0.580	2.645	2.747
	4800	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	2.652	0.593	2.600	2.704
	4800	Lasso tree (BIC)	coarse pruning (LRT 0.01)	2.776	0.488	2.733	2.819
	4800	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	2.708	0.565	2.658	2.758
	4800	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	2.658	0.581	2.607	2.709
	4800	OPERA	coarse pruning (LRT 0.01)	2.634	0.587	2.583	2.685
	4800	OPERA	fine pruning quad (LRT 0.01)	2.602	0.620	2.548	2.656
	4800	OPERA	fine pruning ex (LRT 0.01)	2.590	0.595	2.538	2.642

Table 4.3: The average number of discovered subgroups for the top 3 pruning methods with different initial methods in simulation scenarios with binary outcome

	Sample Size	Initial Method	Pruning Method	Mean	SD	Lower	Upper
	2400	Lasso tree (AIC)	coarse pruning (LRT 0.01)	2.526	0.712	2.464	2.588
	2400	Lasso tree (AIC)	coarse pruning (LRT 0.05)	2.408	0.766	2.341	2.475
	2400	Lasso tree (AIC)	fine pruning quad (LRT 0.05)	2.396	0.800	2.326	2.466
	2400	Lasso tree (BIC)	coarse pruning (LRT 0.01)	2.576	0.682	2.516	2.636
	2400	Lasso tree (BIC)	coarse pruning (LRT 0.05)	2.480	0.717	2.417	2.543
	2400	Lasso tree (BIC)	fine pruning quad (LRT 0.05)	2.446	0.751	2.380	2.512
	2400	OPERA	coarse pruning (LRT 0.01)	2.282	0.777	2.214	2.350
	2400	OPERA	fine pruning ex (LRT 0.05)	2.220	0.811	2.149	2.291
Neighbor	2400	OPERA	fine pruning ex (LRT 0.01)	2.214	0.830	2.141	2.287
-ing	4800	Lasso tree (AIC)	fine pruning ex (LRT 0.01)	2.866	0.390	2.832	2.900
	4800	Lasso tree (AIC)	coarse pruning (LRT 0.01)	2.852	0.422	2.815	2.889
	4800	Lasso tree (AIC)	fine pruning quad (LRT 0.01)	2.788	0.547	2.740	2.836
	4800	Lasso tree (BIC)	fine pruning ex (LRT 0.01)	2.868	0.388	2.834	2.902
	4800	Lasso tree (BIC)	coarse pruning (LRT 0.01)	2.864	0.402	2.829	2.899
	4800	Lasso tree (BIC)	fine pruning quad (LRT 0.01)	2.818	0.515	2.773	2.863
	4800	OPERA	fine pruning ex (LRT 0.01)	2.752	0.464	2.711	2.793
	4800	OPERA	coarse pruning (LRT 0.01)	2.716	0.510	2.671	2.761
	4800	OPERA	fine pruning quad (LRT 0.01)	2.668	0.599	2.616	2.720

Table 4.4: The pairwise comparisons in the average number of discovered subgroups among coarse pruning, fine pruning using exhaustive search and fine pruning using quadratic constraint with LRT ($\alpha = 0.01$) in simulation scenarios with survival outcome (Each cell displays the difference with the corresponding p-value in parentheses)

	Sample Size	Censoring Rate	Initial Method	Coarse vs Ex	Coarse vs Quad	Ex vs Quad
Non-neighbor-ing	1200	0.5	OPERA	-0.002 (0.942)	-0.018 (0.389)	-0.016 (0.462)
	1200	0.5	Lasso tree (AIC)	0.038 (0.122)	-0.010 (0.569)	-0.048 (0.023)
	1200	0.5	Lasso tree (BIC)	0.040 (0.105)	-0.004 (0.831)	-0.044 (0.036)
	1200	0.8	OPERA	0.048 (0.172)	0.040 (0.096)	-0.008 (0.813)
	1200	0.8	Lasso tree (AIC)	0.146 (< 0.001)	0.098 (< 0.001)	-0.048 (0.160)
	1200	0.8	Lasso tree (BIC)	0.114 (0.002)	0.070 (0.009)	-0.044 (0.168)
	2400	0.5	OPERA	-0.034 (0.038)	-0.014 (0.346)	0.020 (0.086)
	2400	0.5	Lasso tree (AIC)	**-0.044** (0.006)	-0.022 (0.071)	0.022 (0.071)
	2400	0.5	Lasso tree (BIC)	-0.042 (0.008)	-0.02 (0.096)	0.022 (0.071)
	2400	0.8	OPERA	0.008 (0.782)	-0.024 (0.257)	-0.032 (0.251)
	2400	0.8	Lasso tree (AIC)	0.044 (0.166)	0.008 (0.752)	-0.036 (0.223)
	2400	0.8	Lasso tree (BIC)	0.038 (0.223)	0.000 (1.000)	-0.038 (0.189)
Neighbor-ing	1200	0.5	OPERA	0.062 (0.039)	0.022 (0.279)	-0.04 (0.103)
	1200	0.5	Lasso tree (AIC)	0.080 (0.007)	0.056 (0.004)	-0.024 (0.352)
	1200	0.5	Lasso tree (BIC)	0.104 (0.001)	0.088 (< 0.001)	-0.016 (0.530)
	1200	0.8	OPERA	0.168 (< 0.001)	0.130 (< 0.001)	-0.038 (0.037)
	1200	0.8	Lasso tree (AIC)	0.248 (< 0.001)	0.234 (< 0.001)	-0.014 (0.538)
	1200	0.8	Lasso tree (BIC)	0.276 (< 0.001)	0.262 (< 0.001)	-0.014 (0.562)
	2400	0.5	OPERA	0.036 (0.075)	0.008 (0.642)	-0.028 (0.027)
	2400	0.5	Lasso tree (AIC)	0.042 (0.050)	0.028 (0.104)	-0.014 (0.337)
	2400	0.5	Lasso tree (BIC)	0.052 (0.019)	0.040 (0.027)	-0.012 (0.406)
	2400	0.8	OPERA	0.134 (< 0.001)	0.068 (< 0.001)	-0.066 (0.024)
	2400	0.8	Lasso tree (AIC)	0.278 (< 0.001)	0.188 (< 0.001)	-0.090 (0.004)
	2400	0.8	Lasso tree (BIC)	0.274 (< 0.001)	0.168 (< 0.001)	**-0.106** (0.001)

Table 4.5: The pairwise comparisons in the average number of discovered subgroups among coarse pruning, fine pruning using exhaustive search and fine pruning using quadratic constraint with LRT ($\alpha = 0.01$) in simulation scenarios with binary outcome (Each cell displays the difference with the corresponding p-value in parentheses)

	Sample Size	Initial Method	Coarse vs Ex	Coarse vs Quad	Ex vs Quad
Non-neighbor-ing	2400	OPERA	-0.024 (0.387)	0.052 (0.067)	0.076 (0.009)
	2400	Lasso tree (AIC)	0.098 (0.001)	0.106 (0.001)	0.008 (0.794)
	2400	Lasso tree (BIC)	0.124 (< 0.001)	0.126 (< 0.001)	0.002 (0.945)
	4800	OPERA	0.044 (0.05)	0.032 (0.113)	-0.012 (0.605)
	4800	Lasso tree (AIC)	0.088 (< 0.001)	0.044 (0.048)	-0.044 (0.067)
	4800	Lasso tree (BIC)	0.118 (< 0.001)	0.068 (0.001)	-0.05 (0.036)
Neighbor-ing	2400	OPERA	0.068 (0.021)	0.124 (< 0.001)	0.056 (0.054)
	2400	Lasso tree (AIC)	0.150 (< 0.001)	0.150 (< 0.001)	0.000 (1.000)
	2400	Lasso tree (BIC)	0.172 (< 0.001)	0.152 (< 0.001)	-0.020 (0.499)
	4800	OPERA	**-0.036** (0.007)	0.048 (0.005)	**0.084** (< 0.001)
	4800	Lasso tree (AIC)	-0.014 (0.209)	0.064 (0.002)	0.078 (< 0.001)
	4800	Lasso tree (BIC)	-0.004 (0.695)	0.046 (0.014)	0.050 (0.009)

Table 4.6: The average time for each initial method along with coarse pruning with LRT(0.01) in simulation scenarios with survival outcome

	Sample Size	Censoring Rate	Initial Method	Pruning Method	Average (s)	Total (s)
	1200	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	95	
	1200	0.5	Lasso tree (AIC)	no pruning	1422	1517
	1200	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	92	
	1200	0.5	Lasso tree (BIC)	no pruning	1567	1660
	1200	0.5	OPERA	coarse pruning (LRT 0.01)	37	
	1200	0.5	OPERA	no pruning	42	80
	1200	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	84	
	1200	0.8	Lasso tree (AIC)	no pruning	255	339
	1200	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	69	
	1200	0.8	Lasso tree (BIC)	no pruning	275	345
	1200	0.8	OPERA	coarse pruning (LRT 0.01)	28	
Non-	1200	0.8	OPERA	no pruning	24	52
neighboring	2400	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	310	
	2400	0.5	Lasso tree (AIC)	no pruning	15294	15604
	2400	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	283	
	2400	0.5	Lasso tree (BIC)	no pruning	14865	15148
	2400	0.5	OPERA	coarse pruning (LRT 0.01)	123	
	2400	0.5	OPERA	no pruning	239	362
	2400	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	268	
	2400	0.8	Lasso tree (AIC)	no pruning	6408	6676
	2400	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	266	
	2400	0.8	Lasso tree (BIC)	no pruning	6817	7083
	2400	0.8	OPERA	coarse pruning (LRT 0.01)	100	
	2400	0.8	OPERA	no pruning	129	229

Table 4.6: The average time for each initial method along with coarse pruning with LRT(0.01) in simulation scenarios with survival outcome

	Sample Size	Censoring Rate	Initial Method	Pruning Method	Average (s)	Total (s)
Neighboring	1200	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	28	
	1200	0.5	Lasso tree (AIC)	no pruning	247	275
	1200	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	26	
	1200	0.5	Lasso tree (BIC)	no pruning	247	273
	1200	0.5	OPERA	coarse pruning (LRT 0.01)	15	
	1200	0.5	OPERA	no pruning	15	30
	1200	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	27	
	1200	0.8	Lasso tree (AIC)	no pruning	108	135
	1200	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	24	
	1200	0.8	Lasso tree (BIC)	no pruning	109	134
	1200	0.8	OPERA	coarse pruning (LRT 0.01)	11	
	1200	0.8	OPERA	no pruning	8	19
	2400	0.5	Lasso tree (AIC)	coarse pruning (LRT 0.01)	223	
	2400	0.5	Lasso tree (AIC)	no pruning	15020	15243
	2400	0.5	Lasso tree (BIC)	coarse pruning (LRT 0.01)	224	
	2400	0.5	Lasso tree (BIC)	no pruning	14590	14813
	2400	0.5	OPERA	coarse pruning (LRT 0.01)	118	
	2400	0.5	OPERA	no pruning	224	342
	2400	0.8	Lasso tree (AIC)	coarse pruning (LRT 0.01)	220	
	2400	0.8	Lasso tree (AIC)	no pruning	6483	6704
	2400	0.8	Lasso tree (BIC)	coarse pruning (LRT 0.01)	214	
	2400	0.8	Lasso tree (BIC)	no pruning	6903	7117
	2400	0.8	OPERA	coarse pruning (LRT 0.01)	103	
	2400	0.8	OPERA	no pruning	142	246

Table 4.7: The average time for each initial method along with coarse pruning with LRT(0.01) in simulation scenarios with binary outcome

	Sample Size	Initial Method	Pruning Method	Average (s)	Total (s)
Non-neighboring	2400	Lasso tree (AIC)	coarse pruning (LRT 0.01)	68	
	2400	Lasso tree (AIC)	no pruning	45	113
	2400	Lasso tree (BIC)	coarse pruning (LRT 0.01)	49	
	2400	Lasso tree (BIC)	no pruning	46	95
	2400	OPERA	coarse pruning (LRT 0.01)	16	
	2400	OPERA	no pruning	98	114
	4800	Lasso tree (AIC)	coarse pruning (LRT 0.01)	393	
	4800	Lasso tree (AIC)	no pruning	295	688
	4800	Lasso tree (BIC)	coarse pruning (LRT 0.01)	424	
	4800	Lasso tree (BIC)	no pruning	414	838
	4800	OPERA	coarse pruning (LRT 0.01)	197	
	4800	OPERA	no pruning	755	952
Neighboring	2400	Lasso tree (AIC)	coarse pruning (LRT 0.01)	92	
	2400	Lasso tree (AIC)	no pruning	135	228
	2400	Lasso tree (BIC)	coarse pruning (LRT 0.01)	59	
	2400	Lasso tree (BIC)	no pruning	142	201
	2400	OPERA	coarse pruning (LRT 0.01)	39	
	2400	OPERA	no pruning	247	286
	4800	Lasso tree (AIC)	coarse pruning (LRT 0.01)	205	
	4800	Lasso tree (AIC)	no pruning	337	542
	4800	Lasso tree (BIC)	coarse pruning (LRT 0.01)	147	
	4800	Lasso tree (BIC)	no pruning	404	551
	4800	OPERA	coarse pruning (LRT 0.01)	192	
	4800	OPERA	no pruning	1174	1366

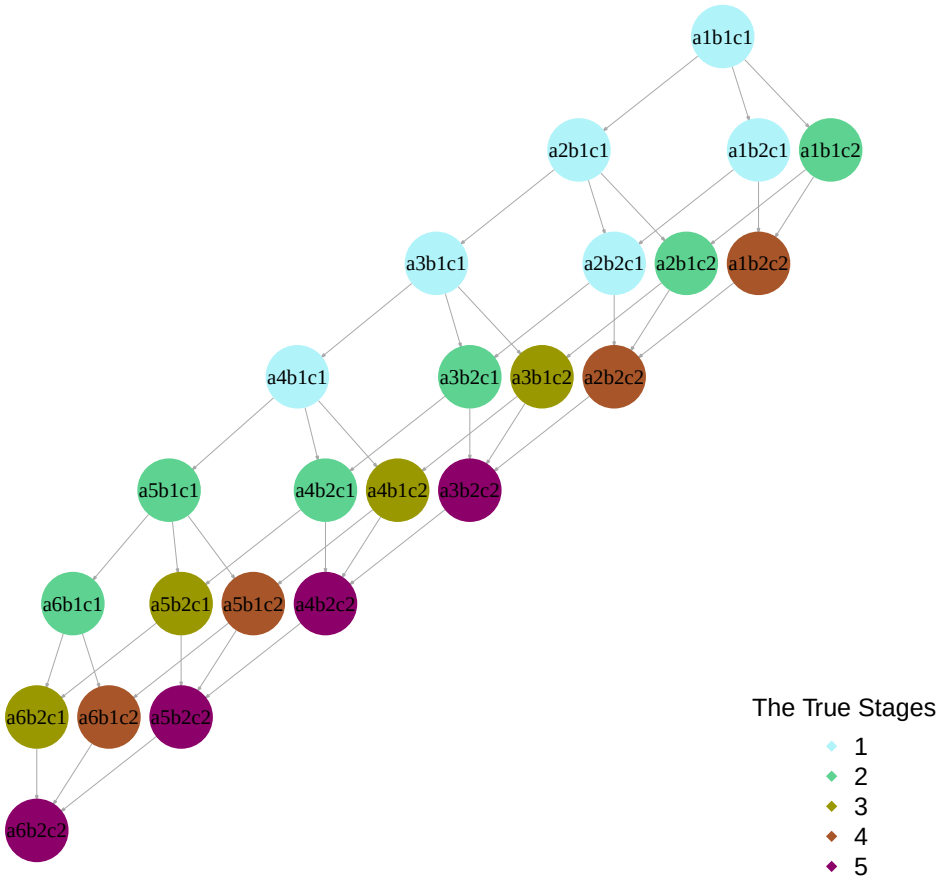


Figure 4.1: The network defined by a continuous risk factor (a) and two ordinal risk factors (b, c) with non-neighbouring staging patterns

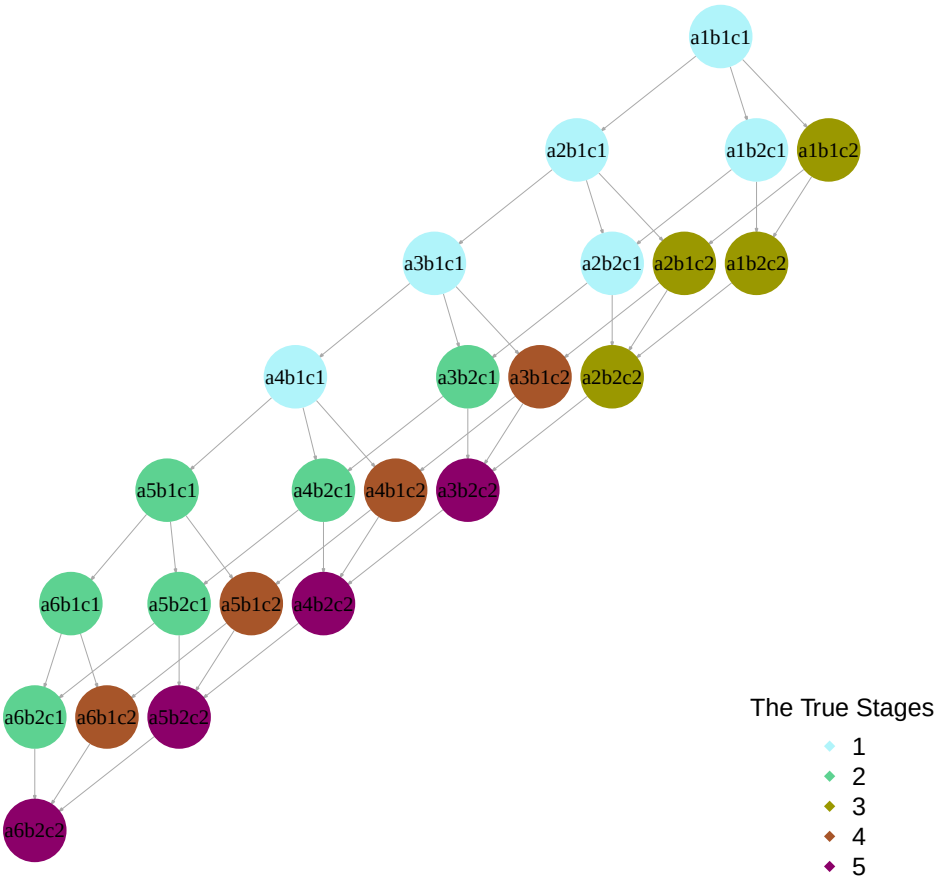


Figure 4.2: The network defined by a continuous risk factor (a) and two ordinal risk factors (b, c) with neighbouring staging patterns

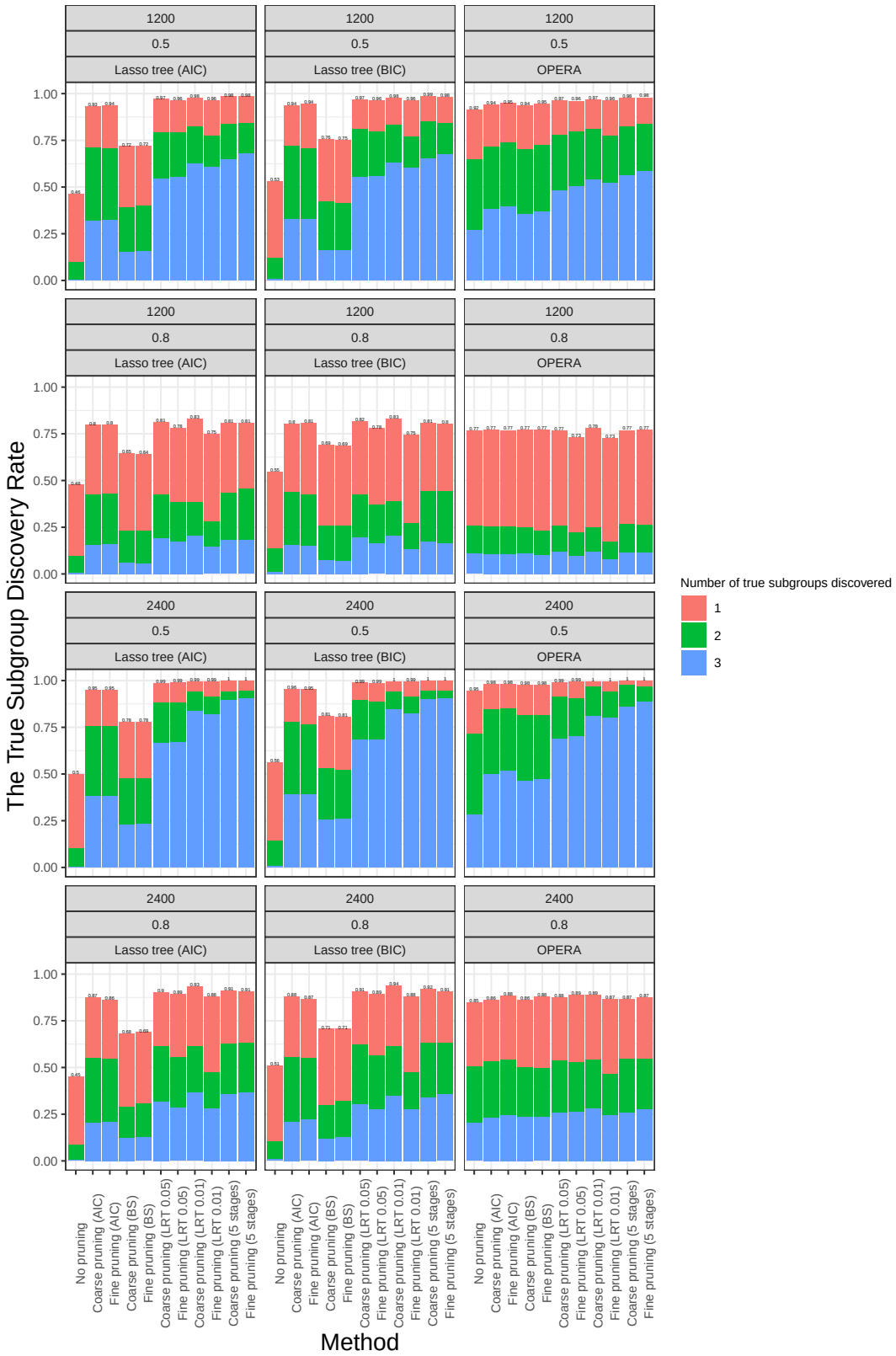


Figure 4.3: The true subgroup discovery rate for a survival outcome with neighbouring staging patterns

