Rhineland-Palatinate Technical University Kaiserslautern-Landau,
Applied Machine Learning group
German Research Center for Artificial Intelligence,
Data Science and its Applications Research Department

Master's Thesis

Survival Prediction with Multi-Omics Data

Author:
Md. Naimur Islam

Matriculation Number: 415972
Submission Date: 08.11.2024

Supervisors: Prof. Dr. Sebastian Vollmer
Dr. David Antony Selby

Declaration of Authorship

Hereby, I declare that I have independently prepared the present work in accordance with the regulations of the Master's program in Computer Science, with the exception of the support provided by my supervisors. I have clearly and thoroughly documented all sources and resources used, including the internet, and have indicated all portions that have been taken from the work of others without alteration, whether verbatim, in abbreviated form, or with paraphrasing.

Md. Naimur Islam

Kaiserslautern, November 8, 2024

Abstract

As an essential field of Statistics, Survival analysis predicts various aspects of time-to-event data that includes censored data. Technological advancement of the analysis of biological molecules enables prediction of survival times, and risk of a specific disease occurrence by leveraging omics data. Omics data generally contains thousands of features compared to the minimal number of samples and the number of features only increases when additional omics types are integrated to make a multi-omics dataset, which is hard to interpret by humans and can also impact any model to analyze the data successfully.

The objective of this thesis is to compare the survival prediction performance of feature-selected subsets with the performance of the full multi-omics dataset. The thesis also explores how does the integration of various omics types within a multi-omics dataset influence survival prediction performance.

The findings show that it is uncertain whether models built on feature-selected subsets consistently outperform models built on the full multi-omics dataset. Similarly, it is unclear whether integrating more omics types to construct multi-omics dataset yields better predictive performance than selectively including a less carefully chosen omics types to construct a multi-omics dataset.

In general, survival analysis prediction performance heavily depends on the chosen multi-omics dataset, integration technique, selected survival analysis model and its configuration, and the considered prediction performance measure.

Contents

1 Introduction

1.1 Motivation

Unlike Genetics, which interrogates single genes or individual variants, Genomics was the first omics discipline, a global or comprehensive assessment of a set of molecules, focused on the entire genomes [1]. With the advancement of technology, analysis of biological molecules such as genomics, transcriptomics, proteomics, epigenomics, metabolomics, etc, are made efficient and performant. Nowadays, many databases such as the Cancer Genome Atlas (TCGA), the Gene Expression Omnibus (GEO), and NCI Genomics Data Commons (GDC) are publicly available that have different kinds of omics data for various diseases [2]. This kind of omics data can be leveraged for survival analysis to predict the survival times and risk of a specific disease occurrence.

At the start of survival analysis with genomics data, only a single omics data was included to build a prediction models [3]. With the availability of more and more omics types to analyze, focuses are shifting towards leveraging more omics types together to predict model outcomes. This kind of integration and combination of multiple omics types into a single dataset led to the term multi-omics data [4]. Integration of various omics data to a multi-omics data is not straightforward, and multiple approaches are proposed, such as early integration, mixed integration, intermediate integration, late integration, and hierarchical integration, each having its own pros and cons [5]. Clinicians mostly prefer data and models that are easily interpretable and applicable [6]. Integration techniques such as early integration or intermediate integration may be preferable because of the easy interpretability of data as they do not transform the omics data when integrating into multi-omics data, unlike mixed integration. Such integration technique raises another kind of problem because of the inherent nature of omics data being very high dimensional [7].

Omics data typically has thousands of features compared to the very small number of samples. Integrating multiple omics types increase the number of features beyond the capacity of any human to interpret the data without external help. In addition to that, biological data often contains redundant, noisy, and irrelevant features due to the nature of biological experiments that are contaminated during the experiment, as well as tools and equipment that often produce noisy data [8]. Reducing the unwanted features from the high-dimensional data is important beyond doubt to improve accuracy and reduce resource costs such as time and memory [9].

This reduction of the unwanted features can produce a subset of the original multi-omics data with a small number of relevant features that are easy to interpret by clinicians but should not come at a high cost of predictive performance [10]. To our knowledge, there is still a lack of studies that neutrally compare the independently selected features subsets predictive performance with the original multi-omics dataset in terms of survival analysis.

Moreover, there are many omics types, such as gene expression (RNA), miRNA expression, copy number segment, protein expression, and DNA methylation, available to construct multi-omics data [11]. Because of the very high dimensional nature of omics data, it can be quite overwhelming for any model to successfully analyze when all available omics types are integrated to make a multi-omics data. Determining the effect of survival prediction performance on incorporating different numbers of omics types into a single multi-omics dataset is also desirable.

1.2 Research Questions

The purpose of the thesis is to provide answers to the following research questions.

1. To what extent does survival prediction performance differ between a feature-selected subset and the full multi-omics dataset?

2. How does the integration of various omics data types within a multi-omics dataset influence survival prediction performance?

1.3 Overview

The thesis comprises six chapters, each building upon the previous to achieve its goal. Chapter 2 lays the background by introducing essential concepts necessary for the research, such as multi-omics integration strategies, dimensionality reduction models, and survival analysis models with evaluation measure metrics. Chapter 3 details the methods and datasets used in the thesis. Chapter 4 describes the experiment design and its results. Chapter 5 delves into the findings based on the achieved results. Finally, Chapter 6 concludes the thesis by summarizing the key findings.

2 Background

2.1 Multi-omics Data Integration Strategies

Omics data contains a specific type of biological information such as genomics, transcriptomics, proteomics, epigenomics, metabolomics, etc of a biological system or an individual [12]. When different omics types are present in one dataset, then that new dataset is called multi-omics data. Omic data generally have rows containing samples and columns containing biological variables. Integration of multiple omics data to multi-omics data is not straightforward, and multiple strategies have been developed for integration, each having its own pros and cons. Five such strategies have been commonly found in the literature and are described below [5].

1. Early integration: Every omics data set is concatenated into a large dataset. The number of observations stays the same, but the number of variables increases. This technique is commonly used because of its easy implementation and simplicity, as well as its more straightforward data interpretation. Although the size difference of the omics dataset can influence learning imbalance, the extent of the influence is not known.