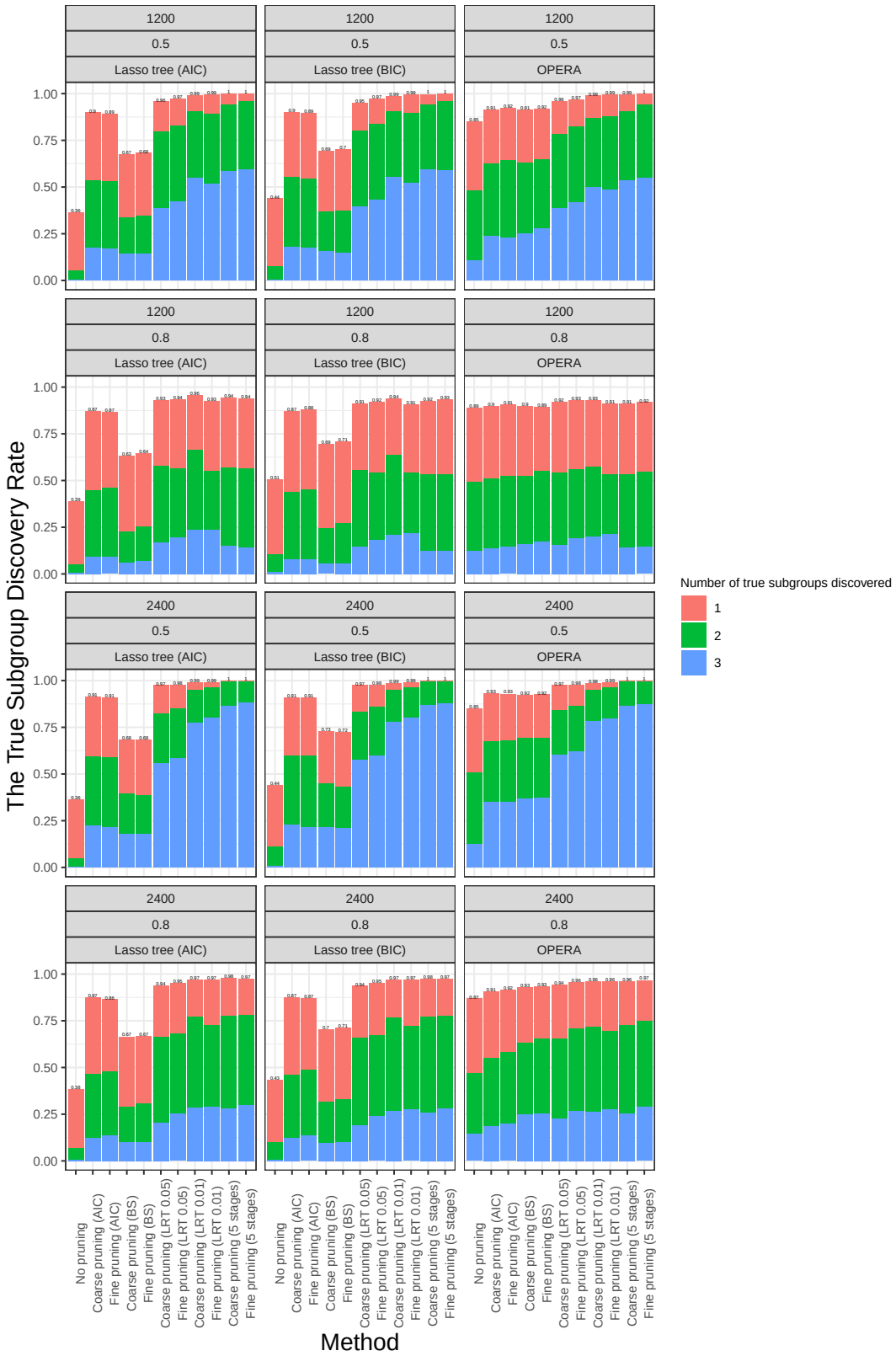


Figure 4.4: The true subgroup discovery rate for a survival outcome with non-neighbouring staging patterns

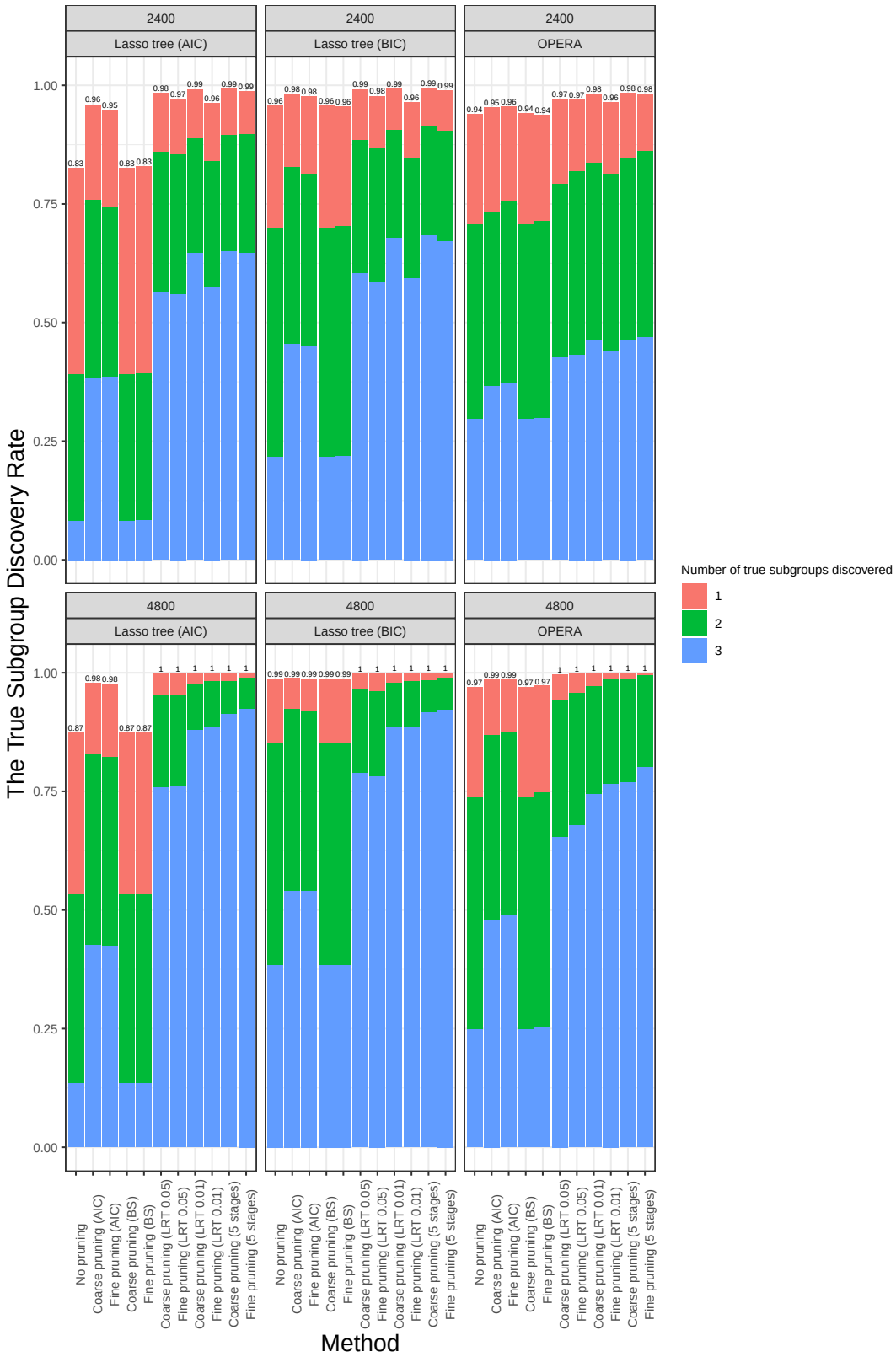


Figure 4.5: The true subgroup discovery rate for a binary outcome with neighbouring staging patterns

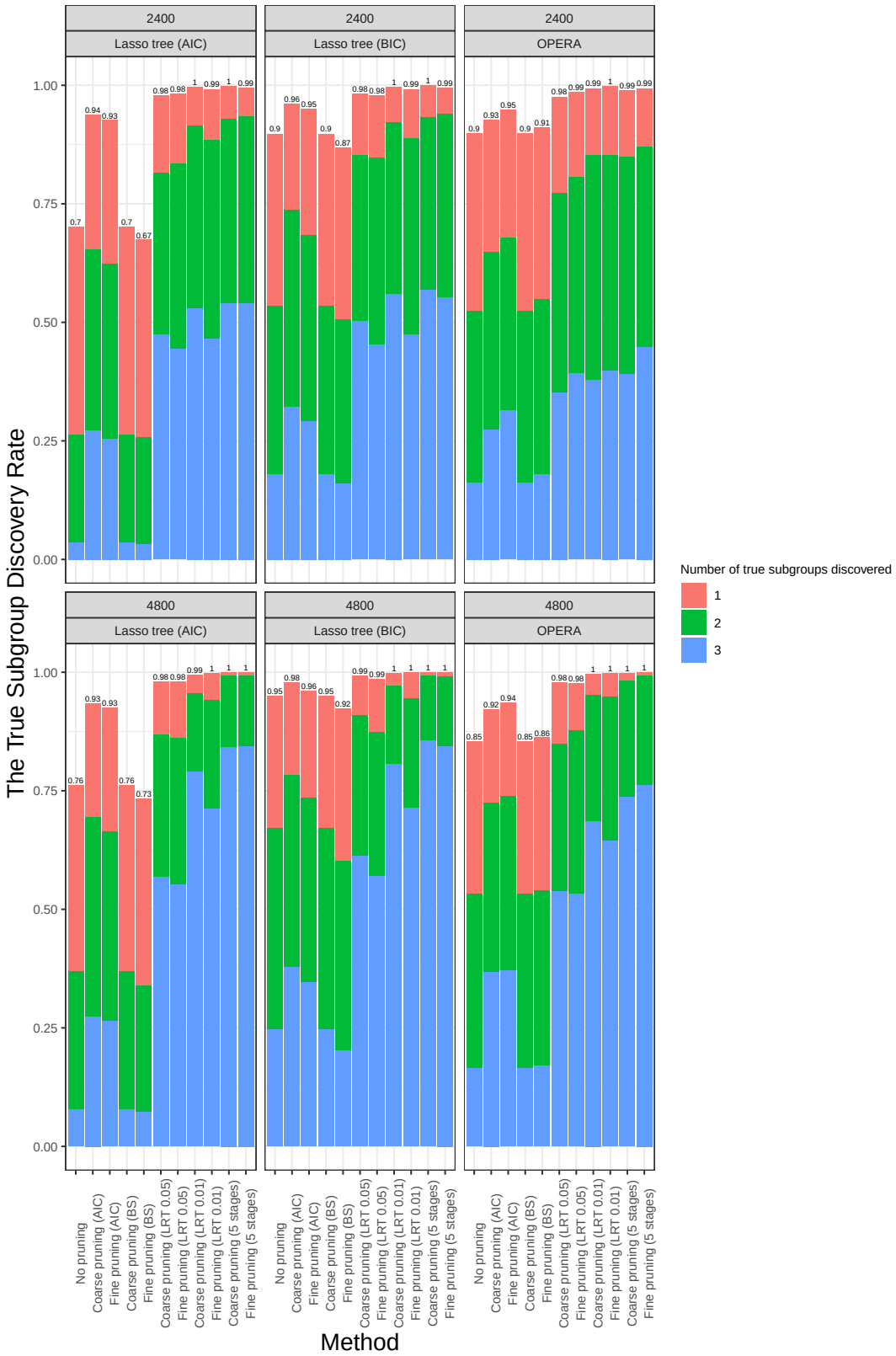


Figure 4.6: The true subgroup discovery rate for a binary outcome with non-neighbouring staging patterns

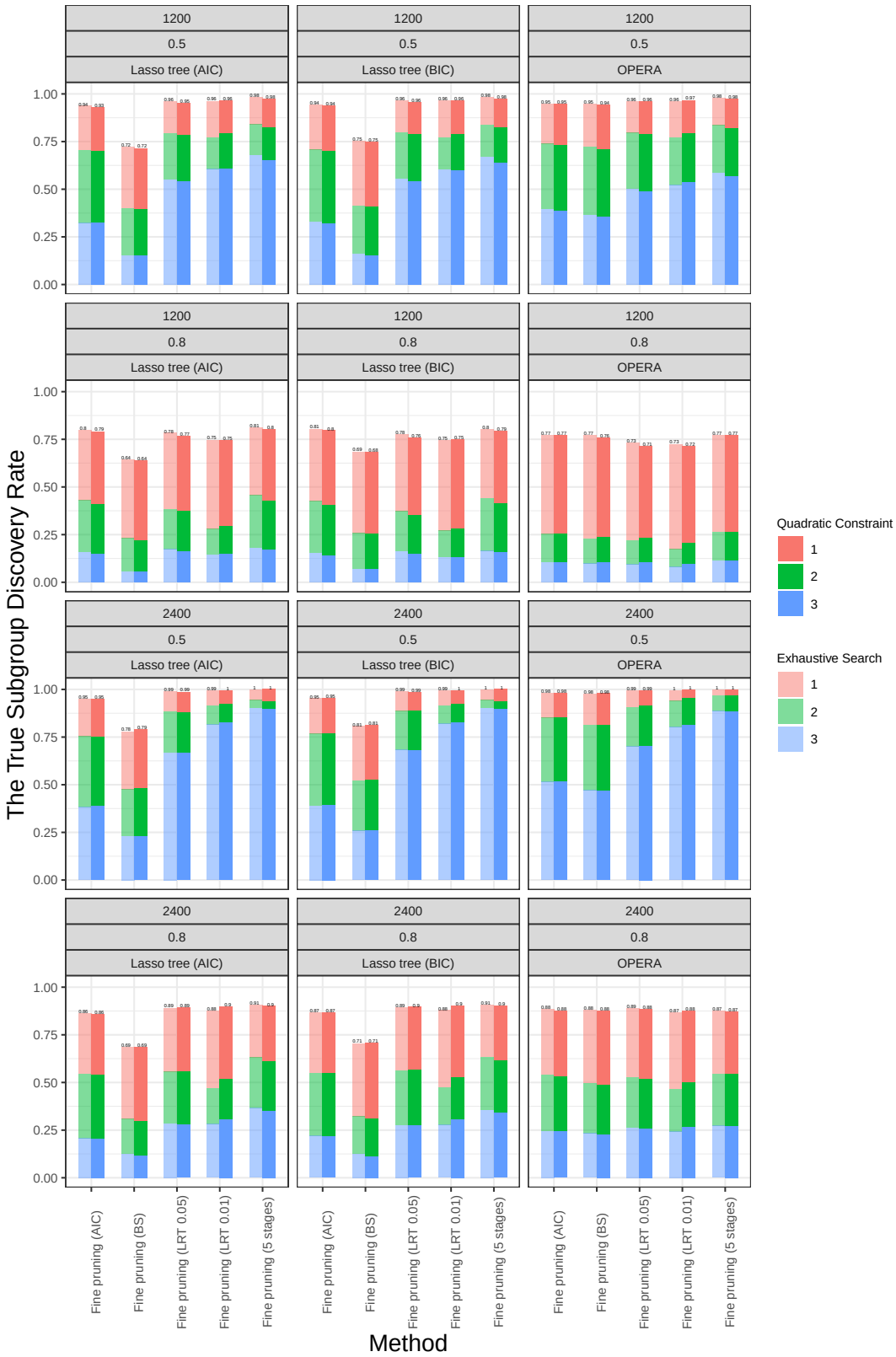


Figure 4.7: The true subgroup discovery rate using quadratic constraint and exhaustive search for a survival outcome with neighbouring staging patterns

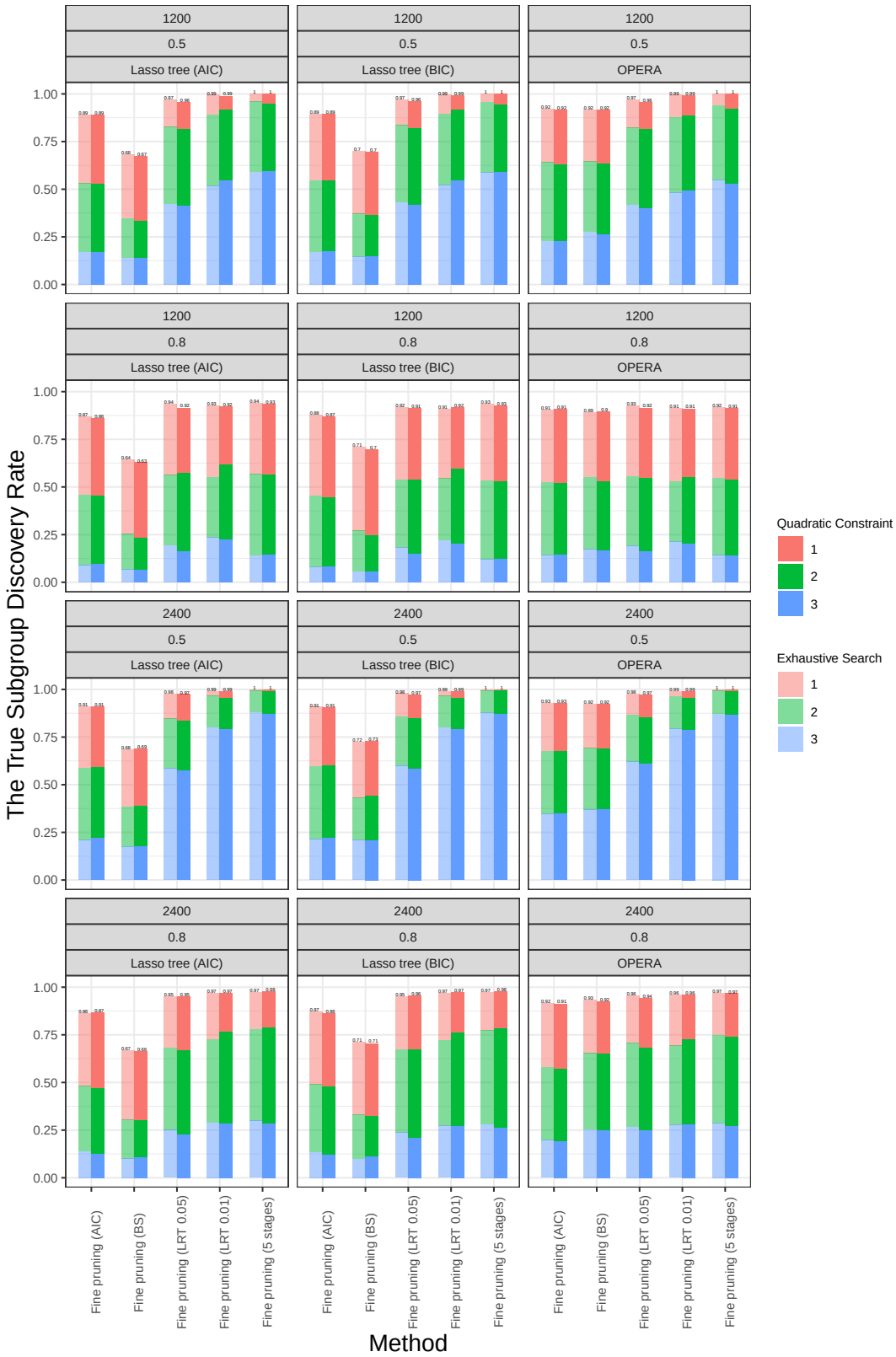


Figure 4.8: The true subgroup discovery rate using quadratic constraint and exhaustive search for a survival outcome with non-neighbouring staging patterns

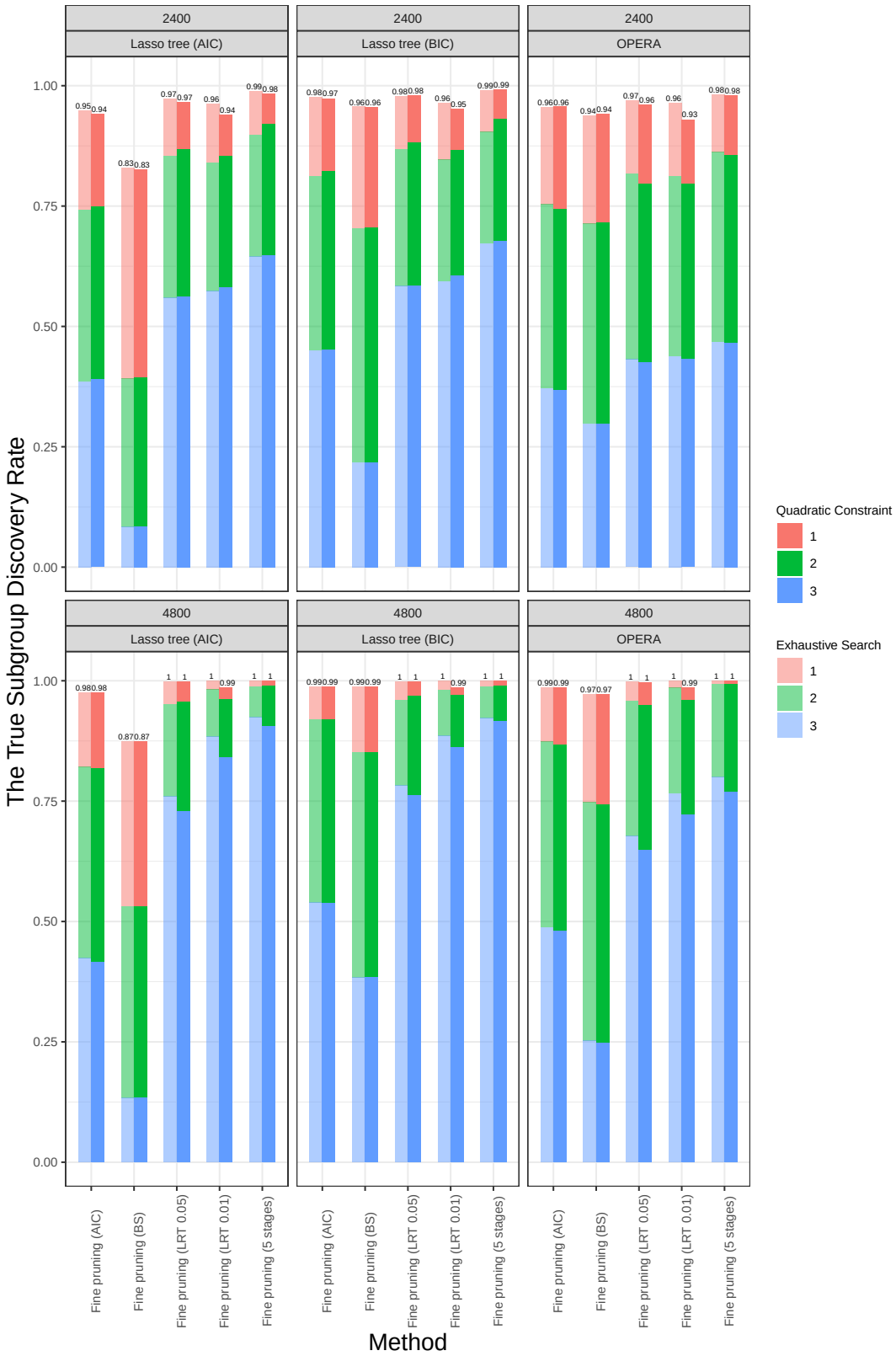


Figure 4.9: The true subgroup discovery rate using quadratic constraint and exhaustive search for a binary outcome with neighbouring staging patterns

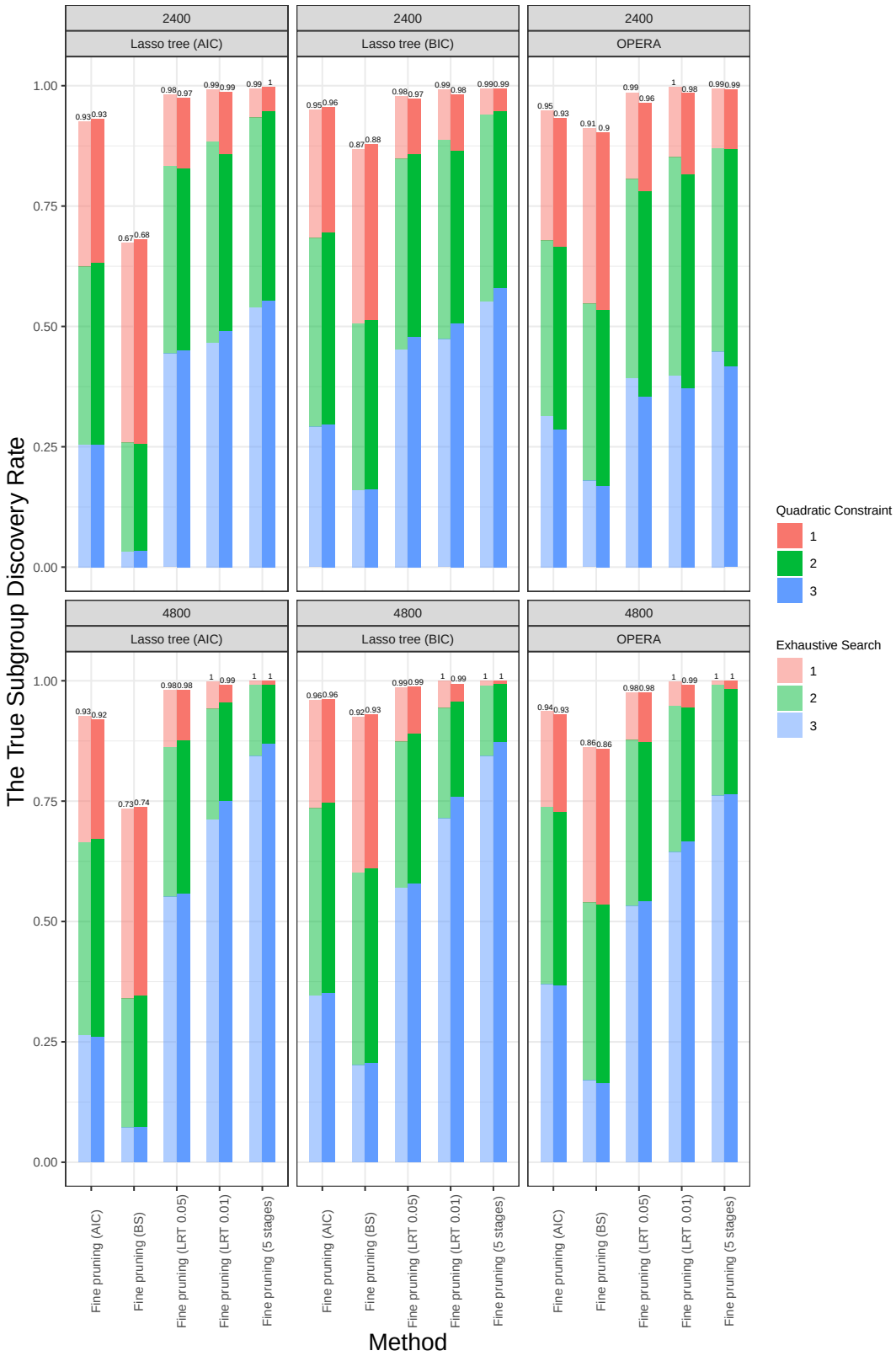


Figure 4.10: The true subgroup discovery rate using quadratic constraint and exhaustive search for a binary outcome with non-neighbouring staging patterns

Chapter 5

Real Data Analyses

OPERA can be applied to real datasets by leveraging multiple risk factors with either survival data or binary data. To illustrate its effectiveness, several public datasets are downloaded from the cBioPortal website. These datasets are used as examples for analyzing survival outcomes. Additionally, a dataset from [Lin+16] is selected to evaluate the performance of OPERA with binary outcomes. By utilizing these diverse datasets, OPERA's capability and versatility can be demonstrated across different types of data and research domains. One difference between previous simulation studies and the current real data analysis is that for simulation studies, there is no restriction on the number of patients in each stage. However, for real data analysis, we need to ensure that each stage has either no fewer than 30 patients or constitutes no fewer than 10% of the total patients. If the initial result does not satisfy this condition, it will activate coarse pruning to recursively merge the stages that do not meet the condition with adjacent stages until the condition is met.

5.1 A METABRIC Breast Cancer Dataset

The Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) study protocol received ethical approval from the University of Cambridge and British Columbia Cancer Research Centre. Funding for the METABRIC project was provided by Cancer Research UK, the British Columbia Cancer Foundation, and the Canadian Breast Cancer Foundation BC/Yukon[Per+16][Rue+19][Cur+12].

A comprehensive collection of over 2,000 clinically annotated primary fresh-frozen breast cancer specimens was assembled from tumor banks in the UK and Canada[Cur+12]. The treatment regimens were homogenous within clinically relevant groups, with almost all estrogen receptor (ER)-positive and/or

lymph node (LN)-negative patients not receiving chemotherapy, while ER-negative and LN-positive patients did receive chemotherapy. Furthermore, none of the HER2+ patients received trastuzumab. A discovery group of 997 tumors was initially analyzed, and an additional set of 995 tumors, for which complete data became available later, was used to assess the reproducibility of the integrative clusters.

This dataset comprises 2,509 patients, each corresponding to a single sample. The median overall survival time is 116.47 months, with 33.4% of samples censored and 21.0% missing.

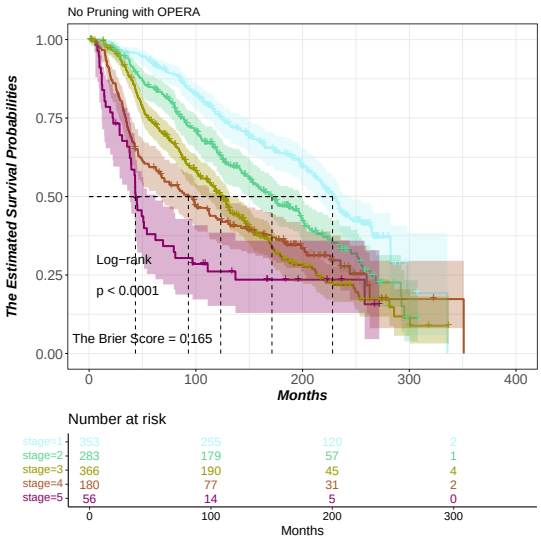
For cancer staging, three risk factors are considered: Pam50 subtype and Claudin-low subtype (including LumA, Normal, LumB, Her2, and Basal), Tumor Stage (1, 2, 3, 4), and Neoplasm Histologic Grade (1, 2, 3). Two different survival outcomes, overall survival and disease-free survival, are analyzed separately. Age is consistently adjusted as a continuous covariate throughout the staging process.

The cancer staging results based on overall survival using OPERA are presented in Figure 5.2a and 5.2b, while the corresponding Kaplan-Meier curves are depicted in Figure 5.1a and 5.1b. We used both coarse pruning and fine pruning with LRT ($\alpha = 0.01$), as both are computationally achievable. Well-separated curves indicate reliable staging results obtained from both OPERA without pruning and OPERA with fine pruning using LRT ($\alpha = 0.01$, exhaustive search). However, the latter approach prunes down one stage and leads to better separation among all the survival curves, as indicated by almost equally spaced median survival times. We only display the result from using coarse pruning due to its slightly superior performance and from using OPERA as the initial method due to its lower computational cost and comparable performance with lasso tree methods.

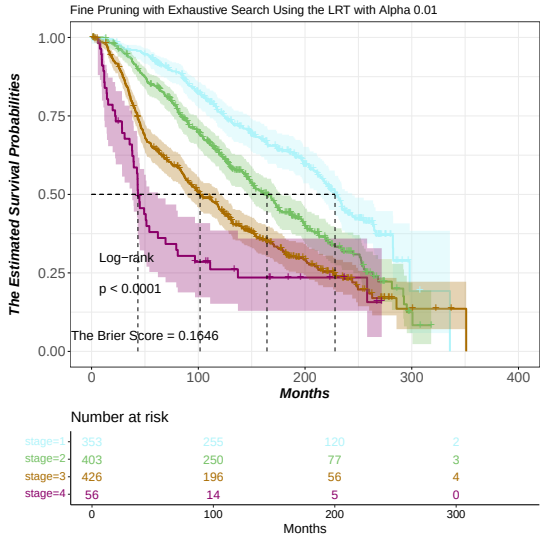
The cancer staging results based on disease-free survival are analyzed using the lasso tree with BIC, and the corresponding results are presented in Figure 5.4a and 5.4b, with the corresponding Kaplan-Meier curves depicted in Figure 5.3a and 5.3b. Since using OPERA only leads to 2 stages even without pruning, only the results obtained from the lasso tree are presented for disease-free survival. Although the lasso tree cannot handle non-neighboring patterns, it can still over-partition the combinations of risk factors, and pruning can help achieve more separated survival curves, as shown in Figure 5.4b. In this case, both coarse pruning with LRT ($\alpha = 0.01$) and fine pruning with LRT ($\alpha = 0.01$, exhaustive search) attain the same result.

5.2 A TCGA Prostate Cancer Dataset

Prostate Adenocarcinoma (TCGA, Firehose Legacy) originated from GDAC Firehose, previously known as TCGA Provisional. This dataset consists of 501 samples, encompassing 500 patients with one patient having two samples while all other patients has only one sample. The median overall survival time is

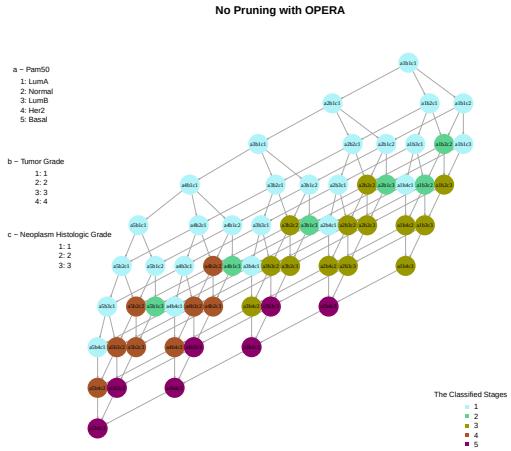


(a) No pruning (OPERA)

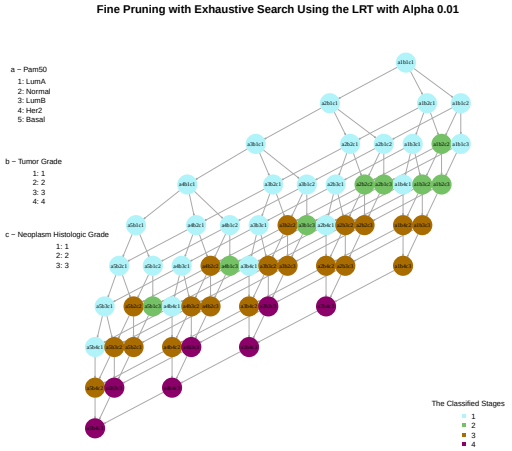


(b) Fine Pruning using LRT ($\alpha = 0.01$, Exhaustive Search)

Figure 5.1: The Kaplan-Meier curves for the overall survival probabilities across different stages obtained from OPERA for breast cancer patients

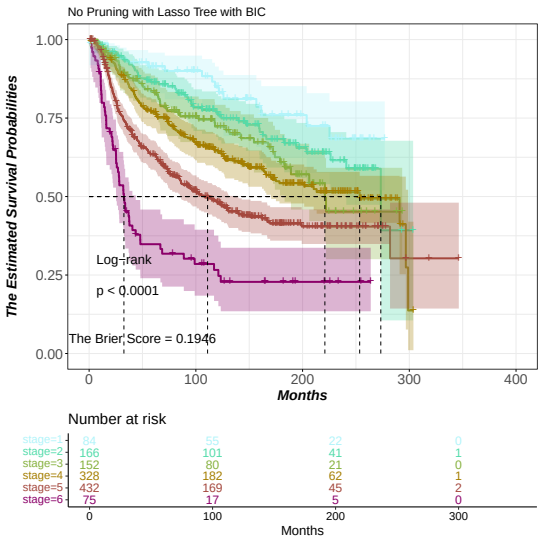


(a) No pruning (OPERA)

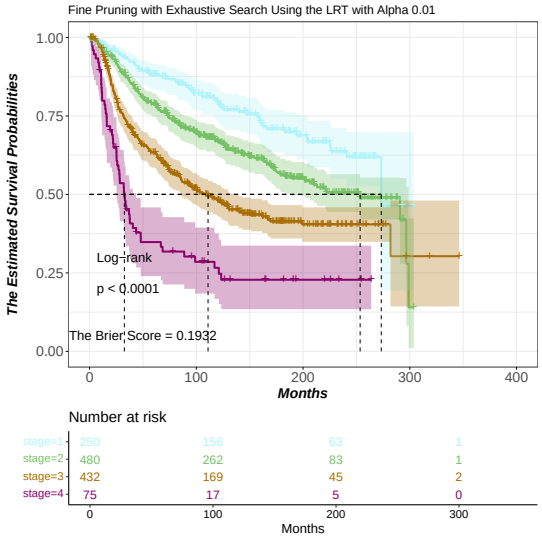


(b) Fine Pruning using LRT ($\alpha = 0.01$, Exhaustive Search)

Figure 5.2: The cancer staging results obtained from OPERA based on overall survival for breast cancer patients

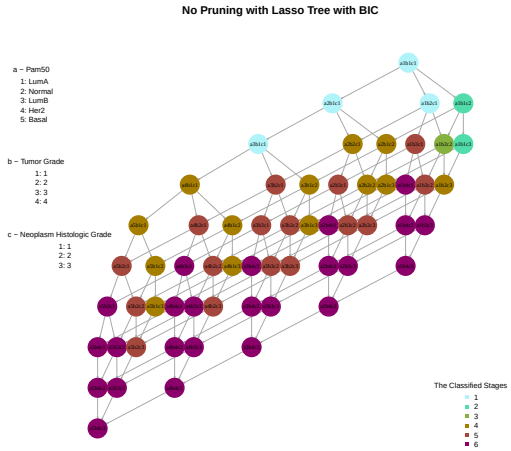


(a) No pruning (Lasso tree)

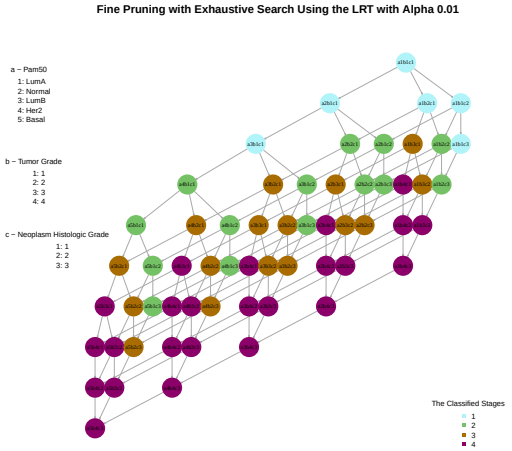


(b) Coarse or Fine Pruning using LRT ($\alpha = 0.01$, Exhaustive Search)

Figure 5.3: The Kaplan-Meier curves for the disease-free survival probabilities across different stages obtained from lasso tree for breast cancer patients

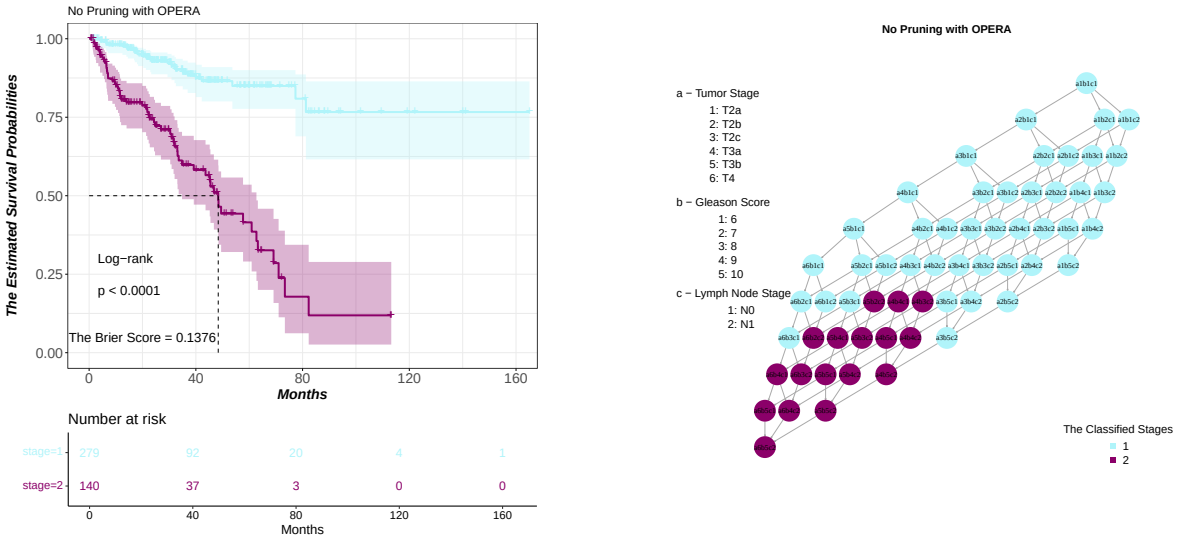


(a) No pruning (Lasso tree)



(b) Coarse or Fine Pruning using LRT ($\alpha = 0.01$, Exhaustive Search)

Figure 5.4: The cancer staging results obtained from lasso tree based on disease-free survival for breast cancer patients



(a) The Kaplan-Meier curves

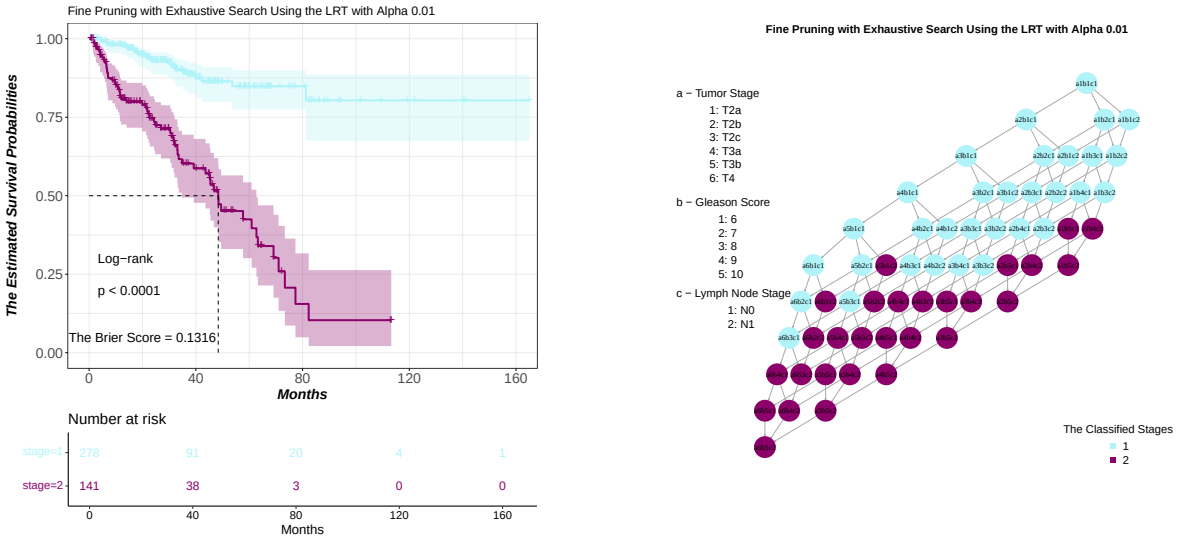
(b) The cancer staging results

Figure 5.5: The results for the disease-free survival probabilities across different stages for prostate cancer patients (1. No pruning using OPERA; 2. Lasso tree with BIC after using coarse pruning with LRT $\alpha = 0.01$)

30.52 months with 98.00% observations being censored.

Three risk factors are considered for cancer staging: Neoplasm Disease Lymph Node Stage (American Joint Committee on Cancer Code, ranging from T2a to T4), Tumor Stage (American Joint Committee on Cancer Code, including N0 and N1), and Radical Prostatectomy Gleason Score for Prostate Cancer (ranging from 6 to 10). Age is consistently adjusted as a continuous covariate throughout the staging process.

The cancer staging results based on disease-free survival are analyzed using both the lasso tree and OPERA, and the corresponding results are presented in Figure 5.5a, 5.5b, 5.6a, and 5.6b. After coarse pruning using LRT ($\alpha = 0.01$), both the lasso tree and OPERA yield the same two stages. After fine pruning using LRT ($\alpha = 0.01$, either using exhaustive search or the quadratic constraint), the lasso tree has a very similar staging result to OPERA. The Kaplan-Meier curves for the two stages demonstrate clear separation and exhibit a decent sample size in each stage, indicating reliable staging results. Due to the high censoring rate and relatively small sample size compared to the number of categories defined by the risk factors, a smaller number of stages is expected.



(a) The Kaplan-Meier curves

(b) The cancer staging results

Figure 5.6: The results for the disease-free survival probabilities across different stages for prostate cancer patients (Lasso tree with BIC after using fine pruning with LRT $\alpha = 0.01$)

5.3 A TCGA Lung Cancer Dataset

The study aimed to compare lung adenocarcinoma (ADC) and lung squamous cell carcinoma (SqCC) and identify new drivers of lung carcinogenesis. The exome sequences and copy number profiles of 660 ADC and 484 SqCC tumor/normal pairs were examined [Cam+16].

The dataset used for this analysis includes 660 lung ADC/normal paired exome sequences (including 274 previously unpublished cases, 227 cases from The Cancer Genome Atlas (TCGA) [Col+14], and 159 cases from the Imielinski cohort [Imi+12]). Additionally, 484 lung SqCC/normal paired exome sequences were studied, comprising 308 previously unpublished cases and 176 cases from TCGA [Net+12]. The goal was to compare the somatic profiles of lung ADC and SqCC and identify novel genetic alterations.

The Pan-Lung Cancer dataset consists of 1,144 patients, each with one corresponding sample. The median overall survival time is 8.10 months, with 61.45% of samples censored and 14.16% of samples missing.

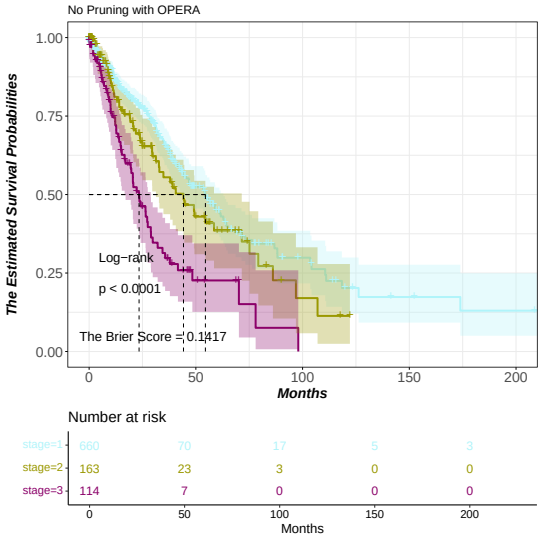
Two risk factors, Lymph Node Stage (N stage, including N0, N1, N2, and N3) and Tumor Stage (T stage, including T1, T2, T3, and T4), are used for cancer staging based on overall survival. Age is consistently adjusted as a continuous covariate throughout the staging process. All types of OPERA methods are applied, including OPERA with no pruning and OPERA with coarse or fine pruning using LRT. The cancer staging results are depicted in Figure 5.8a and 5.8b, while the corresponding Kaplan-

Meier curves are shown in Figure 5.7a and 5.7b. Well-separated curves indicate valid staging results, with OPERA using coarse pruning with LRT ($\alpha = 0.01$) achieving better separation than OPERA without pruning, as indicated by the median survival times. In this case, coarse pruning demonstrates the same performance as fine pruning using either exhaustive search or the quadratic constraint with LRT ($\alpha = 0.01$). Since using lasso tree only leads to 2 stages even without pruning, only the results obtained from OPERA are presented for overall survival.

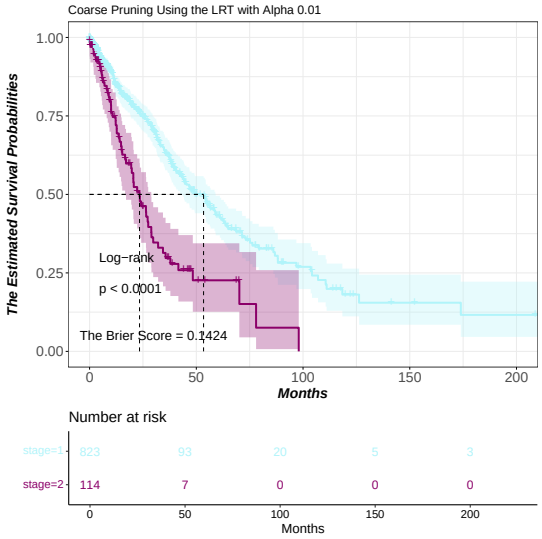
5.4 A TCGA Colorectal Cancer Dataset

The Colorectal Adenocarcinoma (TCGA, Firehose Legacy) dataset is derived from GDAC Firehose, previously known as TCGA Provisional. It comprises 640 samples from 636 patients. The median overall survival time is 21.75 months, with 78.46% of observations censored and 1.10% of observations missing.