2. Mixed integration: In mixed integration, each omics dataset is independently transformed into a more straightforward representation. This transformed representation can be less noisy with lower dimensions. The differences between omics datasets in terms of their data type, size, etc., are removed in the new dataset. Transformation can be done using graph-based methods, kernel-based methods, or deep learning-based methods.

3. Intermediate integration: Intermediate integration can be described as a technique that is capable of integrating multiple omics datasets without using a simple concatenation and prior transformation. The output dataset is a newly constructed representation that is common to all omics datasets. This can reduce both dimensionality and complexity and is often implemented after some robust pre-processing and feature selection. SLIDE [13] and an extension of mRMR [14] are examples of intermediate integration methods.

4. Late integration: Applying different models separately on each omics dataset and then combining their prediction is called late integration. It is a straightforward integration strategy that does not assemble different kinds of omics data. However, this strategy does not share knowledge between models at any point of the models learning process,

and as a reason cannot capture or utilize inter-omics interactions and complementary information. Moreover, combining predictions of different models remains a challenge because the combined output simply cannot be enough to actually exploit or understand multi-omics data. One can even argue whether late integration can be called a multi-omics integration as it boils down to an analysis of multiple single omics.

5. Hierarchical integration: At the molecular level, Hierarchical integration leverages the modular organization structure to exploit the nature of multidimensional data for multi-omics integration. It includes prior knowledge of relationships between different omics data to finally integrate them into multi-omics data.

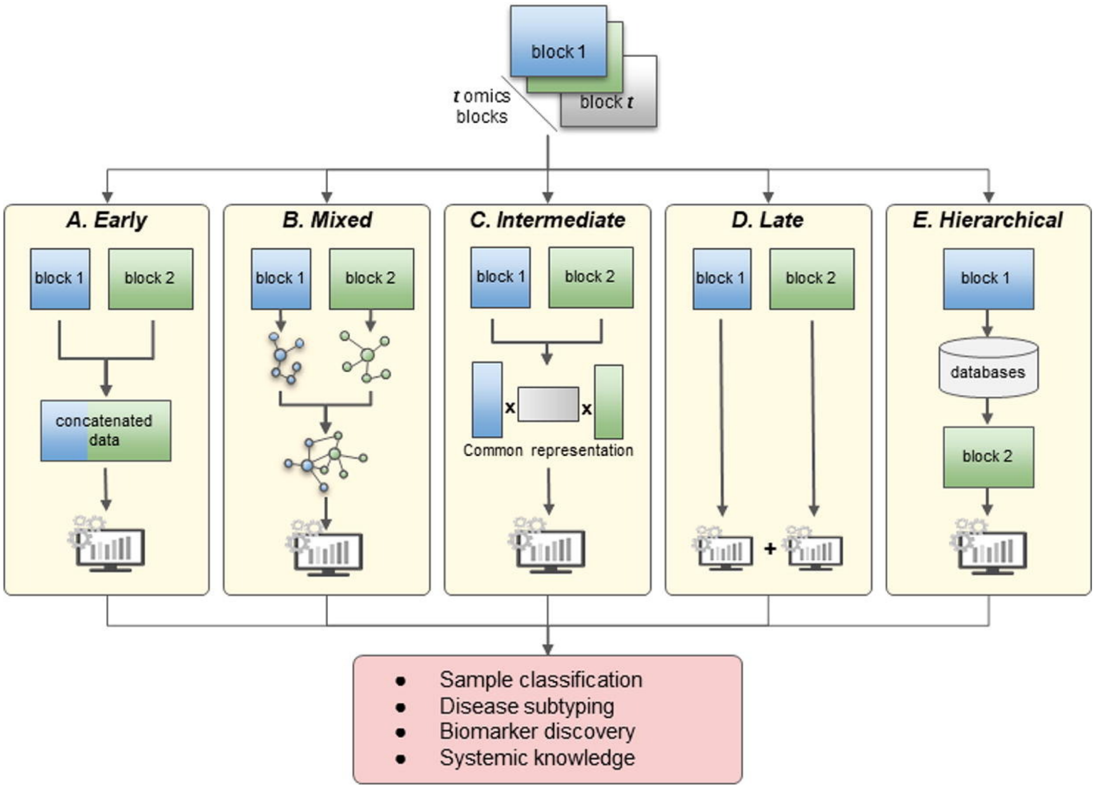


Figure 2.1: Integration strategies of multi-omics data [5].

2.2 Dimensionality Reduction Models

Dimensionality reduction of data, in general, can be categorized into two methods, such as feature selection and feature extraction [15] [16]. A subset of the most important and relevant features of the original features are selected on certain criteria for the given task in feature selection methods [16]. Selected features are not altered and they are a part of the original dataset, which preserves the relationships and meaning among the original features, making them more interpretable.

In comparison to the feature selection methods, the feature extraction methods aim to transform the original features to make a new one that better represents the original data [16]. $x = [x_1, x_2, ..., x_{p'}]$ in a feature space Ω_X of p'-dimension is transformed from a sample $y = [y_1, y_2, ..., y_p]$ in a space Ω_Y of p-dimension using a map of $X = f(Y)$ where $p' < p$ [16]. Feature extraction is less interpretable because the extracted features do not represent the relationships and meaning of the original features.

Feature selection and feature extraction can both be achieved in supervised, semi-supervised, and unsupervised manners where evaluation of selected or extracted features is done by known class labels, while alternative criteria are derived without knowing class labels in unsupervised approaches [17]. As the feature selection methods preserve the relationships among the original features and do not transform the features while making the feature subset, these methods are more interpretable by the domain experts; thus, feature selection methods are often preferred over feature extraction methods in several cases [18].

In multi-omics genomic data, feature selection methods can retain the gene signature by selecting a subset of genes while transforming the gene signatures, the feature extraction methods will not retain the originality of the original gene signature. Because of this reason, it is more suitable to select feature selection methods over feature extraction methods when experimenting with multi-omics data [15]. Depending on the computation of feature evaluation indicators, feature selection methods can be categorized into three such as filter methods, wrapper methods, and embedded methods [19].

2.2.1 Filter Methods

Independent of the selected classification method, filter methods act as pre-processing steps by selecting feature subset. The selection can be carried out using both univariate and multivariate approaches. Univariate approaches such as Infor [20] and Correlation-based Feature Selection [21] ignore feature dependencies and independently evaluate each feature according to specific criteria. Multivariate feature selection methods such as Minimum Redundancy Maximum Relevance (mRMR) [22] and ReliefF [23] are proposed to overcome the feature dependency ignoring problem. Filter methods are easy to implement, classifier independent, and, in general, faster than Wrapper and Embedded feature selection methods. Filter methods ignore the classifier methods to be used later, thus needs to be performed once and the resulted feature subset can be used to evaluate different classification methods.

2.2.2 Wrapper Methods

Wrapper methods take the classifier performance into account that guides the searching at each iteration to iteratively evaluate and select feature subset [24] because ”The m best features are not the best m features” [25] generally. Wrapper methods usually fall into two categories: greedy methods and random search methods [18]. In the hope of leading to a globally optimal solution, well-known greedy methods such as Sequential forward selection and Sequential backward selection make locally optimal choices [26]. They are iterative approaches where an initial solution is selected at the start and updated in each iteration by generating some alternative solutions to calculate their profitability and selecting the solution with the maximum profit to replace the old selection. Either by reaching the maximum iteration number or fulfilling a stopping criterion, the algorithm stops and the best selected feature subset is obtained [27]. Evaluation of the large numbers of possible subsets makes the greedy methods computationally costly for high-dimensional data as the number of features becomes too large.
Alternatively, random search methods such as Genetic Algorithm [28] and Particle Swarm Optimization [29] generate random solution space instead of multiple solutions to obtain the final feature subset. In theory, wrapper methods select more accurate feature subset by considering classifiers that will produce better accurate classification output when wrapper methods and classification methods are combined. But wrapper methods are computationally very resource hungry and if not stopped iterative process early have a higher risk of overfitting.

2.2.3 Embedded Methods

Embedded methods are specific to the given machine learning algorithms and select feature subset in the training process [30]. Regularization methods such as Lasso [31] and Elastic Net [32] and Decision Tree Algorithms such as RF-VI [33] are two categories the Embedded methods fall into. Feature selection is performed implicitly in regularization methods by forcing coefficients of the feature to be small or exactly to zero when the objective function model is regularized with minimizing the feature weighting of estimated generalization error. Unlike wrapper methods, embedded methods may be less prone to overfitting and computationally intensive while considering classifiers like wrapper methods. However, compared to wrapper methods and filter methods, sometimes embedded methods can produce worse classification performance [34].

2.2.4 Feature Selection and Multi-Omics Data

With integrated multidimensional analyses and large-scale genome sequences of more than 30 human tumors, The Cancer Genome Atlas (TCGA) is a public-funded project that has various types of omics data of different types of cancer [11]. TCGA provides a rich set of omics data types that includes Genomics, Transcriptomics, Epigenomics, Proteomics, miRNA-sequencing, and so on. Biological data often contains redundant, noisy, and irrelevant features due to the nature of biological experiments that are contaminated during the experiment, as well as tools and equipment that often produce noisy data [8] [35]. Reducing the unwanted features from the high-dimensional data is important beyond doubt to improve accuracy and reduce resource costs such as time and memory [9].

In theory, Embedded methods such as RF-VI and Wrapper methods such as Genetic Algorithm should produce better accuracy than Filter methods. However, experiments on multi-omics data showed that Filter methods like mRMR produce better accuracy than RF-VI [36] with slightly higher computation time but less memory. Wrapper methods, such as Genetic algorithm and Recursive feature elimination took too much computation time, so they are not suitable for multi-omics data for practical use [36]. These feature selection models were only considered with binary outcomes and were not clear on performance of survival data. Another study showed that the variance filter on omics data outperformed mRMR when applied the selected features to a survival analysis model such as CoxPH [10]. Although filtering was done on single omics data, this can be transferred to multi-omics data.

2.3 Survival Analysis Models

Survival analysis domain analyzes time-to-event data that consists of covariates, outcome (categorical and often binary) and time (until the outcome happens, generally refers as survival time). Survival analysis is different than other areas of Statistics because it incorporates ”censoring”, uncertainty of a real-world event occurring. For example, if a Cancer patient dies after five months of the initial treatment date, then the outcome is known, that is, the patient died after five months. Now, let’s say the patient is observed for ten months, and after that, the patient does not come for the treatment. In this case, we can be sure that the patient survived for ten months and after that the status of the patient being alive or dead is unknown. Then, it can be said that the patient is censored at ten months.

Statistical models generally learn from known outcome data, while Survival analysis tries to incorporate censored data to learn from the maximum information possible without knowing the true value of the outcome. The structure of the survival analysis data is different than that of other domains as the data is engineered to capture observed information rather than be directly modeled. Let,

- $X : X \subseteq R^p, p \in N_{>o}$ be data features / variables.
- $Y : T \subseteq R_{\geq 0}$ be the survival time.
- $C : T \subseteq R_{\leq 0}$ be the censoring time.
- T : min$\{Y, C\}$ be the time of observed outcome.
- $\Delta : \mathbb{I}(Y = T) = \mathbb{I}(Y \leq C)$ be event indicator.

Given the above-mentioned data structure, the event is observed when $Y \leq C$ and $\Delta = 1$, otherwise censored when $\Delta = 0$. There are three main types of censoring:

1. Right Censoring: Right censoring occurs if a subject has not encountered the event of interest (e.g., death) by the time the study ends or drops out from the middle of the study without experiencing the event, making the true outcome of the subject unknown. Formally, if the study period is $[t_s, t_e]$ for some $t_s, t_e \in R_{\geq 0}$, then occurs right censoring when $Y > t_e$ or when $Y \in [t_s, t_e]$ and $C \leq Y$. It is the most common type of censoring, and this study focuses on it.