For cancer staging, two risk factors are considered: Neoplasm Disease Lymph Node Stage (including N0, N1, and N2) and Tumor Stage (including T1, T2, T3, and T4) based on the American Joint Committee on Cancer Code. Age is consistently adjusted as a continuous covariate throughout the staging process. Two different survival outcomes, overall survival, and disease-free survival, are analyzed. The corresponding results are presented in Figure 5.10b and 5.10a for overall survival, and in Figure 5.12b and 5.12a for disease-free survival. The Kaplan-Meier curves are shown in Figure 5.9b and 5.9a for overall survival, and in Figure 5.11b and 5.11a for disease-free survival. The last stage remains consistent across different outcomes and approaches. All survival curves demonstrate clear separation after pruning, indicating robust staging results. For overall survival, fine pruning with LRT ($\alpha = 0.01$) using either quadratic constraint or exhaustive search leads to the same result with improved separation over OPERA without pruning. Coarse pruning with LRT ($\alpha = 0.01$) leads to only 2 stages, so the result is not displayed. For disease-free survival, coarse pruning and fine pruning using exhaustive search with LRT ($\alpha = 0.01$) lead to the same result as lasso tree with BIC without pruning.Fine pruning using quadratic constraint with LRT ($\alpha = 0.01$) demonstrates a very similar staging result, differing only in one node. Since using lasso tree with BIC only leads to 2 stages even without pruning, only the results obtained from OPERA are presented for overall survival.

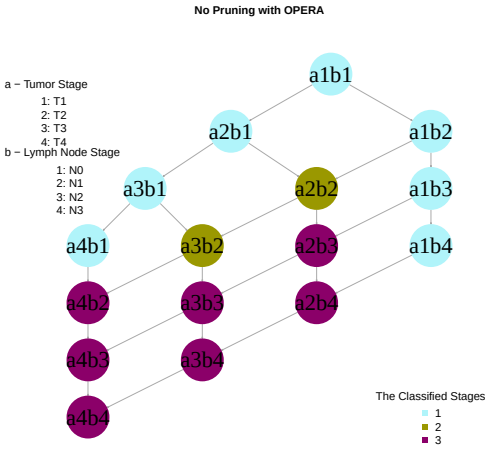


(a) No pruning (OPERA)

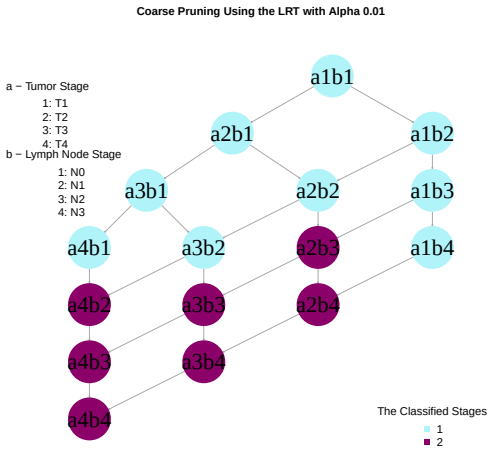


(b) Fine or Coarse Pruning using LRT ($\alpha = 0.01$)

Figure 5.7: The Kaplan-Meier curves for the overall survival probabilities across different stages obtained from OPERA for lung cancer patients

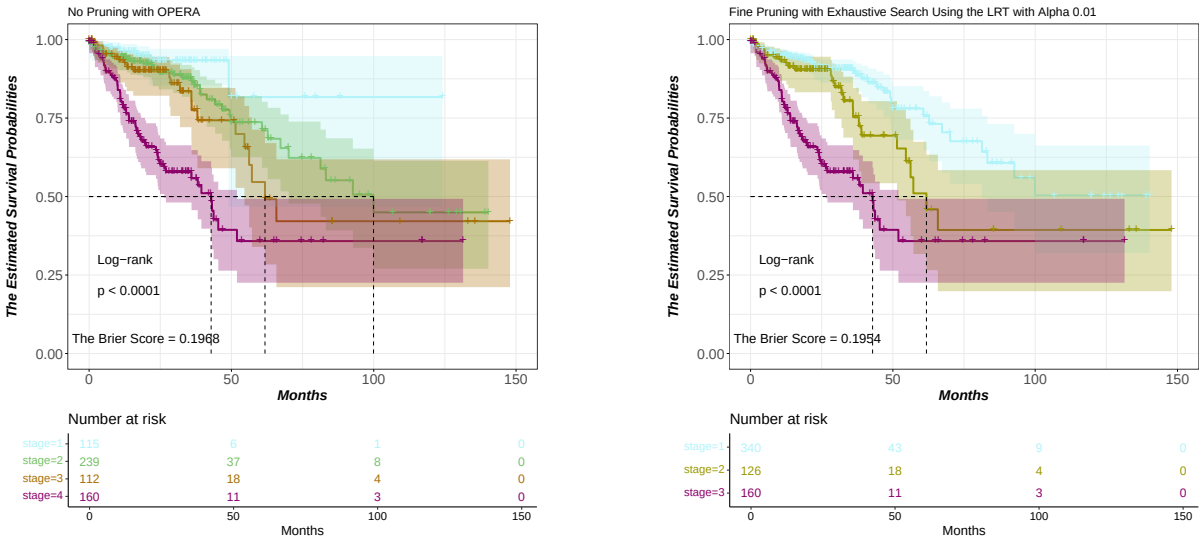


(a) No pruning (OPERA)



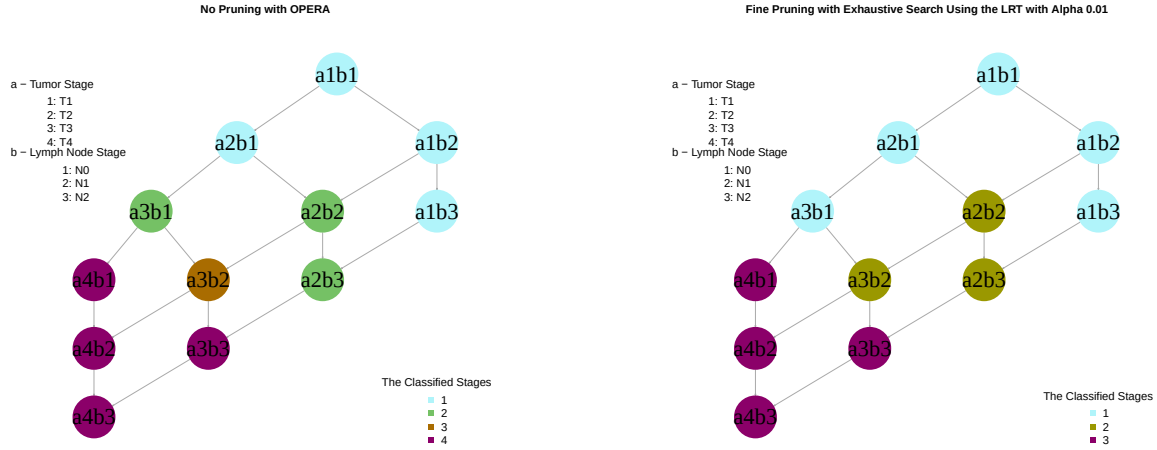
(b) Fine or Coarse Pruning using LRT ($\alpha = 0.01$)

Figure 5.8: The cancer staging results obtained from OPERA based on overall survival for lung cancer patients



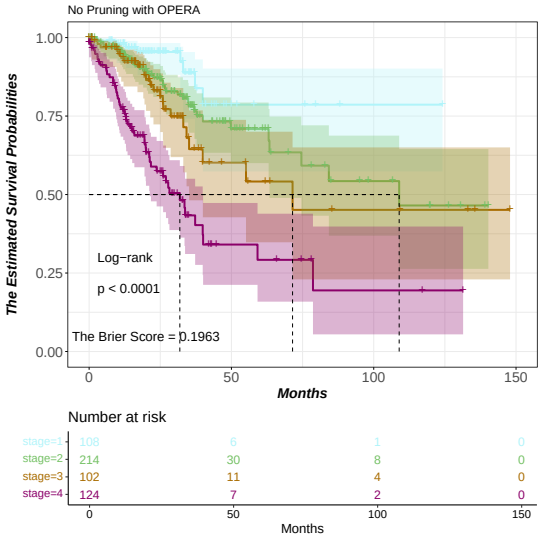
(a) No pruning (OPERA) (b) Fine Pruning using LRT ($\alpha = 0.01$)

Figure 5.9: The Kaplan-Meier curves for the overall survival probabilities across different stages obtained from OPERA for colorectal cancer patients

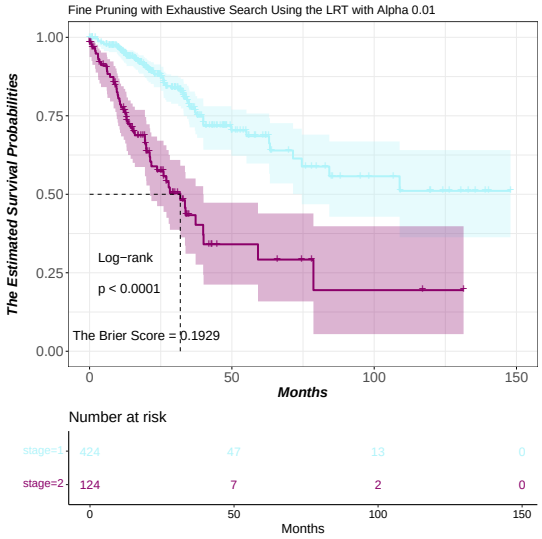


(a) No pruning (OPERA) (b) Fine Pruning using LRT ($\alpha = 0.01$)

Figure 5.10: The cancer staging results obtained from OPERA based on overall survival for colorectal cancer patients

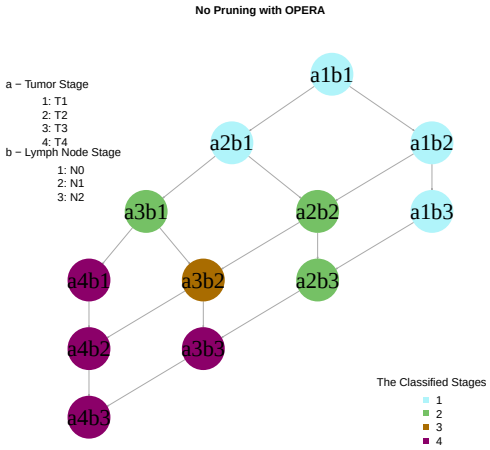


(a) No pruning (OPERA)

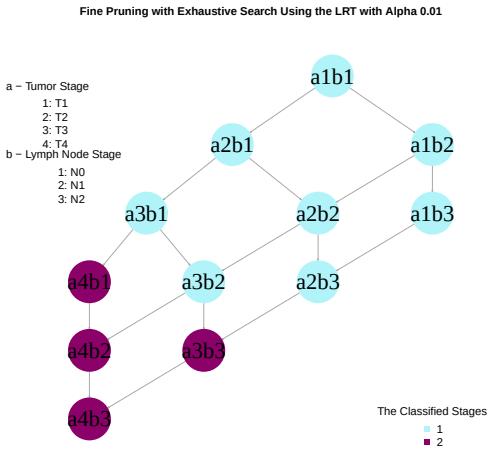


(b) Coarse or Fine Pruning using LRT ($\alpha = 0.01$, Exhaustive Search) or Lasso Tree with BIC (No Pruning)

Figure 5.11: The Kaplan-Meier curves for the disease-free survival probabilities across different stages obtained from OPERA for colorectal cancer patients



(a) No pruning (OPERA)



(b) Coarse or Fine Pruning using LRT ($\alpha = 0.01$, Exhaustive Search) or Lasso Tree with BIC (No Pruning)

Figure 5.12: The cancer staging results obtained from OPERA based on disease-free survival for colorectal cancer patients

5.5 A Breast Cancer Dataset from [Wan20]

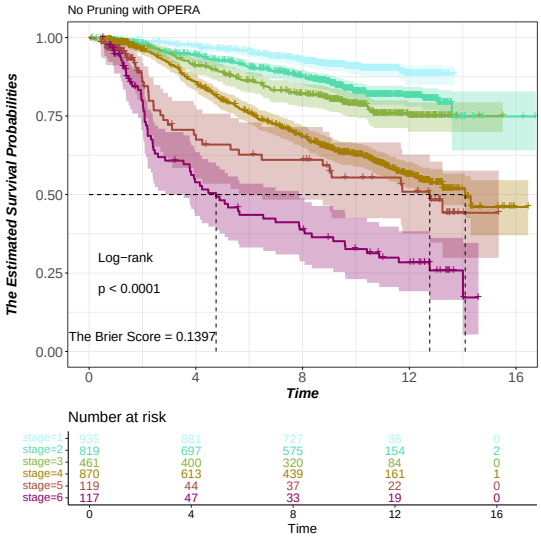
There is an additional breast cancer dataset available from [Wan20], comprising 3,321 patients. The mean follow-up time is 8.20 years, and the censoring rate is 79%. Two risk factors are considered: tumor grade (I, IIA, IIB, III) and ER/PR/HER2 status (ER/PR+/HER2+, ER/PR+/HER2-, ER/PR-/HER2+, ER/PR-/HER2-). Staging is based solely on overall survival outcome, while age and BMI (Body Mass Index) are adjusted as continuous variables during the staging process. The cancer staging results are presented in Figure 5.14a and 5.14b, accompanied by the corresponding Kaplan-Meier curves shown in Figure 5.13a and 5.13b. All survival curves demonstrate clear separation, with each stage having a substantial sample size, as indicated by the distinct median survival times between the last two stages. Coarse pruning and fine pruning using exhaustive search with LRT ($\alpha = 0.01$) lead to the same result and prune down two stages to demonstrate better separation. Lasso tree with BIC used as the initial method actually leads to the same result as OPERA.

5.6 Advanced Colorectal Neoplasia

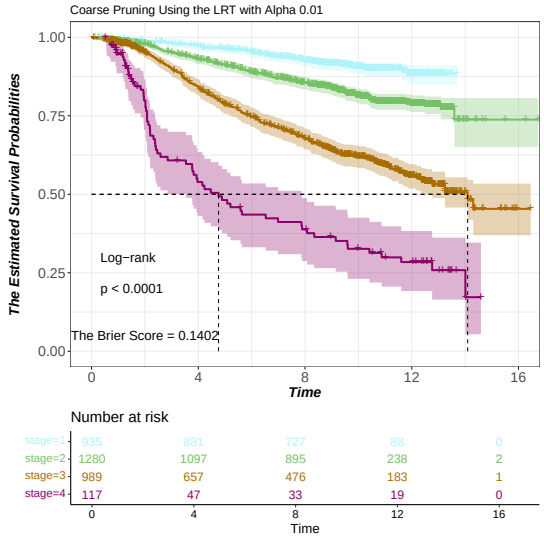
The study included individuals aged 50 to 80 years who underwent their first-time screening colonoscopy between December 2004 and September 2011[Lin+16]. The primary outcome of interest was advanced neoplasia, which encompassed a tubular adenoma greater than 1 cm, a polyp with villous histology or high-grade dysplasia, or colorectal cancer (CRC). Additional baseline characteristics can be found in Table 5.1. For further analysis, both female and male patients were considered.

To address the substantial number of risk factors, we apply logistic regression with the lasso method for variable selection, utilizing cross-validation and the R function *cv.glmnet*. We repeat this process 100 times, consistently identifying race, colonoscopy and polyp history, age, and cigarette smoking as the top four relevant risk factors for male patients. For female patients, the algorithm also constantly selects estrogen use as one of the risk factors. Since race lacks a specific ordering, we treat it as a covariate requiring adjustment when applying OPERA. Colonoscopy and polyp history feature four categories: screened with no polyps, screened with polyps, not screened, and unknown screening or polyps. The first two categories follow an ordered prognosis, while the others do not. Age ($\leq$ 65, 65+), cigarette smoking (0, $0 < - < 20$, 20+), estrogen use (regular user, non-user) are ordinal risk factors with total orderings.

To assess the performance of different pruning methods with various initial approaches, we use a 10-fold cross-validation. Each iteration involves both training and testing processes. During training, we

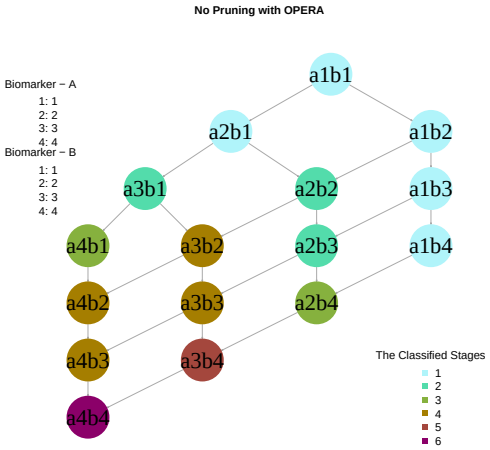


(a) No pruning (OPERA)

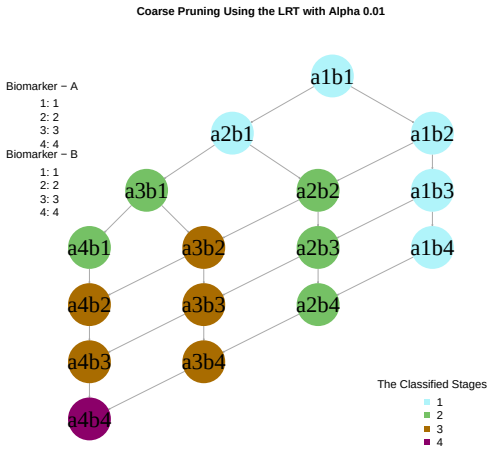


(b) Coarse or fine pruning using LRT ($\alpha = 0.01$, Exhaustive Search) with either OPERA or lasso tree as the initial result

Figure 5.13: The Kaplan-Meier curves for the overall survival probabilities across different stages obtained from OPERA for breast cancer patients



(a) No pruning (OPERA)



(b) Coarse or fine pruning using LRT ($\alpha = 0.01$, Exhaustive Search) with either OPERA or lasso tree as the initial result

Figure 5.14: The cancer staging results obtained from OPERA based on overall survival for breast cancer patients

utilize the training data to obtain stages. During testing, we fit a logistic regression model to the testing data, incorporating stages and race as covariates. We calculate the corresponding Area Under the ROC Curve (AUC) value for each testing dataset and compute the final Cross-Validated Area Under the ROC Curve (cvAUC) [LPL15], along with its 95

Table 5.2 displays the cvAUCs for male patients, while Table 5.3 displays those for female patients. Across all methods, for male patients, all the cvAUC values surpass 0.615, outperforming the results in [Lin+16]. Similarly, for female patients, all methods yield cvAUC values higher than 0.618, also outperforming the results in [Lin+16]. However, pruning does not enhance the cvAUC since the evaluation metric favors more stages. Lasso tree methods outperform OPERA as initial methods, consistent with Chapter 2's findings.

Table 5.1: Advanced colorectal neoplasia dataset baseline characteristics

	Male (N = 2162)	Female (N = 2304)	Overall (N = 4466)
Advanced colorectal neoplasia			
0	1,932, 89.36%	2,161, 93.79%	4,093, 91.65%
1	230, 10.64%	143, 6.21%	373, 8.35%
Race			
African American	65, 3.01%	77, 3.34%	142, 3.18%
Asian	30, 1.39%	43, 1.87%	73, 1.63%
Hispanic	18, 0.83%	17, 0.74%	35, 0.78%
White	2,049, 94.77%	2,167, 94.05%	4,216, 94.40%
Sex			
Female	0, 0.00%	2,304, 100.00%	2,304, 51.59%
Male	2,162, 100.00%	0, 0.00%	2,162, 48.41%
Cigarette smoking			
0	1,173, 54.26%	1,537, 66.71%	2,710, 60.68%
0<-<20	441, 20.40%	424, 18.40%	865, 19.37%
20+	548, 25.35%	343, 14.89%	891, 19.95%
Number of relatives w/ CRC			
0	1,555, 71.92%	1,430, 62.07%	2,985, 66.84%
1	433, 20.03%	575, 24.96%	1,008, 22.57%
2+	174, 8.05%	299, 12.98%	473, 10.59%
Age			
≤65	1,912, 88.44%	2,019, 87.63%	3,931, 88.02%
65+	250, 11.56%	285, 12.37%	535, 11.98%
BMI			
≤24.9	410, 18.96%	1,581, 68.62%	1,991, 44.58%
24.9<-≤29.9	975, 45.10%	723, 31.38%	1,698, 38.02%

29.9+	777, 35.94%	0, 0.00%	777, 17.40%
NSAID use			
Non user	1,013, 46.85%	1,215, 52.73%	2,228, 49.89%
Regular user	1,149, 53.15%	1,089, 47.27%	2,238, 50.11%
Colonoscopy and polyp history			
No screening	1,794, 82.98%	1,938, 84.11%	3,732, 83.56%
Screened and polyps	39, 1.80%	38, 1.65%	77, 1.72%
Screened no polyps	27, 1.25%	14, 0.61%	41, 0.92%
Unknown screen or polyps	302, 13.97%	314, 13.63%	616, 13.79%
Vegetable consumption			
<5	73, 3.38%	141, 6.12%	214, 4.79%
≥ 5	2,089, 96.62%	2,163, 93.88%	4,252, 95.21%
Vigorous activity			
0	545, 25.21%	840, 36.46%	1,385, 31.01%
$0<-\leq 2$	114, 5.27%	134, 5.82%	248, 5.55%
$2<-\leq 4$	161, 7.45%	203, 8.81%	364, 8.15%
4+	1,342, 62.07%	1,127, 48.91%	2,469, 55.28%
Estrogen			
Non user	-	1,351, 58.64%	1,351, 30.25%
Regular user	-	953, 41.36%	953, 21.34%

Table 5.2: The cvAUCs for male patients across different methods

Initial	pruning	Stopping	cvAUC	Lower	Upper
OPERA	Coarse Pruning	AIC	0.651	0.599	0.703
	Coarse Pruning	BS	0.652	0.600	0.704
	Coarse Pruning	LRT (0.01)	0.651	0.599	0.703
	Coarse Pruning	No Pruning	0.652	0.600	0.704
	Exhaustive Search	AIC	0.651	0.599	0.703
	Exhaustive Search	BS	0.652	0.600	0.704
	Exhaustive Search	LRT (0.01)	0.634	0.581	0.687
	Exhaustive Search	No Pruning	0.652	0.600	0.704
	Quadratic Constraint	AIC	0.651	0.599	0.703
	Quadratic Constraint	BS	0.652	0.600	0.704
	Quadratic Constraint	LRT (0.01)	0.636	0.581	0.691
	Quadratic Constraint	No Pruning	0.652	0.600	0.704
Lasso Tree (AIC)	Coarse Pruning	AIC	0.674	0.625	0.724
	Coarse Pruning	BS	0.677	0.631	0.723
	Coarse Pruning	LRT (0.01)	0.661	0.612	0.711
	Coarse Pruning	No Pruning	0.697	0.653	0.741
	Exhaustive Search	AIC	0.672	0.623	0.722
	Exhaustive Search	BS	0.695	0.651	0.740
	Exhaustive Search	LRT (0.01)	0.647	0.593	0.701
	Exhaustive Search	No Pruning	0.697	0.653	0.741
	Quadratic Constraint	AIC	0.676	0.628	0.724
	Quadratic Constraint	BS	0.696	0.651	0.740
	Quadratic Constraint	LRT (0.01)	0.640	0.588	0.692
	Quadratic Constraint	No Pruning	0.697	0.653	0.741
Lasso Tree (BIC)	Coarse Pruning	AIC	0.667	0.618	0.715
	Coarse Pruning	BS	0.688	0.641	0.736
	Coarse Pruning	LRT (0.01)	0.654	0.602	0.706
	Coarse Pruning	No Pruning	0.706	0.660	0.752
	Exhaustive Search	AIC	0.678	0.630	0.727

Table 5.2: The cvAUCs for male patients across different methods

Initial	pruning	Stopping	cvAUC	Lower	Upper
	Exhaustive Search	BS	0.684	0.637	0.731
	Exhaustive Search	LRT (0.01)	0.643	0.590	0.697
	Exhaustive Search	No Pruning	0.706	0.660	0.752
	Quadratic Constraint	AIC	0.673	0.625	0.721
	Quadratic Constraint	BS	0.683	0.635	0.730
	Quadratic Constraint	LRT (0.01)	0.649	0.597	0.701
	Quadratic Constraint	No Pruning	0.706	0.660	0.752

Table 5.3: The cvAUCs for female patients across different methods

Initial	pruning	Stopping	cvAUC	Lower	Upper
OPERA	Coarse Pruning	AIC	0.719	0.651	0.787
	Coarse Pruning	BS	0.729	0.665	0.794
	Coarse Pruning	LRT (0.01)	0.715	0.643	0.786
	Coarse Pruning	No Pruning	0.729	0.665	0.794
	Exhaustive Search	AIC	0.717	0.647	0.788
	Exhaustive Search	BS	0.728	0.661	0.794
	Exhaustive Search	LRT (0.01)	0.704	0.631	0.778
	Exhaustive Search	No Pruning	0.729	0.665	0.794
	Quadratic Constraint	AIC	0.717	0.647	0.788
	Quadratic Constraint	BS	0.728	0.661	0.794
	Quadratic Constraint	LRT (0.01)	0.702	0.629	0.776
	Quadratic Constraint	No Pruning	0.729	0.665	0.794
Lasso Tree (AIC)	Coarse Pruning	AIC	0.791	0.734	0.849
	Coarse Pruning	BS	0.814	0.766	0.862
	Coarse Pruning	LRT (0.01)	0.760	0.688	0.832
	Coarse Pruning	No Pruning	0.812	0.763	0.860
	Exhaustive Search	AIC	0.794	0.734	0.853
	Exhaustive Search	BS	0.815	0.761	0.868
	Exhaustive Search	LRT (0.01)	0.756	0.685	0.828
	Exhaustive Search	No Pruning	0.812	0.763	0.860
	Quadratic Constraint	AIC	0.795	0.736	0.855
	Quadratic Constraint	BS	0.814	0.765	0.863
	Quadratic Constraint	LRT (0.01)	0.759	0.689	0.829
	Quadratic Constraint	No Pruning	0.812	0.763	0.860
Lasso Tree (BIC)	Coarse Pruning	AIC	0.780	0.713	0.847
	Coarse Pruning	BS	0.791	0.727	0.855
	Coarse Pruning	LRT (0.01)	0.749	0.675	0.823
	Coarse Pruning	No Pruning	0.792	0.729	0.854
	Exhaustive Search	AIC	0.782	0.718	0.847

Table 5.3: The cvAUCs for female patients across different methods

Initial	pruning	Stopping	cvAUC	Lower	Upper
	Exhaustive Search	BS	0.792	0.730	0.855
	Exhaustive Search	LRT (0.01)	0.750	0.677	0.823
	Exhaustive Search	No Pruning	0.792	0.729	0.854
	Quadratic Constraint	AIC	0.782	0.718	0.846
	Quadratic Constraint	BS	0.792	0.729	0.854
	Quadratic Constraint	LRT (0.01)	0.755	0.687	0.824
	Quadratic Constraint	No Pruning	0.792	0.729	0.854

Chapter 6

Advanced Illness Patient Clustering

Throughout our development, OPERA has primarily focused on cancer staging problems, but its applicability extends beyond that. OPERA is designed to handle various risk stratification questions that involve ordinal risk factors, whether they possess total orderings or partial orderings. Additionally, OPERA has the capability to adjust for multiple covariates and model both survival outcomes and binary outcomes. By leveraging OPERA, patients with diverse characteristics can be effectively clustered into distinct risk levels, referred to as stages, which exhibit a total ordering aligned with their corresponding prognostic patterns.

An illustrative example of how OPERA, developed by the Health Innovation Program (HIP), extends beyond cancer staging problems is its application in clustering patients with advanced illnesses. By integrating advanced illness triggers (D) with age (A), palliative risk score (R), and count of frailty diagnoses (F), OPERA facilitates the following objectives: (a) enhancing the existing HIP-compiled list of potential triggers for end-of-life communication in advanced illness patients through the inclusion of additional risk factors, and (b) assisting in the identification of patients approaching the end of life, ensuring timely conversations about end-of-life care along the disease trajectory, and facilitating appropriate recommendations for palliative care to those in need. This comprehensive approach ensures that patients receive optimal support and care during their advanced illness journey.

Effective communication of care goals is a pivotal aspect of providing high-quality care to severely ill patients, as emphasized by Bernacki in their work on Communication About Serious Illness Care Goals[BB+14]. Engaging in end-of-life discussions not only contributes to an improved quality of life for patients, but also leads to a reduction in unnecessary medical interventions, lowered healthcare costs, and enhanced outcomes for families. To facilitate this process, HIP has devised an advanced illness trigger

hierarchy that organizes advanced illness triggers into mutually exclusive groupings in a hierarchical order. This following framework ensures a systematic approach to addressing the evolving needs of patients with advanced illness:

- 01: Cancer
  - Any Cancer Trigger
- 02: ESRD/CKD
  - Any ESRD/CKD Trigger
- 03: Multiple Organ Failure with Cognitive Impairment
  - 2+ Any Organ Failure Triggers (Heart Failure, Lung Failure, Liver Failure, or Brain Degeneration) with Cognitive Impairment Trigger
- 04: Multiple Organ Failure without Cognitive Impairment
  - 2+ Any Organ Failure Triggers (Heart Failure, Lung Failure, Liver Failure,or Brain Degeneration) without Cognitive Impairment Trigger
- 05: Organ Failure with Cognitive Impairment
  - Any Organ Failure Trigger(Heart Failure,Lung Failure, Liver Failure, or Brain Degeneration) with Cognitive Impairment Trigger
- 06: Other Advanced Illness with Cognitive Impairment
  - Any Other Advanced Illness Trigger with Cognitive Impairment Trigger
- 07: Organ Failure without Cognitive Impairment (Age 65+)
  - Any Organ Failure Trigger (Heart Failure, Lung Failure, Liver Failure, or Brain Degeneration) without Cognitive Impairment Trigger (Age 65+)
- 08: Other Advanced Illness without Cognitive Impairment (Age 65+)
  - Any Other Advanced Illness Trigger without Cognitive Impairment Trigger (Age 65+)
- 09: Cognitive Impairment (Age 65+)
  - Any Cognitive Impairment Trigger(Age 65+)

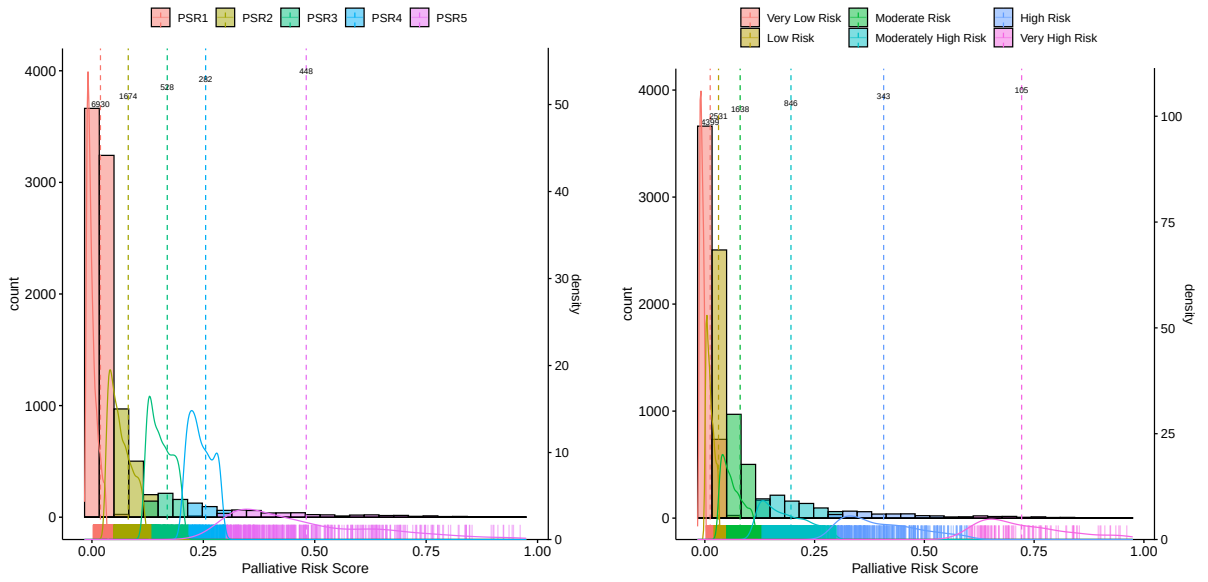
- 10: No Organ Failure or Other Advanced Illness Triggers
  - No Triggers

To enable effective clustering, a combination of several triggers was utilized, excluding the last trigger as it was deemed irrelevant for patients without advanced illnesses. Specifically, triggers 03 and 04 were consolidated into the category of multiple organ failure, while triggers 08 and 09 were merged into a single category. In addition, the Health Innovation Program (HIP) developed a palliative risk score categorized into six levels: very low risk, low risk, moderate risk, moderately high risk, high risk, and very high risk. This risk factor follows a total ordering and is ordinal in nature. Notably, patients with moderate risk and moderately high risk require further exploration to identify those who are in need of palliative care. Consequently, very low risk and low risk were combined as Palliative Risk Score 1 (PRS1), while moderate risk and moderately high risk were divided into PRS2 to PRS4 based on equal-sized palliative risk intervals. High risk and very high risk were consolidated as PRS5.

The distribution of palliative risk values across different groups is depicted in Figure 6.1, where the count is specified on the left y-axis and the density on the right y-axis. The sample sizes are listed at the top of the figure. In addition to the palliative risk score (PRS), age groups (≤ 80, $80-90$, >90), and the count of frailty diagnoses (0, 1, 2, 3, 4+) were utilized as risk factors. All three of these factors are ordinal and have a total ordering. However, the new advanced illness trigger groups, which only consist of seven categories, exhibit a partial ordering. This suggests a total ordering when cancer and end-stage renal disease (ESRD) patients are excluded, and patients with cognitive impairment (age 65+) or other advanced illnesses without cognitive impairment (age 65+) have better diagnoses than either cancer or ESRD patients.

During the clustering process using OPERA, two binary indicators, gender and urbanity, were adjusted. For the analysis, time-to-the-first-event analysis was conducted, focusing on four different survival outcomes: death, hospice utilization, hospitalization, and skilled nursing facility (SNF) utilization. The follow-up period for these outcomes was 12 months. It is important to note that each patient was limited to one episode and had at least one chronic condition. The study sample comprised 9,862 individuals who were Medicare accountable care organization (ACO) beneficiaries. Claims and electronic health record (EHR) data were utilized to create the advanced illness groupings. Additionally, 12-month baseline variables were included in the analysis.

The results, presented in Figure 6.2, demonstrate the utilization of the composite outcome comprising the four events. A comparison is made between the stages classified by OPERA and those developed by HIP. The Kaplan-Meier curves reveal enhanced differentiation among the latter three stages classified



(a) New palliative risk groups (b) Palliative risk groups developed by HIP

Figure 6.1: The distribution of palliative risk values across different groups

by OPERA, evident from the significant disparities in median survival time for each stage. Notably, the final stage, as classified by OPERA, exhibits a relatively smaller sample size, while stages 4 and 5 display larger sample sizes. This observation suggests that a greater number of patients may be recommended for palliative care intervention.

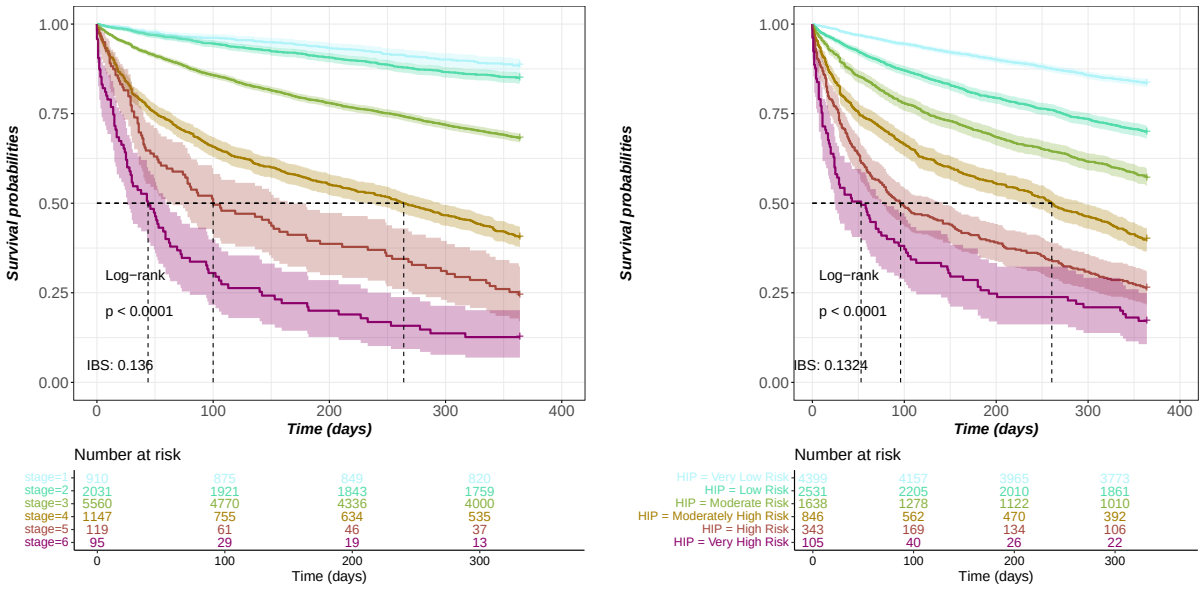
Given the distinct separation achieved using OPERA without pruning, no pruning is applied after the initial classification. Coarse pruning using different criteria yields the same outcome as the result obtained without pruning. With a considerable number of categories defined by risk factors, a parameter tuning value of $N = 10$ is employed in Algorithm 3. The Kaplan-Meier curves illustrating the survival probabilities for experiencing death or hospice are displayed in Figure 6.3. Notably, the early three stages exhibit greater separation when using the stages obtained by OPERA compared to the risk levels developed by HIP.

Furthermore, Figure 6.4 showcases the Kaplan-Meier curves illustrating the survival probabilities for the different risk levels based on palliative risk values (PSV), while maintaining the same sample sizes as the stages obtained from OPERA. These curves highlight the improved risk stratification achieved by incorporating multiple risk factors rather than solely relying on PRS. The survival curves across the stages obtained by OPERA, depicted in Figure 6.2a, show greater separation than those across the corresponding risk levels using only PRS. Similar trends are observed when examining the visualization of death and hospice in Figures 6.3a and 6.4b.

The findings are summarized in Table 6.1, where each row represents a specific risk level developed by HIP (ranging from R1 for very low risk to R6 for very high risk), and each column represents a specific level of frailty (ranging from F0 for no frailty diagnoses to F4 for at least four frailty diagnoses). The information for different stages is presented from stage 1 at the top to stage 6 at the bottom. In each cell, the corresponding age (ranging from A1 for no older than 80 to A3 for older than 90) and advanced illness triggers (ranging from D1 to D7, representing the new illness categories) are specified whenever such a combination exists within the corresponding stage.

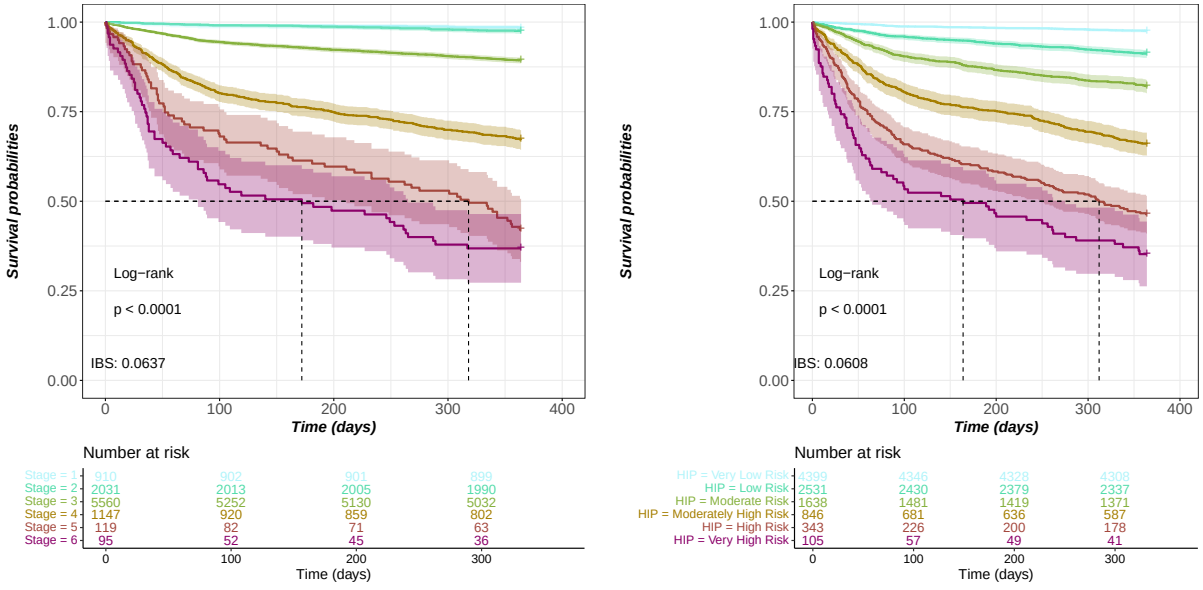
For example, stage 1 comprises patients with very low risk to moderately high risk, no older than 80, with cognitive impairment (age 65+) or other advanced illnesses without cognitive impairment (age 65+), and no frailty diagnoses. On the other hand, stage 2 consists of patients with very low risk to low risk, no older than 80, with cognitive impairment (age 65+) or other advanced illnesses without cognitive impairment (age 65+), and one to two frailty diagnoses, including cancer patients with no frailty.

It is important to note that moderately high risk cancer patients with no more than 3 frailty diagnoses are still categorized as stage 3, while moderately high risk patients with multiple organ failures and at least one frailty diagnosis are classified as stage 4. This indicates that some higher-risk patients with additional risk factors falling into lower-risk categories can be assigned to lower stages, while some relatively lower-risk patients with additional risk factors falling into higher-risk categories can be assigned to higher stages. Only high-risk patients and very high-risk patients are considered for stage 5 and stage 6. As the count of frailty diagnoses increases, the severity of diseases decreases or patients become younger, while controlling for risk level and stage.



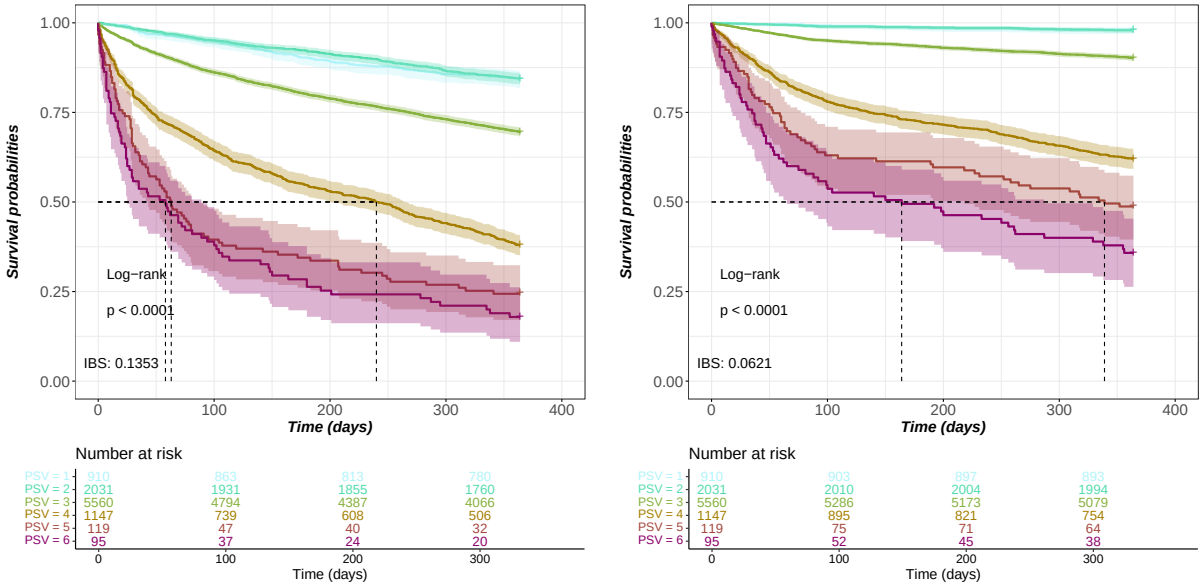
(a) Clusters or staged classified by OPERA (b) Palliative risk groups developed by HIP

Figure 6.2: The Kaplan-Meier curves illustrating various stages based on the composite outcome consisting of four distinct events



(a) Clusters or stages classified by OPERA (b) Palliative risk groups developed by HIP

Figure 6.3: The Kaplan-Meier curves depict the survival probabilities for experiencing death and hospice across the same stages obtained by OPERA. These stages are determined based on a composite outcome composed of four events



(a) Visualizing the composite outcome

(b) Visualizing only death and hospice

Figure 6.4: The Kaplan-Meier curves illustrate the survival probabilities for the different risk levels based on palliative risk values (PSV), while ensuring that the same sample sizes as the stages obtained from OPERA are maintained

Table 6.1: The summary of the results obtained by OPERA using the composite outcome of four events

Stage = 1	**F0**	**F1**	**F2**	**F3**	**F4**
R1 [n = 834]	(A1,D7) [n = 834]				
R2 [n = 65]	(A1,D7) [n = 65]				
R3 [n = 10]	(A1,D7) [n = 10]				
R4 [n = 1]	(A1,D7) [n = 1]				
Stage = 2	**F0**	**F1**	**F2**	**F3**	**F4**
R1 [n = 1578]	(A1,D1) [n = 25]	(A1,D7) [n = 964]	(A1,D7) [n = 589]		

R2 [n = 453]	(A1,D1) [n = 44]	(A1,D7) [n = 186]	(A1,D7) [n = 223]		
Stage = 3	**F0**	**F1**	**F2**	**F3**	**F4**
R1 [n = 1987]	(A1,D2,D3-D6),(A2,D2,D3-D7),(A3,D2,D6-D7) [n = 594]	(A1,D1,D2,D3-D6),(A2,D2,D3-D7),(A3,D3-D7) [n = 545]	(A1,D1,D2,D3-D6),(A2,D2,D3-D7),(A3,D5-D7) [n = 355]	(A1,D1,D2,D3-D7),(A2,D3-D7),(A3,D7) [n = 421]	(A1,D4-D7),(A2,D5-D7) [n = 72]
R2 [n = 2013]	(A1,D2,D3-D6),(A2,D1,D2,D3-D7),(A3,D6-D7) [n = 174]	(A1,D1,D2,D3-D6),(A2,D1,D2,D3-D7),(A3,D2,D3-D7) [n = 512]	(A1,D1,D2,D3-D6),(A2,D2,D3-D7),(A3,D2,D4-D7) [n = 559]	(A1-A2,D1,D2,D3-D7),(A3,D2,D3-D7) [n = 530]	(A1-A2,D2,D3-D7),(A3,D3-D7) [n = 238]
R3 [n = 1184]	(A1,D1,D2,D3-D6),(A2,D1,D2,D3-D7),(A3,D1,D6-D7) [n = 129]	(A1-A3,D1,D2,D4-D7) [n = 270]	(A1-A2,D1,D2,D4-D7),(A3,D2,D4-D7) [n = 372]	(A1-A2,D1,D2,D4-D7),(A3,D2,D7) [n = 348]	(A1-A2,D1,D2,D7),(A3,D2,D7) [n = 65]
R4 [n = 350]	(A1,D1,D3-D6),(A2,D1,D2,D3-D7),(A3,D1,D3) [n = 40]	(A1-A2,D1,D2,D4-D7),(A3,D1,D6-D7) [n = 71]	(A1-A3,D1,D2,D4-D7) [n = 141]	(A1-A2,D1,D4-D7),(A3,D1,D7) [n = 77]	(A1,D2,D1,D7),(A2-A3,D7) [n = 21]
R5 [n = 24]	(A1,D3) [n = 1]	(A1,D2),(A2,D4-D6) [n = 3]	(A1,D2,D5-D6),(A2,D2,D4-D6),(A3,D4-D5) [n = 16]	(A1,D6) [n = 1]	(A1-A2,D7) [n = 3]
R6 [n = 2]	(A1,D7) [n = 1]	(A1,D2) [n = 1]			
Stage = 4	**F0**	**F1**	**F2**	**F3**	**F4**
R3 [n = 444]		(A1-A3,D3) [n = 36]	(A1-A3,D3) [n = 55]	(A1-A2,D3),(A3,D3-D6) [n = 76]	(A1-A3,D3-D6) [n = 277]

R4 [n = 495]		(A1-A3,D3) [n = 17]	(A1-A3,D3) [n = 31]	(A1,D2,D3),(A2-A3,D2,D3-D6) [n = 162]	(A1-A3,D2[1],D3-D6) [n = 285]
R5 [n = 166]	(A1-A3,D1) [n = 8]	(A1-A3,D1) [n = 17]	(A1-A2,D1,D3),(A3,D3) [n = 23]	(A1,D1,D2,D3-D5),(A2-A3,D1,D2,D4-D6) [n = 70]	(A1-A2,D2,D5-D6),(A3,D2) [n = 48]
R6 [n = 42]		(A1,D1),(A2,D3) [n = 5]	(A1-A2,D1),(A3,D3) [n = 6]	(A1,D1,D2), (A2,D1,D4-D5), (A3,D2,D4) [n = 21]	(A1,D2,D5),(A2-A3,D2) [n = 10]
Stage = 5	**F0**	**F1**	**F2**	**F3**	**F4**
R5 [n = 91]					(A1,D3-D4),(A2,D4),(A3,D4-D5) [n = 91]
R6 [n = 28]					(A1,D3-D4),(A2,D4),(A3,D4-D5) [n = 28]
Stage = 6	**F0**	**F1**	**F2**	**F3**	**F4**
R5 [n = 62]				(A2-A3,D3) [n = 15]	(A1,D1),(A2,D1,D3), (A3,D3) [n = 47]
R6 [n = 33]				(A2-A3,D3) [n = 3]	(A1,D1),(A2-A3,D1,D3) [n = 30]

[1]The combination R4F4A1D2 is present in both stage 3 and stage 4, which is due to the utilization of PRS1 to PRS5 for clustering rather than the risk levels developed by HIP. In this particular case, R4 corresponds to different PRS levels, namely PRS2 and PRS3.