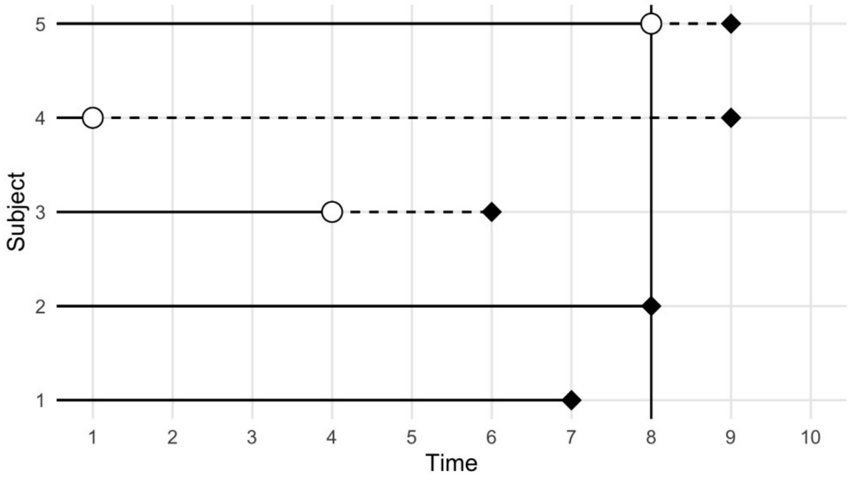


Figure 2.2: Censored and dead subjects (y-axis) over time (x-axis) [37]

Here, the vertical line indicates end time of the study. Censoring time is indicated using white circles, and true death time by black diamonds. Subjects 1 and 2 die in the study time while subjects 3 and 4 are censored. After the end of the study, subject 5 dies.

2. Left Censoring: Left censoring occurs if the subject has already encountered the event of interest (e.g., death) before start of the time of the study given that the exact event time is unknown but certain that the event has happened. Formally, left censoring can be represented as $Y < t_s$.

3. Interval Censoring: Interval censoring happens when the exact time of the event is not known, but certain that the event has occurred within a time interval of the study period.

When the estimated probability of the time of an event is greater than a certain time t, then the probability is represented by the Survival Function ($S(t)$) that is defined as:

$$S(t) = P(T > t)$$

Here, T is the time of observed outcome, and t is the time of interest.
Given that the individual has survived up to time t, then the event occurring instantaneous rate is represented by the hazard function ($h(t)$), which can be defined as:

$$h(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t | T \geq t)}{\Delta t}$$

2.3.1 Classical Survival Models

The Kaplan-Meier is a non-parametric distribution estimator which estimates the survival function from a given dataset. Based on a subject's survivability beyond time t, it defines a survival function $S(t)$ as the probability based on the censoring and the number of events at each time point [38]. The survival function can be formulated as:

$$\hat{S}(t) = \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i}\right)$$

where t_i is the time point at which at least one event occurred, d_i is the number of events at time t_i, and n_i is the number of observations that are yet to experience the event but are at risk just before time t_i.
When cumulative hazard is of interest with the survival function, the Nelson-Aalen estimator is used alongside to estimate the cumulative hazard function in survival analysis. It is an estimator of the accumulated risk over time to experiencing an event [39] [40]. At a specific time t, the Nelson-Aalen estimator for the cumulative hazard function $H(t)$ is:

$$\hat{H}(t) = \sum_{t_i \leq t} \frac{d_i}{n_i}$$

where t_i is the time point at which at least one event occurred, d_i is the number of events at time t_i, and n_i is the number of observations that are yet to experience the event but are at risk just before time t_i.

The survival function $S(t)$ can be estimated by the cumulative hazard $H(t)$ with the following relationship:

$$S(t) = e^{-\hat{H}(t)}$$

This indicates that the survival probability decreases as the cumulative hazard increases. In practice, the Kaplan-Meier estimation of a survival distribution is consistent and the simplest, and it is the most widely used non-parametric estimator as a baseline that is utilized to judge other models to ascertain overall performance [4]. Due to poor predictive performance of explanatory variables exclusion, above mentioned two estimator are not used for prognosis directly but as graphical tools and baselines.

Another non-parametric estimator to calculate cumulative incidence function $F_j(t)$ that represents the probability based on a condition that by time t, a specific event will occur of type j that can be written as Akritas estimator [41]:

$$\hat{F}_j(t) = \sum_{t_i \leq t} \frac{d_{ij}}{n_i} \prod_{t_k < t_i} \left(1 - \frac{d_k}{n_k}\right)$$

where t_i is the event time point. d_{ij} is the number of j type event which occur at time t_i. d_k is the number of events at time t_k, n_i is the number of observation that are yet to experienced the event but are at risk just before time t_i and $\prod_{t_k<t_i}(1 - \frac{d_k}{n_k})$ is the subjects surviving probability past previous time points to ensure the risk of subjects for the interest event by time t_i.

The Cox Proportional Hazards Model (CoxPH) is probably the most used method in survival analysis that estimate effect of covariates or predictors on the hazard rate [42]. It a semi-parametric model that express the effect of covariates multiplicatively on the hazard function without assuming a specific base hazard distribution over time. For an individual, the hazard function $h(t/X)$ with covariates $X = (X_1, X_2, ..., X_n)$ is:

$$h(t|X) = h_0(t)\,\text{exp}(X.\beta)$$

where $h(t|X)$ is the hazard function for an individual with covariates X at time t. $h_0(t)$ is the hazard when all covariates $X = 0$. $\beta = (\beta_1, \beta_2, ..., \beta_n)$ is the vector of each covariate regression coefficients. $\text{exp}(X.\beta) = \text{exp}(X_1\beta_1 + X_2\beta_2 + ... + X_n\beta_n)$ is the proportional effect of X on the hazard. CoxPH is a highly accessible and transparent model for survival analysis the has excellent predictive performance and has been routinely outperforms sophisticated Machine Learning models or at least has not unperformed [37].

2.3.2 Machine Learning Survival Models

Random Survival Forest (RSF) is a powerful machine learning model for survival analysis which has been studied extensively for the past four decades [37] [4]. RSF can be summarized into the following five models:

1. Relative Risk Tree (RRT): RRT adopts proportional hazard model of CoxPH to use a deviance splitting rule for ranking prediction at terminal node [43]. This model is the least transparent and accessible among all the RSF models discussed here. The prediction is harder to interpret and the assumptions the model makes may not be that realistic. Moreover, the performance is worse than other types of RSF models [37].