Chapter 7

An R Package - OPERAP

This chapter introduces the OPERAP R package, designed to facilitate the implementation of the lasso tree method and the OPEPA algorithm with pruning. The package offers flexibility in handling both survival outcomes and binary outcomes, allowing for adjustment of non-risk-factor covariates during model fitting. With OPERAP, cancer staging using ordinal risk factors can be performed, and the staging results can be conveniently saved to a specified filepath.

When dealing with time-to-event outcomes, OPERAP also enables the generation of Kaplan-Meier curves, providing a visual assessment of the effectiveness in separating different stages. The package supports various pruning methods, including coarse pruning and fine pruning. Fine pruning can be performed through exhaustive search or quadratic programming constraint.

Additionally, users can define different criteria to determine when to stop pruning. Options include AIC (Akaike Information Criterion), the (integrated) Brier score, the likelihood ratio test, or a predefined total number of stages. OPERAP offers a comprehensive toolkit for efficient and customizable cancer staging analysis.

To illustrate the implementation of cancer staging using our package, we utilize the METABRIC dataset discussed in Chapter 4. The dataset consists of 1,238 patients after removing missing values. Three risk factors, namely Pam50, tumor grade, and neoplasm histologic grade, are employed for cancer staging. Additionally, age at diagnosis can be incorporated as a non-risk-factor covariate. Within our package, the pivotal function for conducting cancer staging is **runOpera()**.

7.1 Installation

To install our R package, we will need to use the **devtools** package. If we haven't installed **devtools** yet, we can do so by running the following code:

```
install.packages('devtools')
```

Once we have **devtools** installed, we can proceed with installing our R package:

```
library(devtools)
devtools::install_github("yzliu1995/operap")
```

Make sure to load the package after installation to access the functions and features provided by our package.

```
library(operap)
```

7.2 Data

Once our package is loaded, we can proceed to explore the example dataset included within it. This particular dataset originates from the well-known METABRIC study, which we previously mentioned and discussed.

```
data("bric_bc_os")
```

7.3 Model Fitting

To facilitate cancer staging, there are two initial options available: utilizing OPERA or lasso tree. These methods can be employed to initialize the stages, followed by employing pruning with various stopping rules to refine and reduce the number of stages. In order to provide a comprehensive understanding of this bottom-up approach and how it can be implemented in our R package, we will illustrate the process step by step using our provided example dataset.

7.3.1 No Pruning

OPERA

runOpera() is a crucial function to utilize when implementing our R package for cancer staging. Below are explanations for the essential arguments of the function. To perform cancer staging without pruning using OPERA, the following list of arguments needs to be specified:

ncat: A vector that indicates the number of levels for each risk factor. In our example dataset, we have three risk factors, with 5 levels in the first one, 4 levels in the second one, and 3 levels in the third one. Therefore, we specify it as **c(5L, 4L, 3L)**.

dat: The dataset in the data.frame format. In this case, we specify it as **bric_bc_os**, which contains the example dataset.

plt: A flag to determine whether to plot the figures for visualizing the risk categories. By default, it is set to **TRUE**.

filepath: The path where all the results and figures should be saved. In our example, it is set as **"./results/no pruning/"**. If the path does not exist, it will be automatically created. Note that there must be a slash at the end of the file path string.

riskVariables: A vector containing the variable names of the risk factors used for staging. In our example, the three variables are **"Pam50 + Claudin - low subtype"**, **"Tumor Stage"**, and **"Neoplasm Histologic Grade"**.

riskFactors: A list specifying the levels for each risk factor in ascending order of prognosis. In our example, it is represented as **list(c("LumA", "Normal", "LumB", "Her2", "Basal"), c(1, 2, 3, 4), c(1, 2, 3))**. If a risk factor does not have a total ordering, we need to set the argument **ifE = TRUE** and provide the partial orderings as edges in the network using the **edges** argument.

riskNames: A vector containing the names of the risk factors, used to rename them in the resulting figures. In our example, they are **"a - Pam50"**, **"b - Tumor Grade"**, and **"c - Neoplasm Histologic Grade"**.

TimeN: The variable name for survival times. In our example, it is **"Overall Survival (Months)"**.

yN: The variable name for a binary outcome. In this case, it is NULL since we are using a survival outcome.

cenN: The variable name for censoring times. It should be numeric, with zeros indicating censored observations and ones indicating events. In our example, it is **"censoringStatus"**.

covN: The variable name(s) for covariates. In our example, it is **"Age at Diagnosis"**.

withCov: Specifies whether any covariates need adjustment.

type: The type of outcome, either "surv" for survival outcome or "bin" for binary outcome.

minObs: The minimum number of patients required in each stage. In our example, any number of patients is allowed.

legend_size: The size of the legend in the network of risk categories figure.

x_pos: The horizontal position of the legend in the network of risk categories figure.

yratio: A ratio proportional to the number of levels in each factor, controlling the vertical position of the legend in the network of risk categories figure.

yaxis_min: The minimum value of the y-axis for the Kaplan-Meier curves.

xpos_bs: The horizontal position of the label for the Brier score on the x-axis scale.

x_label: The label of the x-axis for the Kaplan-Meier curves.

Additionally, there are several other parameters for figure configuration that can be passed to **graphics::title()** and **igraph::plot.igraph()** through our function.

The crucial argument is to set **useOPERA = TRUE** since we specifically intend to employ OPERA as the staging method. Below is an example of the code and its corresponding output:

```
r_bric_bc_cp_opera_os <- runOpera(ncat = c(5L, 4L, 3L),
                                  dat = bric_bc_os,
                                  plt = T,
                                  filepath = "./results/no pruning/",
                                  riskVariables = c("Pam50 + Claudin-
    low subtype", "Tumor Stage", "Neoplasm Histologic Grade"),
                                  riskFactors = list(c("LumA", "Normal"
    , "LumB", "Her2", "Basal" ), c(1, 2, 3, 4), c(1, 2, 3)),
                                  riskNames = c("a - Pam50", "b - Tumor
     Grade", "c - Neoplasm Histologic Grade"),
                                  useOPERA = T,
                                  TimeN = "Overall Survival (Months)",
                                  cenN = "censoringStatus",
                                  yN = NULL,
                                  type = "surv",
                                  covN = "Age at Diagnosis",
                                  withCov = T,
                                  minObs = 0,
```

```
                                    # parameters for figures
                                    cex.main = 1,
                                    xpos_bs = 50,
                                    x_label = "Months",
                                    edge.width = 0.5,
                                    edge.arrow.size = 0.1,
                                    vertex.size = 11,
                                    vertex.frame.color  = NA,
                                    vertex.label.cex = 0.5,
                                    vertex.label.color =  "black",
                                    legend_size = 0.7,
                                    x_pos = -1.5,
                                    vertex.color = "lightblue")
```

```
## [1] "Find the stage 1"
##     mu a1b1c1 a2b1c1 a1b2c1 a1b1c2 a3b1c1 a2b2c1 a2b1c2 a1b3c1
   a1b2c2 a1b1c3
## -0.830  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
   0.186  0.000
## a4b1c1 a3b2c1 a3b1c2 a2b3c1 a2b2c2 a2b1c3 a1b4c1 a1b3c2 a1b2c3
   a5b1c1 a4b2c1
##  0.000  0.000  0.000  0.000  0.305  0.146  0.000  0.197  0.412
   0.000  0.000
## a4b1c2 a3b3c1 a3b2c2 a3b1c3 a2b4c1 a2b3c2 a2b2c3 a1b4c2 a1b3c3
   a5b2c1 a5b1c2
##  0.000  0.000  0.544  0.146  0.000  0.591  0.544  0.253  0.613
   0.000  0.000
## a4b3c1 a4b2c2 a4b1c3 a3b4c1 a3b3c2 a3b2c3 a2b4c2 a2b3c3 a1b4c3
   a5b3c1 a5b2c2
##  0.000  0.741  0.146  0.000  0.591  0.544  0.591  0.613  0.613
   0.000  0.741
## a5b1c3 a4b4c1 a4b3c2 a4b2c3 a3b4c2 a3b3c3 a2b4c3 a5b4c1 a5b3c2
   a5b2c3 a4b4c2
```

```
##  0.146  0.000  0.741  0.741  0.591  1.139  2.794  0.000  0.741
   0.794  0.741
## a4b3c3 a3b4c3 a5b4c2 a5b3c3 a4b4c3 a5b4c3
##  1.175  2.794  0.741  1.175  2.794  2.794
## [1] "The best lambda = 0.5672"
## Age_at_Diagnosis
##           0.035
## [1] "Find the stage 2"
##     mu a1b2c2 a2b2c2 a2b1c3 a1b3c2 a1b2c3 a3b2c2 a3b1c3 a2b3c2
   a2b2c3 a1b4c2
##  0.472  0.000  0.083  0.000  0.000  0.174  0.310  0.000  0.442
   0.310  0.203
## a1b3c3 a4b2c2 a4b1c3 a3b3c2 a3b2c3 a2b4c2 a2b3c3 a1b4c3 a5b2c2
   a5b1c3 a4b3c2
##  0.523  0.527  0.000  0.442  0.310  0.442  0.523  0.523  0.527
   0.000  0.527
## a4b2c3 a3b4c2 a3b3c3 a2b4c3 a5b3c2 a5b2c3 a4b4c2 a4b3c3 a3b4c3
   a5b4c2 a5b3c3
##  0.527  0.473  0.921  2.909  0.527  0.555  0.527  0.955  2.914
   0.527  0.955
## a4b4c3 a5b4c3     S1
##  2.914  2.914 -0.362
## [1] "The best lambda = 0.2838"
## Age_at_Diagnosis
##           0.035
## [1] "Find the stage 3"
##     mu a2b2c2 a1b2c3 a3b2c2 a2b3c2 a2b2c3 a1b4c2 a1b3c3 a4b2c2
   a3b3c2 a3b2c3
##  0.196  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.154
   0.000  0.000
## a2b4c2 a2b3c3 a1b4c3 a5b2c2 a4b3c2 a4b2c3 a3b4c2 a3b3c3 a2b4c3
   a5b3c2 a5b2c3
##  0.000  0.000  0.000  0.154  0.154  0.154  0.000  0.561  2.015
```

```
   0.154  0.216
## a4b4c2 a4b3c3 a3b4c3 a5b4c2 a5b3c3 a4b4c3 a5b4c3     S1     S2
##  0.154  0.586  2.015  0.154  0.586  2.015  2.015 -0.691 -0.379
## [1] "The best lambda = 0.6952"
## Age_at_Diagnosis
##           0.035
## [1] "Find the stage 4"
##     mu a4b2c2 a5b2c2 a4b3c2 a4b2c3 a3b3c3 a2b4c3 a5b3c2 a5b2c3
   a4b4c2 a4b3c3
##  0.285  0.000  0.000  0.000  0.000  0.353  2.382  0.000  0.000
   0.000  0.394
## a3b4c3 a5b4c2 a5b3c3 a4b4c3 a5b4c3     S1     S2     S3
##  2.436  0.000  0.394  2.436  2.436 -0.932 -0.622 -0.286
## [1] "The best lambda = 0.2152"
## Age_at_Diagnosis
##           0.036
## [1] "Find the stage 5"
##     mu a3b3c3 a2b4c3 a4b3c3 a3b4c3 a5b3c3 a4b4c3 a5b4c3     S1
   S2     S3
##  0.536  0.000  2.043  0.000  2.175  0.000  2.175  2.175 -1.338
   -1.028 -0.691
##     S4
## -0.418
## [1] "The best lambda = 0.0982"
## Age_at_Diagnosis
##           0.036
## [1] "Find the stage 6"
##     mu a2b4c3 a3b4c3 a4b4c3 a5b4c3     S1     S2     S3     S4
   S5
## -0.796  0.000  0.000  0.000  0.000 -3.541 -3.230 -2.894 -2.621
   -2.213
## [1] "The best lambda = 0.125"
## Age_at_Diagnosis
```

```
##            0.036
## [1] "The initial staging result:"
## a1b1c1 a2b1c1 a1b2c1 a1b1c2 a3b1c1 a2b2c1 a2b1c2 a1b3c1 a1b2c2
   a1b1c3 a4b1c1
##      1      1      1      1      1      1      1      1      2
   1      1
## a3b2c1 a3b1c2 a2b3c1 a2b2c2 a2b1c3 a1b4c1 a1b3c2 a1b2c3 a5b1c1
   a4b2c1 a4b1c2
##      1      1      1      3      2      1      2      3      1
   1      1
## a3b3c1 a3b2c2 a3b1c3 a2b4c1 a2b3c2 a2b2c3 a1b4c2 a1b3c3 a5b2c1
   a5b1c2 a4b3c1
##      1      3      2      1      3      3      3      3      1
   1      1
## a4b2c2 a4b1c3 a3b4c1 a3b3c2 a3b2c3 a2b4c2 a2b3c3 a1b4c3 a5b3c1
   a5b2c2 a5b1c3
##      4      2      1      3      3      3      3      3      1
   4      2
## a4b4c1 a4b3c2 a4b2c3 a3b4c2 a3b3c3 a2b4c3 a5b4c1 a5b3c2 a5b2c3
   a4b4c2 a4b3c3
##      1      4      4      3      5      6      1      4      4
   4      5
## a3b4c3 a5b4c2 a5b3c3 a4b4c3 a5b4c3
##      6      4      5      6      6
```

Upon examining the files within our result directory, we can see the following files:

```
No Pruning with OPERA_K-M.pdf
No Pruning with OPERA_tree-like structure.pdf
No Pruning with OPERA_tree_-like structure.html
No Pruning with OPERA_tree_-like structure_files
tree-like structure.html
tree-like structure.pdf
tree-like structure_files
```

The first file, titled *No Pruning with OPERA_K-M.pdf*, contains the Kaplan-Meier curves representing different stages. The second file, named *No Pruning with OPERA_tree-like structure.pdf*, displays the network of risk factors in a tree-like structure, with the stages labeled. Additionally, the third file, *No Pruning with OPERA_tree-like structure.html*, is an interactive HTML file that enables visualization of the network of risk factors with labeled stages. The files *tree-like structure.html* and *tree-like structure.pdf* represent the network structure before the stages are labeled. The two additional folders are utilized for generating the HTML files.

The visualization of the first three files is depicted in Figures 7.1, 7.2, and 7.3. If users prefer each stage to have a minimum number of patients (e.g., 30), they can modify the argument **minObs** as **minObs = 30**. This adjustment enables coarse pruning to select the optimal staging result, ensuring that each stage consists of at least 30 patients.

In the example above, we assume that each risk factor has a total ordering. However, our package can also deal with some risk factor(s) with only partial ordering. The key is to set the correct edges representing the partial ordering.

For example, if *Pam50* is a partially ordered risk factor with the partial ordering as $LumB \leq Her2 \leq Basal$, and no ordering for either *LumA* or *Normal* with respect to other levels. The network of all three risk factors will be composed of one sub-network associated with levels from $LumB \leq Her2 \leq Basal$, and the other two sub-networks associated with *LumA* and *Normal* respectively. Below is the code that shows how to define the edges in this scenario. To perform the cancer staging, users only need to add **ifE = TRUE** and **edges = edges**. Note that there is no need to change the value passed to **riskFactors** as we still want *a1* to represent *LumA*, *a2* to represent *Normal*, and so on.

```
# One sub-network associated with levels from ‘LumB <= Her2 <= Basal‘
sub_1 <- edgesHasse(ncat = c(3L, 4L, 3L), e =  c())
for(i in 1:3){
  sub_1 <- gsub(paste0("a", i), paste0("e", 2+i), sub_1)
}
sub_1 <- gsub("e", "a", sub_1)

# The other two sub-networks associated with ‘LumA‘ and ‘Normal‘
sub_2 <- edgesHasse(ncat = c(1L, 4L, 3L), e =  c())
sub_3 <-gsub("a1", "a2", sub_2)
edges <- c(sub_2, sub_3, sub_1)
```

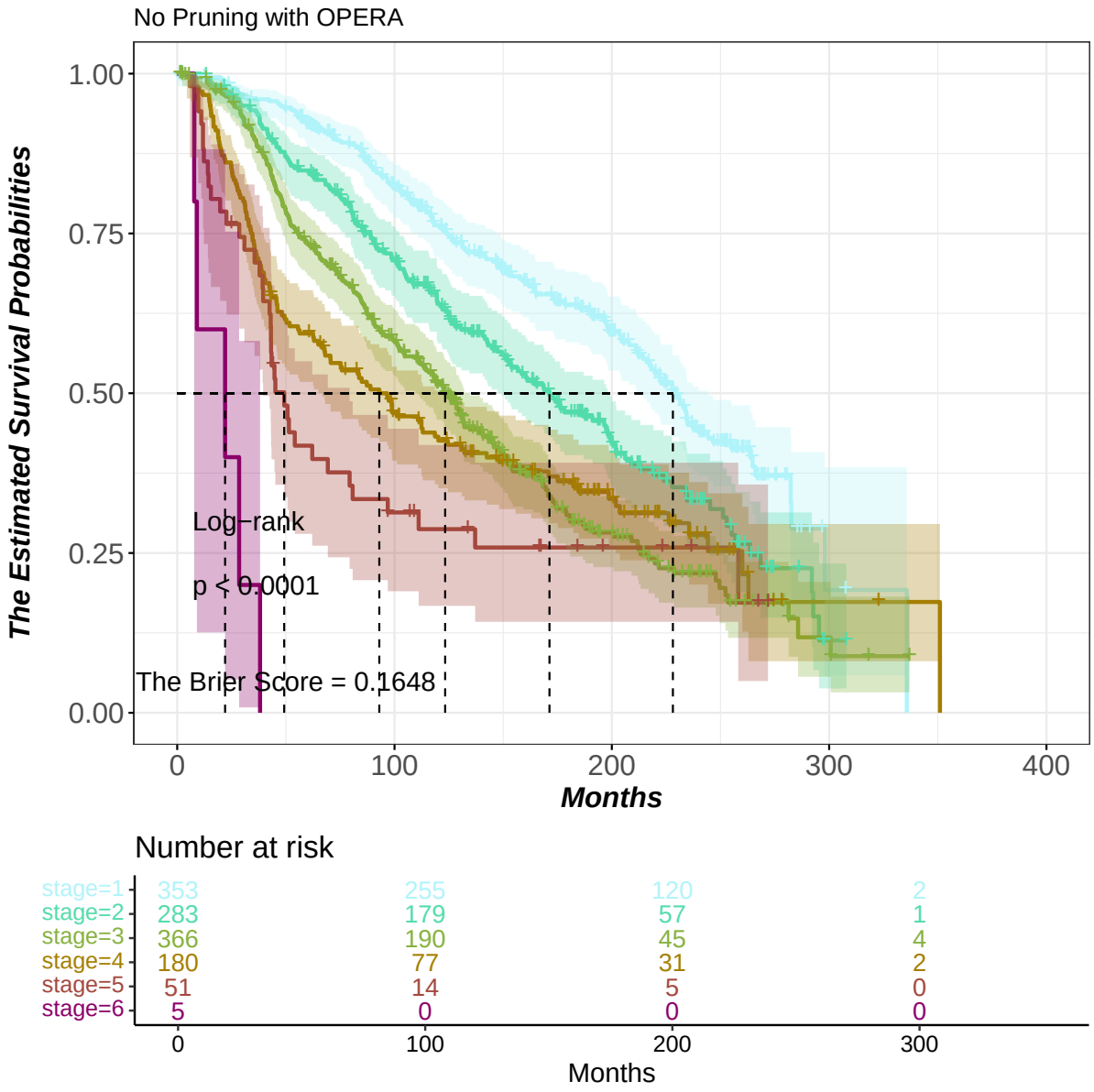


Figure 7.1: The Kaplan-Meier curves illustrating the stages obtained using OPERA, without any pruning or restriction on the number of patients in each stage.