2. Relative Risk Forest (RRF): RRF is a extension of RRT which unlike RRT prediction during tree growing process, predicts after the tree is grown in every iteration [44]. Although in theory RRF can outperform RRT, but there are no implementation or usage found in literature. As a result RRF can not be considered because of it's predictive performance is simply unknown.

3. RSDF-DEV: Another extension of RRT assuming a PH and creating a random forest by introducing a bagging composition with a deviance splitting rule is RSDF-DEV [45]. Terminal node ranking prediction is altered with bootstrapped Kaplan-Meier prediction to make the model more transparent and accessible. However, the predictive performance is worse than RSF [37].

4. Random Survival Conditional Inference Framework Forest (RSCIFF): RSCIFF is a conditional inference model that predict log-survival time using weighted average of Kaplan-Meier estimation in the terminal node where inverse probability of censoring loss function is leveraged for splitting rule [46]. The implementation of RSCIFF is complex, making it less transparent and accessible to include in benchmark studies.

5. RSDF-STAT: Leveraging bootstrapped Nelson-Aalen estimation for terminal node prediction with a choice of log-rank and log-rank score hypothesis tests, RSDF-STAT probably is the most general and used model among the RSF variation [47]. It is highly accessible, transparent and performs well on most data. There are several implementations available in different programming languages. In this thesis, RSDF-STAT will be refereed as RSF from now on.

Gradient Boosting Machines (GBM) considers survival context by a sensible choice selection of loss function unlike other machine learning algorithms that typically ignored survival analysis at their early stages [48]. Although there has been a long gap of developing GBM in survival context, but recently GBM are catching on. Instead of predicting survival distribution directly, in general GBM makes ranking predictions.

GBM-COX estimate coefficients of Cox model using proportional hazard assumption for data distribution prediction in it's boosting framework [48]. Suitable loss function is minimized to predict $\hat{g}(X^*) = \hat{n} := X^*\hat{\beta}$. Minimizing negative partial log-likelihood, $-l$ with proportional hazard assumption can be formulated as:

$$l(\beta) = \sum_{i=1}^{n} \Delta_i \left[n_i - \log \left(\sum_{j \in R_{t_i}}^{n} \exp(n_i) \right) \right]$$

where $n_i = X_i\beta$ and R_{t_i} is sample set with risk at time t_i.

Although GBM are well-understood and transparent, but GBM-COX is not very flexible for custom implementation.

CoxBoost model optimize partial log likelihood penalization to boosts CoxPH by taking mandatory variates [49]. Mandatory variates make the model more interpretable and allow inclusion of prior expert knowledge. A componentwise framework of CoxBoost is implemented using R package "CoxBoost", though a non-componentwise framework also exists [50]. CoxBoost performs well, but due to the algorithm being complex, it is less transparent [37]. Moreover, it is less accessible as only one off-shelf implementation exist and custom implementation is harder than other GBM methods.

GBM-COX and CoxBoost are GBM models for proportional hazards data. For non proportional hazards data, GBM for accelerated failure time models are proposed that are fully parametric and estimate linear predictor, $\hat{g}(X_i) = \hat{\eta}$, simultaneously [51]. These models do not mandate assumption of often-unrealistic proportional hazards on the data to predict ranking. Experiments suggest that they can outperform CoxPH and are transparent and accessible as GBM-COX.

GBM can be modified to measure different kinds of measurement, such as Gehan loss, Buckley-James imputation, Harrell's C and Uno's C. Although GBM models are useful in survival analysis that can outperform a classical model such as CoxPH, but these models are resource intensive. One should be careful to leverage GBM in very high-dimensional data due to the limitation of the resource at hand and not look only at predictive performance.

Artificial Neural Networks (ANN) have been adopted for survival analysis for decades in contrast to other machine learning models. ANN are in general consider as a black-box model and are less interpretable that decrease rapidly with an increased number of nodes or hidden layers. On the other hand, ANN adaptation of survival analysis are mostly implemented using Python language and with fewer implementations available in R language make them less accessible [37]. Many researchers claim to adopt ANN for survival analysis, but in reality the task of the adopted model is to find probability of death at a specific time point which cannot be call as a survival analysis. Also, superior performance of classification task over Cox models cannot be considered as a survival task [37].

ANN-COX estimates prediction function $\hat{g}(X^*) = \phi(X^*\hat{\beta})$ to propose a non-linear proportional hazard model [52]:

$$h(t|X_i, \theta) = h_0(t)\,\text{exp}(\phi(X_i\beta))$$

where $\theta = \beta$ are model weights and ϕ is the sigmoid function. The model is trained with partial-likelihood:

$$L(\hat{g}, \theta|D_0) = \prod_{i=1}^{n} \frac{\text{exp}(\sum_{m=1}^{M} \alpha_m \hat{g}_m(X^*))}{\sum_{j\in R_{t_i}} \text{exp}(\sum_{m=1}^{M} \alpha_m \hat{g}_m(X^*))}$$

where at time t_i the risk group is R_{t_i}, number of hidden unit is M, model weights are $\theta = \{\beta, \alpha\}$ and $\hat{g}_m(X^*) = (1 + \text{exp}(-X^*\hat{\beta}_m))^{-1}$. ANN-COX has one hidden layer that is trained using back propagation with Newton-Raphson weight optimization. This model does not outperform a CoxPH even with independent studies using pre-processing and hyper-parameter tuning [52].

COX-NNET adopt ANN-COX by maximising regularized partial log-likelihood [53]:

$$L(\hat{g}, \theta \mid D_0, \lambda) = \sum_{i=1}^{n} \Delta_i \left[\hat{g}(X_i) - \text{log}\left(\sum_{j\in R_{t_i}} \text{exp}(\hat{g}(X_j)) \right) \right] + \lambda\left(\|\beta\|_2 + \|w\|_2\right)$$

with weights $\theta = (\beta, w)$, bias b, tanh activation function σ and $\hat{g}(X_i) = \sigma(wX_i + b)^T\beta$. Overfitting is prevented by incorporating dropout with weight decay. COX-NNET is not performant in terms of CoxPH performance.

DeepSurv extends ANN-COX and COX-NNET with multiple hidden layers where weight decay average negative log partial likelihood is chosen as error function [54]:

$$L(\hat{g}, \theta \mid D_0, \lambda) = -\frac{1}{n^*} \sum_{i=1}^{n} \Delta_i \left[\left(\hat{g}(X_i) - \text{log} \sum_{j\in R_{t_i}} \text{exp}(\hat{g}(X_j)) \right) \right] + \lambda\|\theta\|_2^2$$

where $n^* := \sum_{i=1}^{n} \prod(\Delta_i = 1)$ is the uncensored observations number and $\hat{g}(X_i) = \phi(X_i|\theta)$ is the prediction object as ANN-COX. The author claimed that it can outperform CoxPH, but independent experiments do not confirm this claim [55].

DNNSurv [55] trains a regression ANN with squared error loss and sigmoid activation to compute pseudo survival probability using jackknife-style estimator at first with:

$$\tilde{S}_{ij}(T_{j+1}, R_{t_j}) = n_j \hat{S}(T_{j+1} \mid R_{t_j}) - (n_j - 1)\hat{S}^{-i}(T_{j+1} \mid R_{t_j})$$

where for risk set R_{t_j}, $\hat{S}$ is inverse probability of censoring weighted Kaplan-Meier estimator, for all observation excluding i in R_{t_j}, $\hat{S}^{-i}$ is the Kaplan-Meier estimator and $n_j := |R_{t_j}|$. DNNSurv author did not find any improvement over other models mentioned above in terms of C-index and Brier score evaluation. With a valid proportional hazard assumption, DNNSurv failed to outperform the CoxPH model, although on the non-portional hazard dataset, DNNSurv outperformed Cox models.

DeepHit uses a deep neural network to directly learn survival time distributions without assuming any stochastic underlying process that allows the possibility of covariates and risk relationship changing over time [56]. The survival function is found using $\hat{S}(t_k|X^*) = 1 - \sum_{k=1}^{K} \hat{g}_i(t_k|X^*)$ where $k = 1, 2, ..., K$ is distinct time interval and g_i is prediction of failure at each time interval. With respect to separation, DeepHit outperformed CoxPH and RSF, but demonstrated worse performance than CoxPH with respect to integrated Brier score [57].

While above mentioned survival ANN focus on probabilistic prediction, RankDeepSurv [58] tackles the deterministic problem by predicting survival time $\hat{T} = (\hat{T}_1, \hat{T}_2, ..., \hat{T}_n)$. The proposed composite loss function is:

$$L(\hat{T}, \theta | D_0, \alpha, \gamma, \lambda) = \alpha L_1(\hat{T}, T, \Delta) + \gamma L_2(\hat{T}, T, \Delta) + \lambda \|\theta\|_2^2$$

where model weights are θ and shrinkage parameter is $\alpha, \gamma \in R_{>0}, \lambda$. With an unclear comparison, the author claimed superiority of RankDeepSurv over CoxPH, RSF and DeepSurv, making an independent study necessary to support the claim [37].

DeepOmix is another recent ANN model that enables multi-omics dataset to incorporate user define prior biological knowledge that is said to outperform CoxPH, RSF [59]. No implementation of DeepOmix is found to independently verify the claim. The complex structure of DeepOmix makes it less transparent and lack of implementation makes it less accessible for independent researchers.

2.3.3 Evaluation Measure for Survival Models

The simplest and most common way to measure a model's ability to distinguish risk levels is with concordance indices (C-index) [2]. These indices look at how often the model successfully separates pairs of observations into 'low-risk' and 'high-risk' groups. C-index value resides between [0, 1] with 0 being no separation, 1 being perfect separation and 0.5 being random separation. C-index may also be reported as a value in [0, 1], as a discriminatory power or as a percentage.

Improvement percentage of a model above the baseline 0.5 refers as discriminatory power. With a concordance of 0.7, the discriminatory power of the model is (0.7 - 0.5)/0.5 = 40%. Discriminatory power represents a model improvement over a baseline, but can be easily confused with percentage reporting of concordance of 0.7 as 70%. Harrell's concordance index, C_H [60] and Uno's concordance index, C_U [61] are two of the most common C-index measurement metric found in literature.

C_H ignores the pairs which are censored for shorter survival time and is affected by the presence of censoring [62]. C_U also suffers drastically with increased censoring than other concordance measures [62]. Both of the measures are not perfect as they are affected by censoring to some extent that can lead to over-confidence and under-confidence for a model discriminatory ability. C_U has observed to report value as 0.2 when the true estimation was 0.6 and reporting C_H = 0.7 may be incorrect as different amounts of censoring can mean different things [62].

Both C_H and C_U tend to produce similar values and comparison of studied models with same dataset will not be affected by the instability from censoring. As a result, utilizing C-index for this thesis model evaluation is not of concern as the censoring affects equally for selected models and are free from above mentioned problems. However, comparing a concordance from different study remains a challenge where the datasets differ in terms of censoring proportion with the sample size [37].

Another scoring rule for classification is the Brier score [63] where scores are minimized with true prediction.

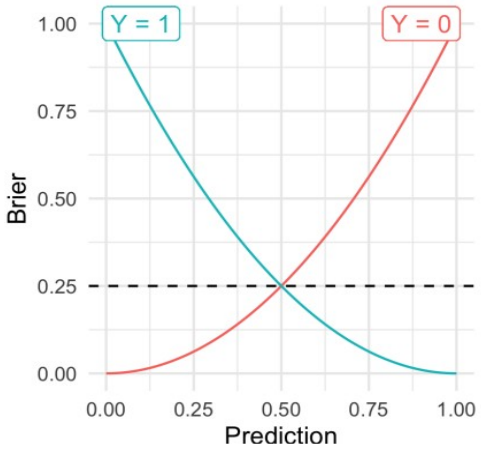


Figure 2.3: Brier scoring rules for probabilistic predictions. [37]

Here x-axis is probabilistic prediction and y-axis is Brier score. Blue line is varying Brier score when true outcome is 1 and red line is varying Brier score when true outcome is 0. Both scores are minimized for correct prediction. The range for the Brier score is [0, 1] and lower value indicates better survival prediction performance.
From the Figure 2.3 cutoff, it can be interpreted that a value below 0.25 has better predictive performance than an uninformed prediction of value 0.5. By predicting cumulative distribution function with the true event over the entire distribution, instead of prediction of correctness at a single point, Brier score for classification can be extended for regression and used as Integrated Brier Score (IBS) for survival settings. This thesis also leverage IBS alongside C-index as IBS is probably the most utilized measurement after C-index [2].