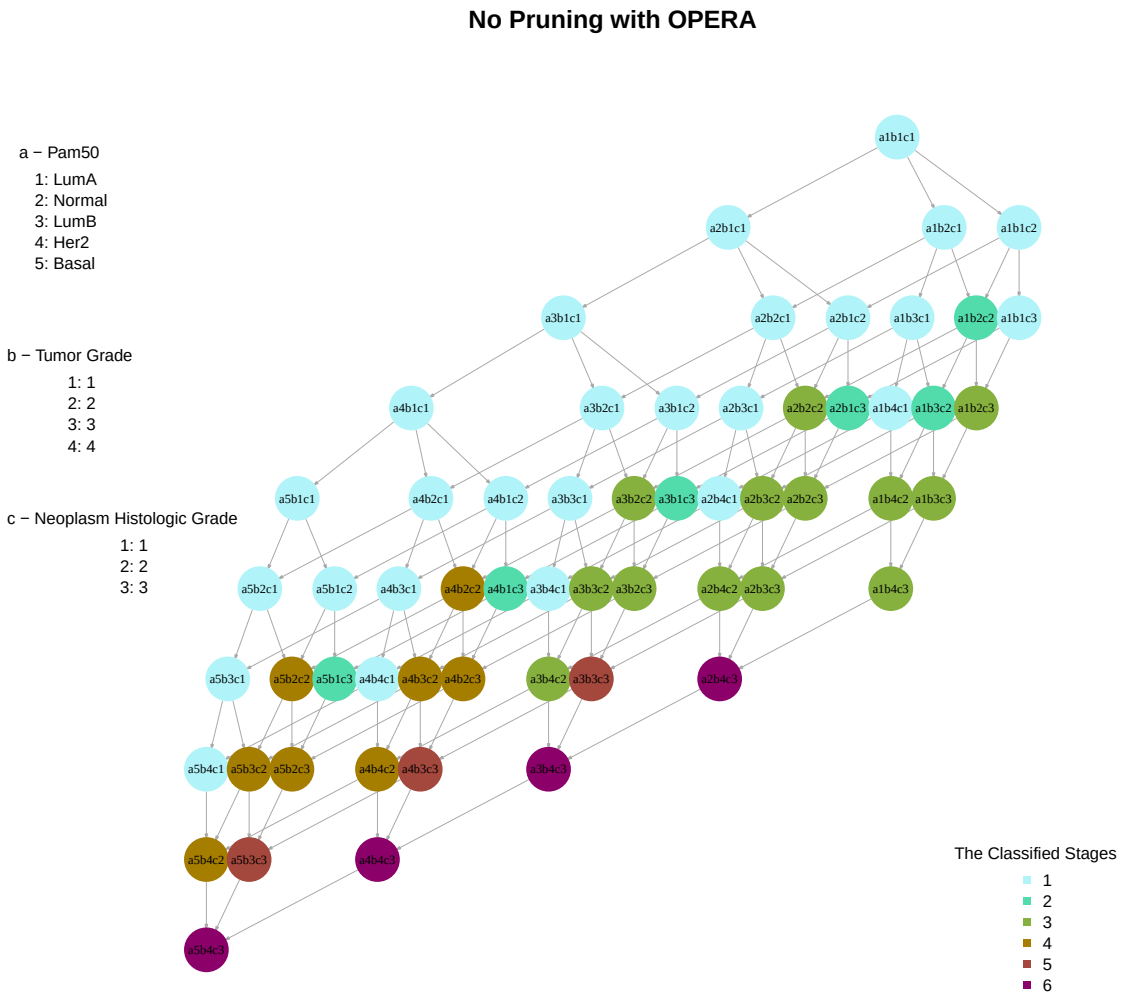


Figure 7.2: The tree-like network of risk factors labeled with stages obtained using OPERA, without any pruning or restriction on the number of patients in each stage.

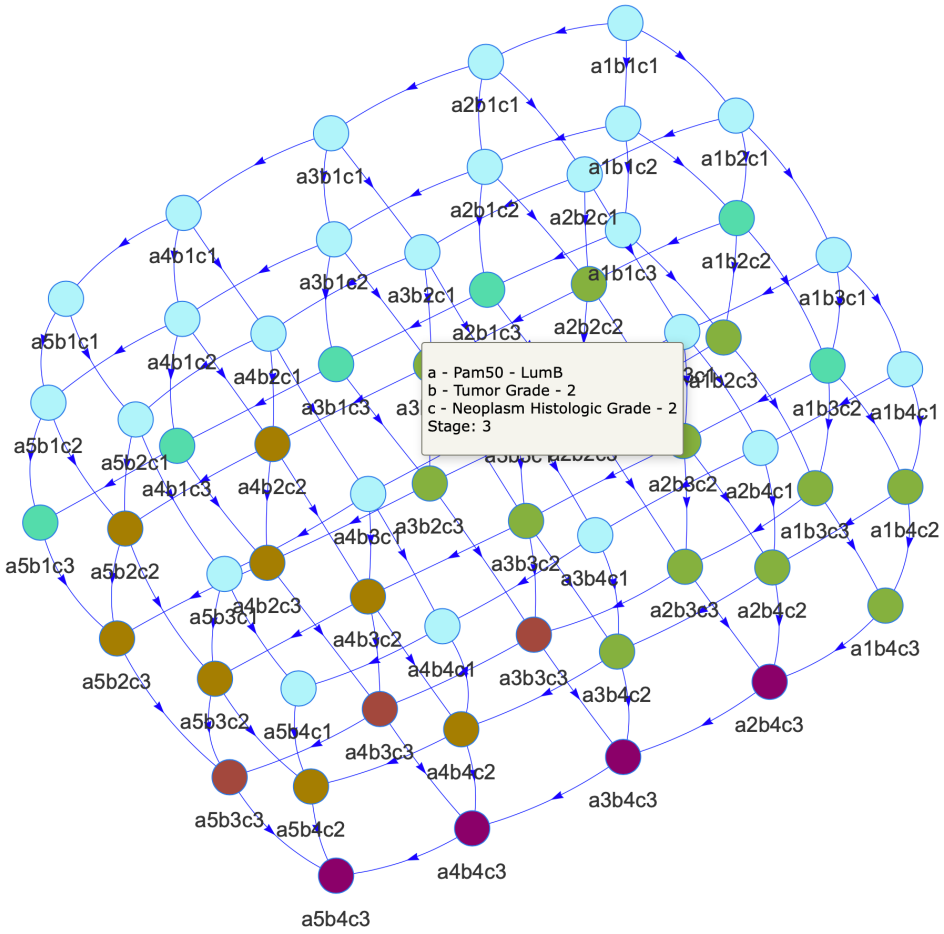


Figure 7.3: The interactive network of risk factors labeled with stages obtained using OPERA, without any pruning or restriction on the number of patients in each stage.

Lasso Tree

The crucial argument to note is to set **useLassoT = TRUE** instead of **useOPERA = TRUE**, as we have chosen to utilize the lasso tree as the staging method. With this change, all other steps and procedures can be followed in a similar manner to when we used OPERA. Moreover, the tolerance accuracy for convergence can be adjusted by setting the value of **eps_lasso**, which controls the speed at which the algorithm converges. Additionally, during parameter tuning, we have the option to use the AIC instead of the BIC by specifying **useBIC = FALSE**.

7.3.2 Pruning

To enable pruning, we need to specify the argument **usePruning = TRUE**. There are four different stopping rules available: the likelihood ratio test with a pre-specified Type I error rate α, the AIC, the (integrated) Brier score, or a predefined total number of stages. In our package, if we choose the likelihood ratio test, we can set the argument **useLRT = TRUE**, and the corresponding α (e.g., 0.01) can be specified using **threshold = 0.01**. Using the AIC, the (integrated) Brier score, or a predefined total number of stages (e.g., 5) can be achieved by setting **useAIC = TRUE**, **useIbs = TRUE**, or **prefix_stage = 5**, respectively.

We recommend using the likelihood ratio test with an $\alpha = 0.01$ when the final number of stages is unknown, as it has shown superior performance in simulation studies. Alternatively, if the number of stages is known in advance, using a predefined number is a suitable option. There are three fundamental pruning methods available: coarse pruning, fine pruning using exhaustive search, and fine pruning using quadratic constraint. In our package, coarse pruning can be implemented by setting **coarse_pruning = TRUE**, fine pruning using exhaustive search by setting **fine_pruning = TRUE**, and fine pruning using quadratic constraint by setting **fine_pruning_quad = TRUE**. It is important to choose only one of the three fundamental pruning methods, depending on the available computational resources.

We recommend fine pruning using exhaustive search due to its superior performance in simulation studies. However, it can be computationally expensive when the total number of risk categories is high. Fine pruning using quadratic constraint and coarse pruning offer decent accuracy comparable to fine pruning using exhaustive search but with less computational burden. Therefore, we suggest choosing either fine pruning using exhaustive search, fine pruning using quadratic constraint, or coarse pruning based on the computational feasibility in the specific scenario.

To demonstrate the implementation of pruning using our package, we will use the likelihood ratio test with an $\alpha = 0.01$ and apply fine pruning using exhaustive search. The same steps can be followed

for other methods by specifying the corresponding arguments, as discussed earlier. The code snippet and the resulting output are provided below:

```
r_bric_bc_fp_opera_os <- runOpera(ncat = c(5L, 4L, 3L),
                                  dat = bric_bc_os,
                                  plt = T,
                                  filepath = "./results/pruning/LRT/",
                                  riskVariables = c("Pam50 + Claudin-low subtype", "Tumor Stage", "Neoplasm Histologic Grade"),
                                  riskFactors = list(c("LumA", "Normal", "LumB", "Her2", "Basal" ), c(1, 2, 3, 4), c(1, 2, 3)),
                                  riskNames = c("a - Pam50", "b - Tumor Grade", "c - Neoplasm Histologic Grade"),
                                  useOPERA = T,
                                  # adds the arguments for pruning
                                  usePruning = T,
                                  fine_pruning = T,
                                  useLRT = T,
                                  threshold = 0.01,
                                  # Other arguments
                                  TimeN = "Overall Survival (Months)",
                                  cenN = "censoringStatus",
                                  yN = NULL,
                                  type = "surv",
                                  covN = "Age at Diagnosis",
                                  withCov = T,
                                  minObs = 0,
                                  # parameters for figures
                                  cex.main = 1,
                                  xpos_bs = 50,
                                  x_label = "Months",
                                  edge.width = 0.5,
                                  edge.arrow.size = 0.1,
```

```
                                    vertex.size = 11,
                                    vertex.frame.color  = NA,
                                    vertex.label.cex = 0.5,
                                    vertex.label.color =  "black",
                                    legend_size = 0.7,
                                    x_pos = -1.5,
                                    vertex.color = "lightblue")
```

```
## [1] "Find the stage 1"
##     mu a1b1c1 a2b1c1 a1b2c1 a1b1c2 a3b1c1 a2b2c1 a2b1c2 a1b3c1
   a1b2c2 a1b1c3
## -0.830  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
   0.186  0.000
## a4b1c1 a3b2c1 a3b1c2 a2b3c1 a2b2c2 a2b1c3 a1b4c1 a1b3c2 a1b2c3
   a5b1c1 a4b2c1
##  0.000  0.000  0.000  0.000  0.305  0.146  0.000  0.197  0.412
   0.000  0.000
## a4b1c2 a3b3c1 a3b2c2 a3b1c3 a2b4c1 a2b3c2 a2b2c3 a1b4c2 a1b3c3
   a5b2c1 a5b1c2
##  0.000  0.000  0.544  0.146  0.000  0.591  0.544  0.253  0.613
   0.000  0.000
## a4b3c1 a4b2c2 a4b1c3 a3b4c1 a3b3c2 a3b2c3 a2b4c2 a2b3c3 a1b4c3
   a5b3c1 a5b2c2
##  0.000  0.741  0.146  0.000  0.591  0.544  0.591  0.613  0.613
   0.000  0.741
## a5b1c3 a4b4c1 a4b3c2 a4b2c3 a3b4c2 a3b3c3 a2b4c3 a5b4c1 a5b3c2
   a5b2c3 a4b4c2
##  0.146  0.000  0.741  0.741  0.591  1.139  2.794  0.000  0.741
   0.794  0.741
## a4b3c3 a3b4c3 a5b4c2 a5b3c3 a4b4c3 a5b4c3
##  1.175  2.794  0.741  1.175  2.794  2.794
## [1] "The best lambda = 0.5672"
## Age_at_Diagnosis
```

```
##            0.035
## [1] "Find the stage 2"
##     mu a1b2c2 a2b2c2 a2b1c3 a1b3c2 a1b2c3 a3b2c2 a3b1c3 a2b3c2
   a2b2c3 a1b4c2
##  0.472  0.000  0.083  0.000  0.000  0.174  0.310  0.000  0.442
   0.310  0.203
## a1b3c3 a4b2c2 a4b1c3 a3b3c2 a3b2c3 a2b4c2 a2b3c3 a1b4c3 a5b2c2
   a5b1c3 a4b3c2
##  0.523  0.527  0.000  0.442  0.310  0.442  0.523  0.523  0.527
   0.000  0.527
## a4b2c3 a3b4c2 a3b3c3 a2b4c3 a5b3c2 a5b2c3 a4b4c2 a4b3c3 a3b4c3
   a5b4c2 a5b3c3
##  0.527  0.473  0.921  2.909  0.527  0.555  0.527  0.955  2.914
   0.527  0.955
## a4b4c3 a5b4c3     S1
##  2.914  2.914 -0.362
## [1] "The best lambda = 0.2838"
## Age_at_Diagnosis
##            0.035
## [1] "Find the stage 3"
##     mu a2b2c2 a1b2c3 a3b2c2 a2b3c2 a2b2c3 a1b4c2 a1b3c3 a4b2c2
   a3b3c2 a3b2c3
##  0.196  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.154
   0.000  0.000
## a2b4c2 a2b3c3 a1b4c3 a5b2c2 a4b3c2 a4b2c3 a3b4c2 a3b3c3 a2b4c3
   a5b3c2 a5b2c3
##  0.000  0.000  0.000  0.154  0.154  0.154  0.000  0.561  2.015
   0.154  0.216
## a4b4c2 a4b3c3 a3b4c3 a5b4c2 a5b3c3 a4b4c3 a5b4c3     S1     S2
##  0.154  0.586  2.015  0.154  0.586  2.015  2.015 -0.691 -0.379
## [1] "The best lambda = 0.6952"
## Age_at_Diagnosis
##            0.035
```

```
## [1] "Find the stage 4"
##     mu a4b2c2 a5b2c2 a4b3c2 a4b2c3 a3b3c3 a2b4c3 a5b3c2 a5b2c3
   a4b4c2 a4b3c3
##  0.285  0.000  0.000  0.000  0.000  0.353  2.382  0.000  0.000
   0.000  0.394
## a3b4c3 a5b4c2 a5b3c3 a4b4c3 a5b4c3     S1     S2     S3
##  2.436  0.000  0.394  2.436  2.436 -0.932 -0.622 -0.286
## [1] "The best lambda = 0.2152"
## Age_at_Diagnosis
##           0.036
## [1] "Find the stage 5"
##     mu a3b3c3 a2b4c3 a4b3c3 a3b4c3 a5b3c3 a4b4c3 a5b4c3     S1
   S2     S3
##  0.536  0.000  2.043  0.000  2.175  0.000  2.175  2.175 -1.338
   -1.028 -0.691
##     S4
## -0.418
## [1] "The best lambda = 0.0982"
## Age_at_Diagnosis
##           0.036
## [1] "Find the stage 6"
##     mu a2b4c3 a3b4c3 a4b4c3 a5b4c3     S1     S2     S3     S4
   S5
## -0.796  0.000  0.000  0.000  0.000 -3.541 -3.230 -2.894 -2.621
   -2.213
## [1] "The best lambda = 0.125"
## Age_at_Diagnosis
##           0.036
## [1] "The initial staging result:"
## a1b1c1 a2b1c1 a1b2c1 a1b1c2 a3b1c1 a2b2c1 a2b1c2 a1b3c1 a1b2c2
   a1b1c3 a4b1c1
##      1      1      1      1      1      1      1      1      2
   1      1
```

```
## a3b2c1 a3b1c2 a2b3c1 a2b2c2 a2b1c3 a1b4c1 a1b3c2 a1b2c3 a5b1c1
   a4b2c1 a4b1c2
##      1      1      1      3      2      1      2      3      1
   1      1
## a3b3c1 a3b2c2 a3b1c3 a2b4c1 a2b3c2 a2b2c3 a1b4c2 a1b3c3 a5b2c1
   a5b1c2 a4b3c1
##      1      3      2      1      3      3      3      3      1
   1      1
## a4b2c2 a4b1c3 a3b4c1 a3b3c2 a3b2c3 a2b4c2 a2b3c3 a1b4c3 a5b3c1
   a5b2c2 a5b1c3
##      4      2      1      3      3      3      3      3      1
   4      2
## a4b4c1 a4b3c2 a4b2c3 a3b4c2 a3b3c3 a2b4c3 a5b4c1 a5b3c2 a5b2c3
   a4b4c2 a4b3c3
##      1      4      4      3      5      6      1      4      4
   4      5
## a3b4c3 a5b4c2 a5b3c3 a4b4c3 a5b4c3
##      6      4      5      6      6
## [1] "The fine pruning has started using the exhaustive enumeration."
## [1] "The staging result after pruning down one stage with the
   largest likelihood is"
## a1b1c1 a2b1c1 a1b2c1 a1b1c2 a3b1c1 a2b2c1 a2b1c2 a1b3c1 a1b2c2
   a1b1c3 a4b1c1
##      1      1      1      1      1      1      1      1      2
   1      1
## a3b2c1 a3b1c2 a2b3c1 a2b2c2 a2b1c3 a1b4c1 a1b3c2 a1b2c3 a5b1c1
   a4b2c1 a4b1c2
##      1      1      1      2      2      1      2      2      1
   1      1
## a3b3c1 a3b2c2 a3b1c3 a2b4c1 a2b3c2 a2b2c3 a1b4c2 a1b3c3 a5b2c1
   a5b1c2 a4b3c1
##      1      3      2      1      3      3      3      3      1
   1      1
```

```
## a4b2c2 a4b1c3 a3b4c1 a3b3c2 a3b2c3 a2b4c2 a2b3c3 a1b4c3 a5b3c1
   a5b2c2 a5b1c3
##      3      2      1      3      3      3      3      3      1
   3      2
## a4b4c1 a4b3c2 a4b2c3 a3b4c2 a3b3c3 a2b4c3 a5b4c1 a5b3c2 a5b2c3
   a4b4c2 a4b3c3
##      1      3      3      3      4      5      1      3      3
   3      4
## a3b4c3 a5b4c2 a5b3c3 a4b4c3 a5b4c3
##      5      3      4      5      5
## [1] "The p-value equals 0.7048"
## [1] "The staging result after pruning down one stage with the
   largest likelihood is"
## a1b1c1 a2b1c1 a1b2c1 a1b1c2 a3b1c1 a2b2c1 a2b1c2 a1b3c1 a1b2c2
   a1b1c3 a4b1c1
##      1      1      1      1      1      1      1      1      2
   1      1
## a3b2c1 a3b1c2 a2b3c1 a2b2c2 a2b1c3 a1b4c1 a1b3c2 a1b2c3 a5b1c1
   a4b2c1 a4b1c2
##      1      1      1      2      2      1      2      2      1
   1      1
## a3b3c1 a3b2c2 a3b1c3 a2b4c1 a2b3c2 a2b2c3 a1b4c2 a1b3c3 a5b2c1
   a5b1c2 a4b3c1
##      1      3      2      1      3      3      3      3      1
   1      1
## a4b2c2 a4b1c3 a3b4c1 a3b3c2 a3b2c3 a2b4c2 a2b3c3 a1b4c3 a5b3c1
   a5b2c2 a5b1c3
##      3      2      1      3      3      3      3      3      1
   3      2
## a4b4c1 a4b3c2 a4b2c3 a3b4c2 a3b3c3 a2b4c3 a5b4c1 a5b3c2 a5b2c3
   a4b4c2 a4b3c3
##      1      3      3      3      3      4      1      3      3
   3      3
```

```
## a3b4c3 a5b4c2 a5b3c3 a4b4c3 a5b4c3
##      4      3      3      4      4
## [1] "The p-value equals 0.0045"
```

After inspecting the files in our result directory, the following files are observed:

```
Fine Pruning with Exhaustive Search Using the LRT with Alpha 0.01_K-M.
   pdf
Fine Pruning with Exhaustive Search Using the LRT with Alpha 0.01_tree-
   like structure.pdf
Fine Pruning with Exhaustive Search Using the LRT with Alpha 0.01_tree_
   -like structure.html
Fine Pruning with Exhaustive Search Using the LRT with Alpha 0.01_tree_
   -like structure_files
No Pruning with OPERA_K-M.pdf
No Pruning with OPERA_tree-like structure.pdf
No Pruning with OPERA_tree_-like structure.html
No Pruning with OPERA_tree_-like structure_files
tree-like structure.html
tree-like structure.pdf
tree-like structure_files
```

It is important to note that we have the same set of files as obtained when using OPERA without pruning. However, these files represent the results after pruning has been applied. Since the initial staging is always performed before pruning, we can observe both sets of files within our result directory. The first three files depicting the results after pruning are visualized in Figure 7.4, 7.5 and 7.6.

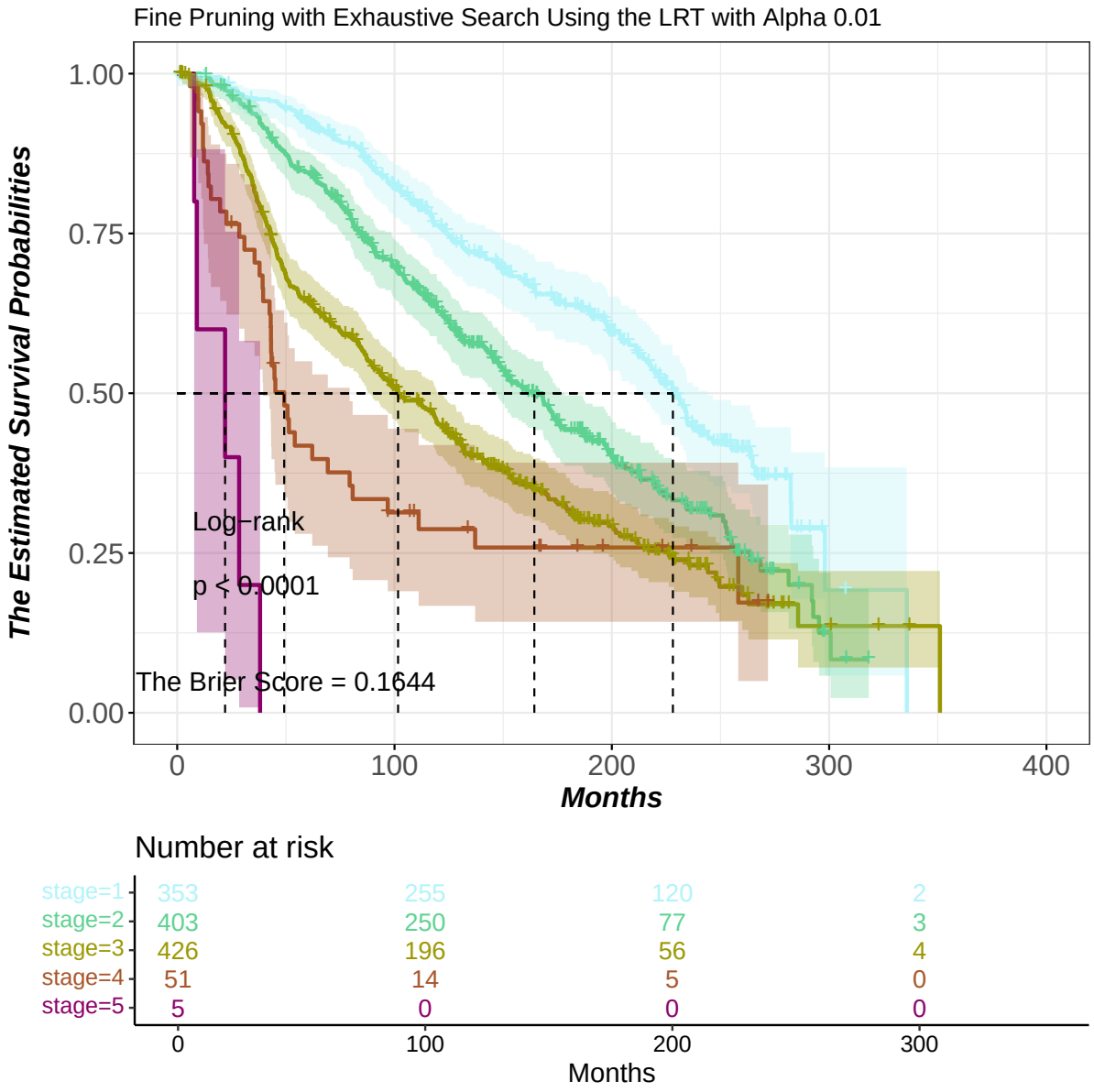


Figure 7.4: The Kaplan-Meier curves illustrating the stages obtained using OPERA, after pruning or restriction on the number of patients in each stage.