2.3.4 Survival Prediction and Multi-Omics Data

In the experiment done in [37], author stated that among the machine learning models, GBM and RSF were methods that generally perform well. From classical models, CoxPH also performed well, though it may fall sort when dealing with high dimensional data. Another benchmark study was done in [4] with multi-omics data where CoxBoost and RSF were two models that deemed applicable for multi-omics survival prediction. A general workflow of multi-omisc survival analysis is shown in Figure 2.4.

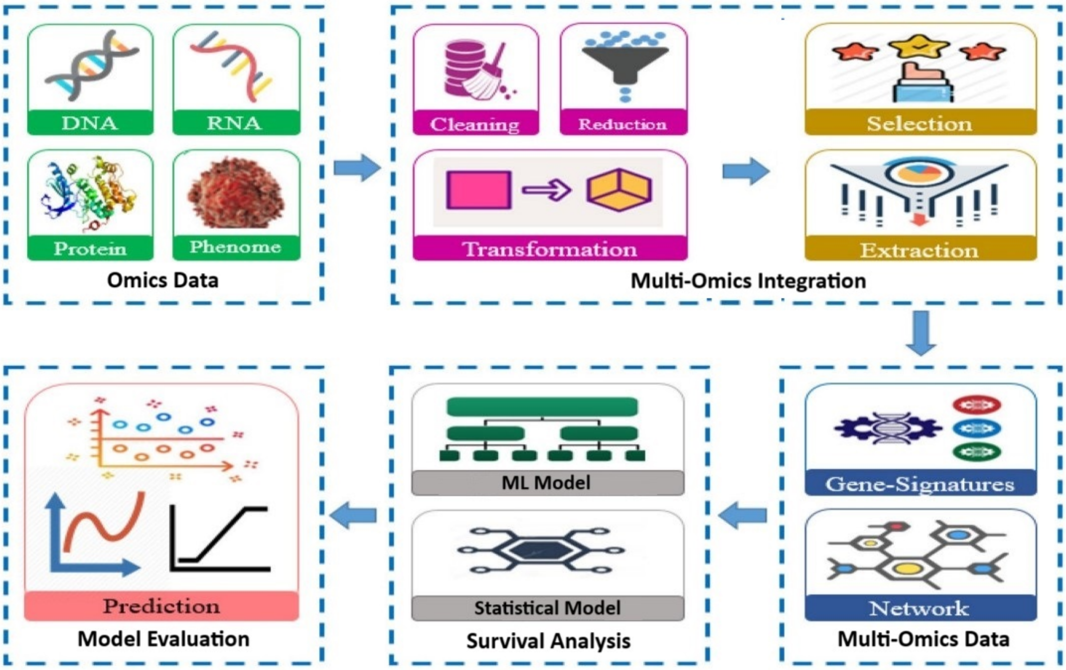


Figure 2.4: A General workflow of multi-omics survival analysis.

3 Methods and Datasets

3.1 Variance Filter

Variance filter is a feature selection method that creates a subset of the given dataset by including features that have very high variance in the dataset assuming that the features with high variance provide more useful information for distinguishing between target classes and remove features of the given dataset that have very low variance under the assumption that low variance features do not provide that much useful information for target classes distinction [19].

Features with low variance in general have values that are quite same across all samples and are unlikely to contribute to model's predictive performance compared to features with high variance. For a feature set $X = \{X_1, X_2, ..., X_N\}$ with N observations, the variance σ_X^2 of X is calculated as:

$$\sigma_X^2 = \frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})^2$$

where X_i is the i-th value of X and $\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$ is the mean of X.

To include or exclude the feature to the subset, a threshold θ is determined by which features are be discarded. This can be a fixed value or a percentile of the variances among the original data. Steps to perform variance filter is:

- Decide a threshold θ by which features will be included or excluded.
- For each feature of X, calculate σ_X^2
- Include any feature of X that has $\sigma_X^2 \geq \theta$ and exclude any feature of X that has $\sigma_X^2 < \theta$ to the new dataset.

3.2 Cox Proportional Hazards (CoxPH)

Assuming the event has not occurred yet, the CoxPH model assesses the effect of predictor variable on the hazard rate that represents event occurring risk at a particular time. CoxPH does not require a specific probability distribution of survival time to analyze the influence of multiple covariates on survival [42]. The hazard function $h(t|X)$ is defined as:

$$h(t|X) = h_0(t)\,\text{exp}(X.\beta)$$

where $h(t|X)$ is hazard rate for an individual with covariates X at time t. $h_0(t)$ is the baseline hazard function at time t when all covariates $X = 0$. $\text{exp}(X.\beta) = \text{exp}(X_1\beta_1 + X_2\beta_2 + ... +$

$X_n\beta_n)$ is the proportional effect of X on the hazard and $\beta = \{\beta_1, \beta_2, ..., \beta_n\}$ is the coefficients representing the effect on the hazard of each covariate X_i.
For a covariate X_i, the hazard ratio can be described as:

$$\text{Hazard Ratio for } X_i = \exp(\beta_i)$$

Here, if $\beta_i < 0$, then an increase of X_i decreases the hazard or risk of event, leads to longer survival time. If $\beta_i = 0$, then covariate X_i has no effect on survival. Finally, if $\beta_i > 0$, then an increase of X_i also increases the hazard or risk of event, suggest a shorter survival time [64].
Proportional hazards of the CoxPH assume that the hazard ratio between to training observations is constant over time. The ratio for two observations with covariates X_A and X_B does not depend on t and mathematically:

$$\frac{h(t|X_A)}{h(t|X_B} = \exp\left(\sum_{i=1}^{p} \beta_i (X_{A,i} - X_{B,i})\right)$$