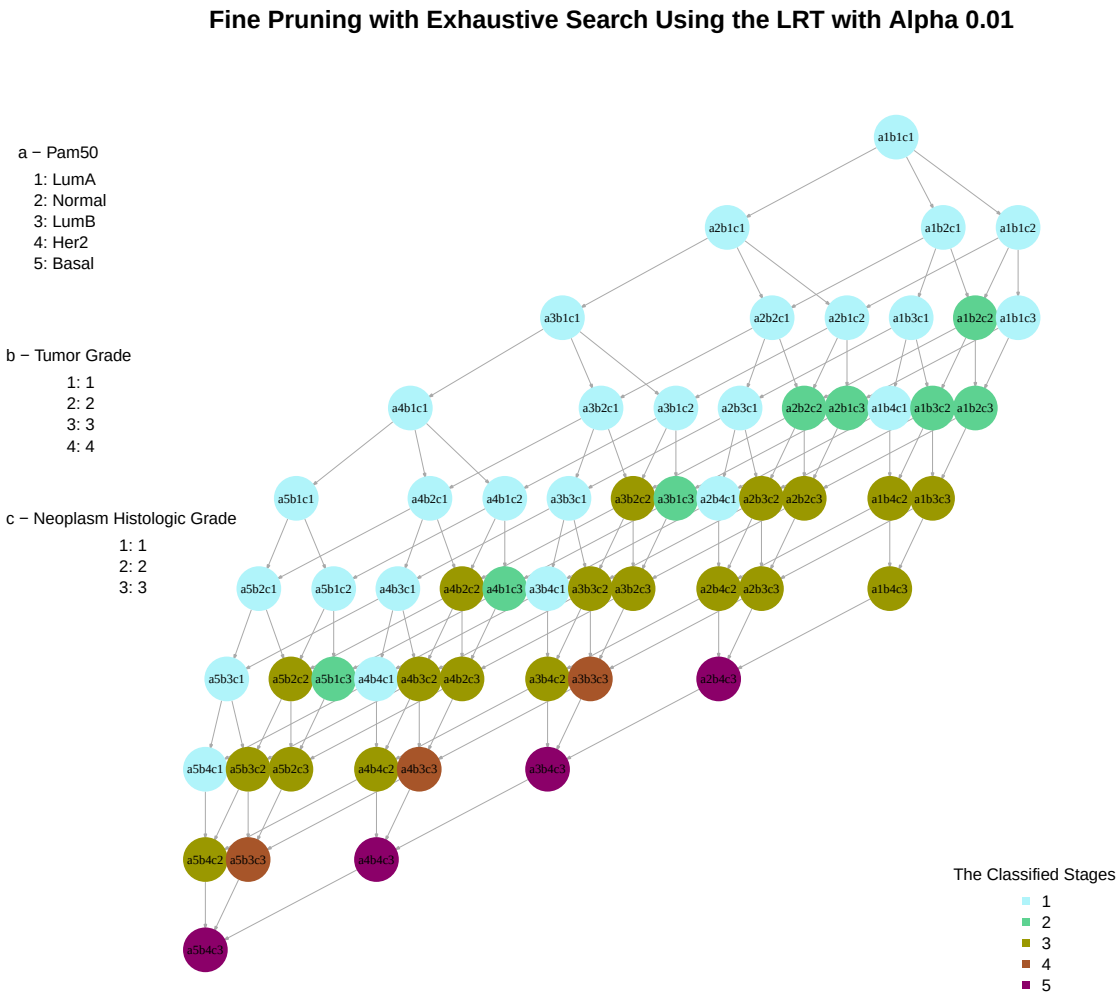


Figure 7.5: The tree-like network of risk factors labeled with stages obtained using OPERA, after pruning or restriction on the number of patients in each stage.

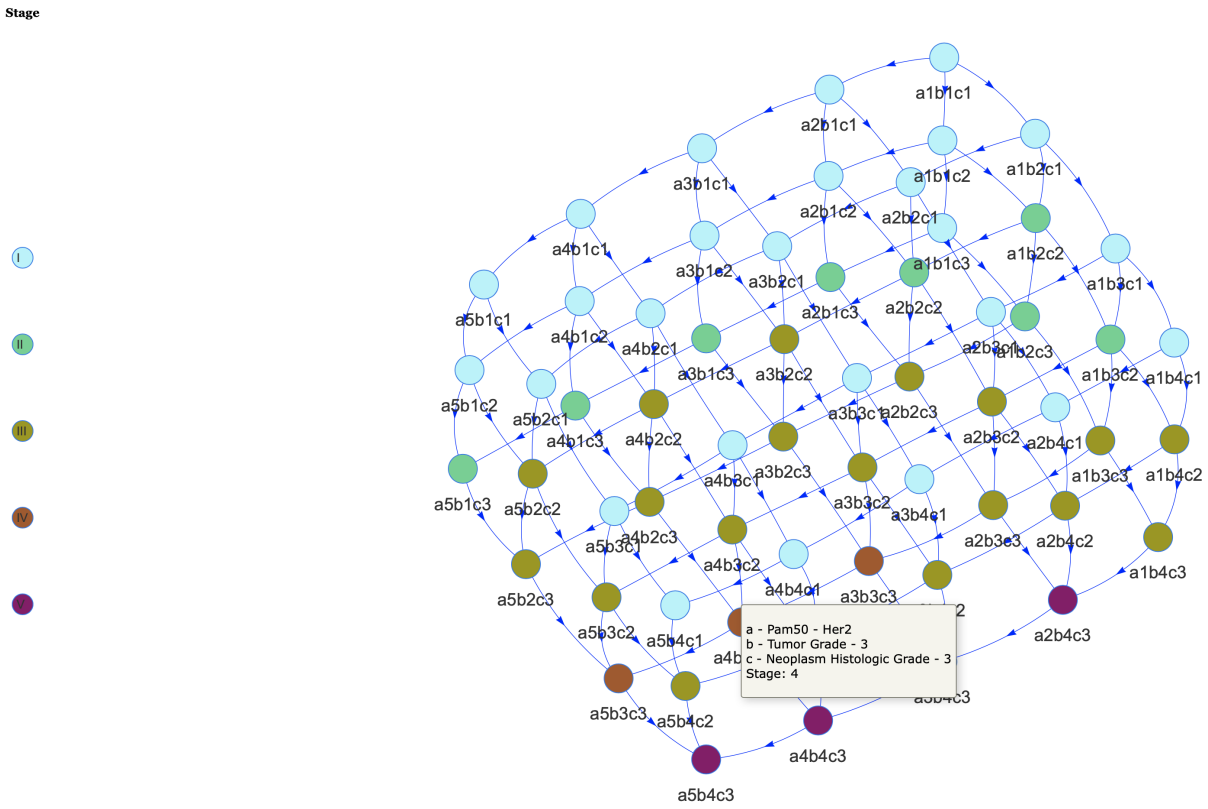


Figure 7.6: The interactive network of risk factors labeled with stages obtained using OPERA, after pruning or restriction on the number of patients in each stage.

Chapter 8

Summary and Discussion

8.1 Summary

Our primary research question revolves around cancer staging using ordinal risk factors. Previous attempts have employed tree-based methods such as CART, but they were limited in generating flexible grouping patterns, often resulting in rectangular shapes, as discussed in Chapter 1. To address this limitation, the lasso tree was developed, which could accommodate two ordinal risk factors and survival outcomes. Although it allowed for neighboring patterns like triangular shapes, it still struggled with generating non-neighboring patterns. Consequently, our focus turned to OPERA, a method specifically designed for two risk factors and survival outcomes, capable of generating grouping patterns that satisfy the ordering constraints. In Chapter 2, we extended OPERA to accommodate multiple risk factors and binary outcomes, while also comparing it to the extension of the lasso tree for the same scenarios. Simulation studies in Chapter 2 and Chapter 3 demonstrated that OPERA performed comparably to the lasso tree with neighboring patterns and exhibited substantial improvement with non-neighboring patterns for survival outcomes and binary outcomes, considering varying numbers of risk factors.

To address the issue of overfitting or over-partitioning, we introduced pruning as an additional step within the initial OPERA framework. This approach follows a bottom-up strategy similar to the commonly employed pruning concept in tree methods. Pruning can be applied to both OPERA and the lasso tree methods and has consistently demonstrated significant improvements in performance, as highlighted in Chapter 2-4. It is noteworthy that while the lasso tree may not generate non-neighboring patterns, it can still produce over-partitioned groups that can be refined through pruning to achieve performance comparable to or even better than OPERA.

Simulation studies consistently supported the notion that pruning does not result in inferior performance across various scenarios, regardless of the number of risk factors, outcome types, or the true underlying patterns. Two types of pruning methods are commonly employed: fine pruning and coarse pruning. In coarse pruning, all combinations of a stage are merged with another stage, while in fine pruning, some combinations are merged with one adjacent stage, and others are merged with the other adjacent stage. Both pruning methods can be implemented iteratively until a stopping criterion is met, such as the likelihood ratio test (LRT), Brier Score (BS), or a predefined number of stages. Among all criteria, without knowing the true number of stages, the LRT ($\alpha = 0.01$) consistently stands out as the best stopping rule, with very slight inferior performance in the worst-case scenario.

Across all simulation scenarios, coarse pruning consistently achieves performance comparable to fine pruning, with a very slight inferior performance in the worst-case scenario. Coarse pruning is generally recommended as it reduces computational burden without sacrificing much performance. When it comes to real data, on which fine pruning is computationally achievable, people can easily try both coarse and fine pruning using our R package **operap** in Chapter 7 and choose the result with better separation of the survival curves or better staging results. Fine pruning can be realized through two approaches: a brute-force method called exhaustive search and a quadratic programming constraint. Using a quadratic programming constraint can achieve performance on par with exhaustive search without the need to enumerate all possible solutions, resulting in a more efficient and effective pruning process when using exhaustive search is not computationally achievable.

In Chapter 4, we attempt to discretize continuous risk factors and observe how our methods find the true subgroups by combining risk categories with the same staging pattern. Simulation results demonstrate the same conclusion that coarse pruning with LRT ($\alpha = 0.01$) remains the best approach. In Chapter 5, we utilize several cancer datasets with survival outcomes to perform cancer staging, considering other covariates as adjusting factors during the staging process. Both the lasso tree and OPERA with pruning demonstrate reliable stages, as evidenced by well-separated survival curves. Notably, for binary outcomes, our methods showcase superior performance compared to penalized logistic regression models. Moving on to Chapter 6, we extend the application of OPERA beyond cancer staging to address a broader risk stratification problem. Specifically, we aim to cluster heterogeneous patients with advanced illnesses into homogeneous groups. By incorporating multiple risk factors, our approach improves upon the risk levels defined solely by a single variable known as the advanced illness trigger. The inclusion of additional risk factors results in more distinct stratifications, leading to a refined system. This case study serves as a demonstration of the versatility of our methods in tackling general clustering problems while adhering to the ordering constraints.

One widely known limitation of cancer staging research is the irreducible confounding effect from treatments patients received during their disease prognoses. We also assume that patients can obtain the best treatments they could have and focus on important baseline risk factors to predict the possible disease prognoses for patients. Future work will also focus on addressing the scalability of our methods, as the computational burden grows exponentially with an increase in the number of risk factors, even without considering pruning. Pruning, in itself, can be computationally demanding when utilizing the brute-force approach to fine pruning, and the computational cost escalates further with a higher number of risk factors. However, computational concerns about pruning can be alleviated, since coarse pruning can perform equally well as fine pruning with LRT ($\alpha = 0.01$) as the stopping rule. One possible way to deal with a large number of ordinal risk factors is to apply isotonic or monotonic regression models, which can be more computationally efficient. In addition to scalability, further exploration of the type I error rate in pruning with LRT is essential. This exploration will help identify the most effective approach to handle the multiple comparison issue that arises from pruning. By delving into these areas, we aim to enhance the efficiency and accuracy of our methods, paving the way for their broader application in real-world scenarios.

8.2 Discussion

8.2.1 Risk Adjustors and Risk Factors

Risk adjusters, denoted as Z, are the non-risk-factor covariates we adjust for during staging. They are the variables that would not appear in the final staging system, but during modeling, these variables have been adjusted for or conditioned on. Usually, risk adjusters have no inherent ordering. Instead of being conditioned on each category of a risk adjuster and leading to multiple staging systems (each corresponding to a specific category of a risk adjuster), we adjust for risk adjusters during modeling and control their impact on the outcome, which leads to one staging system on average across all categories of all risk adjusters.

Risk factors, denoted as r, are the variables that would appear in the final staging system, which clinicians can use to determine staging for patients. The difference between risk adjusters and risk factors is that, based on the final staging system, patients with differences in risk adjusters but no differences in risk factors are always in the same stage. However, patients with differences in risk factors are usually not in the same stage, regardless of their differences in risk adjusters.

Choosing between risk factors and risk adjusters depends on the context of the diseases. Risk

factors should be associated with staging, each of which has a monotonic relationship with the outcome measure. However, risk adjusters are not necessarily associated with staging but may be associated with the outcome measure or even act as a discriminatory variable. For instance, in some cancer studies, we only adjust for age but do not use age as a risk factor, since age can be associated with survival but is not necessarily directly related to stage.

8.2.2 The Total Number of Stages

The total number of stages can be determined by our staging methods without pre-specifying the total number of stages. By reducing (or increasing) the weight for the penalty term to relax (or tighten) the criterion used for classifying each stage, we can control either more (or fewer) stages in the end. Also, pruning has the feature of reducing the number of stages, either in a statistical way or by using pre-fixed stages.

In practice, clinical interpretation and complexity need to be considered when determining stages. In cancer stages, clinicians usually use no more than 5 stages, with each stage demonstrating a different survival rate. Currently, our pruning methods prune stages based on statistical significance when the pre-fixed number of stages is not given. However, a small, statistically significant difference between stages with similar outcome measures (e.g., median survival rates) should also be considered for merging in clinical practice to facilitate clinical interpretation in staging.

8.2.3 Uncertainty Quantification

In our simulation studies, we calculate the estimated mean edge misclassification rate with its 95% confidence interval to quantify its uncertainty for each simulation scenario. Due to the Monte Carlo process, the estimated mean edge misclassification rate converges after 500 simulations, and we are able to estimate the standard error and obtain the confidence interval. In the advanced colorectal neoplasia study, we use a 10-fold cross-validation to calculate the estimated cross-validation AUC and its 95% confidence interval.

In real data analysis or a simulation study, we can also use bootstrapping to quantify the uncertainty of related statistics, such as the estimated number of stages. Bootstrapping is a resampling approach, and each time, we can have a new dataset resampled from the original dataset. Depending on the type of outcome, we can also resample the datasets conditional on risk adjusters to ensure an unchanged impact from risk adjusters on the outcome measure.

Appendix A

A.1 Topological Properties

Lemma A.1.1. Suppose we have a collection of total ordered sets $S_1, S_2, ..., S_n$, each with m_i elements, denoted as $|S_i| = m_i$. Consider the Hasse diagram of the poset $(S_1 \times S_2 \times ... \times S_n, \leq)$, where the partial order is defined component-wise. The total number of edges in this Hasse diagram is given by $\sum_{i=1}^{n}(m_i - 1)\Pi_{j\neq i}m_j$.

Proof. Consider a set of elements $(s_1, s_2, ..., s_n) \in (S_1 \times S_2 \times ... \times S_n, \leq)$. If s_i is not fixed while all the other elements in the set are fixed, then s_i can take on values from 1 to 2, 2 to 3, ..., $m_i - 1$ to m_i, resulting in $(m_i - 1)$ edges. The total number of edges in the Hasse diagram for all possible values of the other elements is denoted as $\Pi_{j\neq i}m_j$, and thus the number of edges for s_i is $(m_i - 1)\Pi_{j\neq i}m_j$. To obtain the total number of edges in the Hasse diagram, we sum over all i. □

Lemma A.1.2. Let $S_1, S_2, ..., S_n$ be total ordered sets, where S_i has m_i elements denoted as $|S_i| = m_i$. Consider the Hasse diagram of the poset $(S_1 \times S_2 \times ... \times S_n, \leq)$, where the partial order is defined component-wise. The total number of rows in this Hasse diagram is given by $\sum_i m_i - n + 1$.

Proof. Suppose we have a set of elements arranged in a Hasse diagram, where the sum of all levels of elements in each node in a row is equal to the sum in the preceding row plus one. Specifically, let the sum in the first row be n, and the sum in the last row be $\sum_i m_i$. Then, the total number of rows in this Hasse diagram is $\sum_i m_i - n + 1$. □

Lemma A.1.3. Consider the Hasse diagram of the poset $(S_1 \times S_2 \times ... \times S_n, \leq)$, where $S_1, S_2, ..., S_n$ are total ordered sets with $|S_i| = m_i$ elements. Let the number of nodes in the jth row of this Hasse diagram be denoted by N_j. Then, we have $N_j = N_{\sum_i m_i - n - j + 2}$.

Proof. Let $N(j)$ denote the total number of nodes in the jth row of the Hasse diagram of the poset $(S_1 \times S_2 \times ... \times S_n, \leq)$, where $S_1, S_2, ..., S_n$ are total ordered sets with $|S_i| = m_i$ elements. The value of $N(j)$ can be obtained by solving the problem of balls in bins with limited capacity. Specifically, $N(j)$ is the number of ways to distribute $j-1$ indistinguishable balls into n bins, where each bin has a capacity of $m_i - 1$ balls. Since the distribution of empty spaces is symmetrical to that of balls, we have $N(j) = N(\sum_i m_i - n - j + 2)$. □

A.2 Important Mathematical Symbols

Table A.1: Important Mathematical Symbols

Letters	Meanings
S	poset
S_i	stage
y	outcome
r	risk categories or risk factors
Z	covariates or risk adjustors
ξ	coefficients for risk categories
α	coefficients for covariates
l	likelihood
γ	differences between the coefficients for risk categories and the coefficient for the reference level
μ	coefficient for the reference level
η	linear predictor
π	probability of experiencing a binary outcome
β	all coefficients
X	design matrix
u	the derivative of the likelihood with respect to linear predictor
A	The second derivative, also known as the Hessian matrix, of the likelihood with respect to the linear predictor
z	linear predictor plus the inverse of the Hessian matrix multiplied by the first derivative
λ	tuning parameter
D	down-set
U	residual set
m	total number of stages
s	an element in a poset
ζ	coefficients for stages
AIC	AIC value
BS	BS value

Table A.1: Important Mathematical Symbols

Letters	Meanings
δ	censoring

Bibliography

[Aka73] Hirotugu Akaike. "Information theory and an extension of the maximum likelihood principle,[w:] proceedings of the 2nd international symposium on information, bn petrow, f". In: *Czaki, Akademiai Kiado, Budapest* (1973).

[BB+14] Rachelle E Bernacki, Susan D Block, et al. "Communication about serious illness care goals: a review and synthesis of best practices". In: *JAMA internal medicine* 174.12 (2014), pp. 1994–2003.

[Bre+17] Leo Breiman et al. *Classification and regression trees*. Routledge, 2017.

[Bri+50] Glenn W Brier et al. "Verification of forecasts expressed in terms of probability". In: *Monthly weather review* 78.1 (1950), pp. 1–3.

[Bru10] A Brualdi Richard. *Introductory Combinatorics*. 2010.

[Cam+16] Joshua D Campbell et al. "Distinct patterns of somatic genome alterations in lung adenocarcinomas and squamous cell carcinomas". In: *Nature genetics* 48.6 (2016), pp. 607–616.

[Col+14] EACJ Collisson et al. "Comprehensive molecular profiling of lung adenocarcinoma: The cancer genome atlas research network". In: *Nature* 511.7511 (2014), pp. 543–550.

[Cur+12] Christina Curtis et al. "The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups". In: *Nature* 486.7403 (2012), pp. 346–352.

[Den52] PF Denoix. "Nomenclature classification des cancers". In: *Bull. Inst. Nat. Hyg.(Paris)* 7 (1952), pp. 743–748.

[Gar15] Vijay K Garg. *Introduction to lattice theory with computer science applications*. John Wiley & Sons, 2015.

[GI82] Donald Goldfarb and Ashok Idnani. "Dual and primal-dual methods for solving strictly convex quadratic programs". In: *Numerical analysis*. Springer, 1982, pp. 226–239.

[GI83] Donald Goldfarb and Ashok Idnani. "A numerically stable dual method for solving strictly convex quadratic programs". In: *Mathematical programming* 27.1 (1983), pp. 1–33.

[Giu+17] Armando E Giuliano et al. "Breast cancer—major changes in the American Joint Committee on Cancer eighth edition cancer staging manual". In: *CA: a cancer journal for clinicians* 67.4 (2017), pp. 290–303.

[Gra+99] Erika Graf et al. "Assessment and comparison of prognostic classification schemes for survival data". In: *Statistics in medicine* 18.17-18 (1999), pp. 2529–2545.

[GW10] Mithat Gönen and Martin R Weiser. "Whither TNM?" In: *Seminars in oncology*. Vol. 37. 1. Elsevier. 2010, pp. 27–30.

[Has17] Trevor J Hastie. "Generalized additive models". In: *Statistical models in S*. Routledge, 2017, pp. 249–307.

[Hig88] Nicholas J Higham. "Computing a nearest symmetric positive semidefinite matrix". In: *Linear algebra and its applications* 103 (1988), pp. 103–118.

[Imi+12] Marcin Imielinski et al. "Mapping the hallmarks of lung adenocarcinoma with massively parallel sequencing". In: *Cell* 150.6 (2012), pp. 1107–1120.

[Kah62] Arthur B Kahn. "Topological sorting of large networks". In: *Communications of the ACM* 5.11 (1962), pp. 558–562.

[Lin+16] Yunzhi Lin et al. "Advanced colorectal neoplasia risk stratification by penalized logistic regression". In: *Statistical methods in medical research* 25.4 (2016), pp. 1677–1691.

[Loh14] Wei-Yin Loh. "Fifty years of classification and regression trees". In: *International Statistical Review* 82.3 (2014), pp. 329–348.

[LPL15] Erin LeDell, Maya Petersen, and Mark van der Laan. "Computationally efficient confidence intervals for cross-validated area under the ROC curve estimates". In: *Electronic journal of statistics* 9.1 (2015), p. 1583.

[LWC13] Yunzhi Lin, Sijian Wang, and Richard J Chappell. "Lasso tree for cancer staging with survival data". In: *Biostatistics* 14.2 (2013), pp. 327–339.

[Net+12] Cancer Genome Atlas Research Network et al. "Comprehensive genomic characterization of squamous cell lung cancers". In: *Nature* 489.7417 (2012), p. 519.

[OV17] Edouard Ollier and Vivian Viallon. "Regression modelling on stratified data with the lasso". In: *Biometrika* 104.1 (2017), pp. 83–96.

[Par+09] Joel S Parker et al. "Supervised risk predictor of breast cancer based on intrinsic subtypes". In: *Journal of clinical oncology* 27.8 (2009), p. 1160.

[PC14] Carol A Parise and Vincent Caggiano. "Breast cancer survival defined by the ER/PR/HER2 subtypes and a surrogate classification according to tumor grade and immunohistochemical biomarkers". In: *Journal of cancer epidemiology* 2014 (2014).

[Per+00] Charles M Perou et al. "Molecular portraits of human breast tumours". In: *nature* 406.6797 (2000), pp. 747–752.

[Per+16] Bernard Pereira et al. “The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes”. In: *Nature communications* 7.1 (2016), pp. 1–16.

[Rue+19] Oscar M Rueda et al. “Dynamics of breast-cancer relapse reveal late-recurring ER-positive genomic subgroups”. In: *Nature* 567.7748 (2019), pp. 399–404.

[Tur19] Maintainer Berwin A Turlach. “Package ‘quadprog’”. In: (2019).

[Wan20] Tianjie Wang. *New Risk Stratification Algorithms to Harness Ordered Information in Categorical Variables*. The University of Wisconsin-Madison, 2020.