Since the baseline hazard $h_0(t)$ is unspecified, parameters β are estimated using partial likelihood [65]. Rather than exact timing, the partial likelihood $L(\beta)$ focuses on the order of the events occurring at times $T_1, T_2, ..., T_D$ for a dataset with N individuals.

$$L(\beta) = \prod_{i=1}^{D} \frac{\exp(X_i \cdot \beta)}{\sum_{j \in R(T_i)} \exp(X_j \cdot \beta)}$$

where D is the set of events and $R(T_i)$ is the risk set at time T_i.
Given X covariates, the probability of surviving beyond time t, the survival function $S(t|X_i)$ can be estimated using hazard by:

$$S(t|X_i) = \exp(-h_0(t) \exp(X.\beta))$$

One problem with the proportional hazard assumption is that the risk of event over time may not be constant and the proportional hazard assumption may be unrealistic [37].

3.3 Random Survival Forests (RSF)

Random Survival Forests (RSF) is an extension of the random forest developed by Breiman [33] where multiple survival trees are created using a bootstrapped sample of the original data and each node is split with a random subset of variables [47]. Each tree predicts the survival probability or at a given time the risk of event occurrence also known as hazard function and the forest provides an averaged prediction. Each tree is a survival tree in RSF and rather than traditional regression or classification criteria, each node is split based on survival criteria while constructing each tree. Log-rank statistics is a common splitting criterion which assesses the wellness of a split separating groups with different survival in RSF [47].

Let's assume T_i is the observed survival time for i-th sample. δ_i is the event where $\delta_i = 0$ means censored and $\delta_i = 1$ means the event occurred and D is the set of all individuals in a node. Maximizing log-rank statistic L over potential splits can maximize survival difference between the child nodes.

$$L = \frac{\left(\sum_{j \in L} d_j - \frac{N_L}{N} \sum_{j \in D} d_j\right)^2}{\frac{N_L}{N} \sum_{j \in D} d_j (1 - \frac{N_L}{N})}$$

where N is the number of samples in the original node D, N_L is the number of samples in node L and d_j is the number of events at time j.

For each sample, a cumulative hazard function is produced in each tree and once the multiple survival trees are grown, the forest averages them to produce a stable cumulative hazard function. Based on a single tree k at time t, the cumulative hazard function can be denoted by $h_k(t|X_i)$ for a sample i that has a feature vector X_i [47]. With the total number of trees K in the forest, the overall cumulative hazard function can be averaged across all trees of the forest for a sample i:

$$h(t|X_i) = \frac{1}{K} \sum_{k=1}^{K} h_k(t|X_i)$$

The survival function $S(t|X_i)$ for each sample can be estimated by the relationship between the cumulative hazard function $h(t|X_i)$ and the survival as [66]:

$$S(t|X_i) = \exp(-h(t|X_i))$$

The following algorithm is based on [47]:

1. Draw B bootstrap samples from the original data.
2. For each bootstrap sample, grow a survival tree by randomly select p candidate variables that maximizes survival difference between daughter nodes.
3. Grow a full size tree that should have $d_0 > 0$ unique deaths at terminal node.
4. Calculate cumulative hazard function for each tree and obtain the ensemble cumulative hazard function by averaging them.

3.4 Datasets

The Cancer Genome Atlas (TCGA) is a publicly funded project with a goal to make over 30 human tumours large scale genome sequencing omics data publicly available to improve diagnosis, treatment and ultimately prevent cancer [11]. TCGA offers different types of omics data such as Transcriptomics, Genomics, Proteomics, Epigenomics that can be utilized for multi-omics data analysis. Among different cancer types, six of them are selected for this study with three types of omics data such as Gene Expression RNA Sequencing (RNA), miRNA Expression Quantification (miRNA), and Protein Expression Quantification (Proteome). Overall survival times and status of six cancer patients from TCGA are shown in Figure 3.1.

- TCGA-BRCA: TCGA-BRCA refers to Breast Invasive Carcinoma dataset for breast cancer. This dataset includes Luminal A, Luminal B, HER2-enriched, Triple-negative breast cancer, etc molecular subtypes of breast cancer.

Table 3.1: Details of TCGA-BRCA dataset.

Omics Type	Sample Size	Feature Size
RNA	1095	60660
miRNA	1079	3762
Proteome	881	217
Clinical	1098	70

- TCGA-LUAD: One of the common forms of lung cancer is Lung Adenocarcinoma, a subtype of non-small cell lung cancer, is presented in TEGA-LUAD dataset. Including subtypes such as Epidermal growth factor receptor (EGFR), KRAS Mutant, and TP53 Mutant play a significant role in tailoring treatment plans as different subtypes respond differently to therapies and medicine.

Table 3.2: Details of TCGA-LUAD dataset.

Omics Type	Sample Size	Feature Size
RNA	517	60660
miRNA	513	3762
Proteome	365	216
Clinical	585	71

- TCGA-BLCA: TCGA-BLCA refers to Bladder Urothelial Carcinoma dataset representing Bladder cancer that affects mainly the lining of bladder. This includes Luminal, Basal, Neuroendocrine molecular subtype.

Table 3.3: Details of TCGA-BLCA dataset.

Omics Type	Sample Size	Feature Size
RNA	406	60660
miRNA	409	3762
Proteome	343	216
Clinical	412	71

- TCGA-COAD: Colon Adenocarcinoma is a malignant tumor which is from epithelial cells of the colon. It is the most common type of colon cancer and the corespondent data resides in TCGA-COAD dataset. Common molecular subtypes of colon adenocarcinoma are Microsatellite Instable Immune, Carnonical, Metabolic, and Mesenchymal.

Table 3.4: Details of TCGA-COAD dataset.

Omics Type	Sample Size	Feature Size
RNA	458	60660
miRNA	444	3762
Proteome	360	216
Clinical	461	70

- TCGA-LIHC: Dataset TCGA-LIHC represents Liver Hepatocellular Carcinoma cancer genome sequencing. It is the most common type of primary liver cancer with molecular subtypes such as TP53, CTNNB1 (β-catenin), and ACIN1.

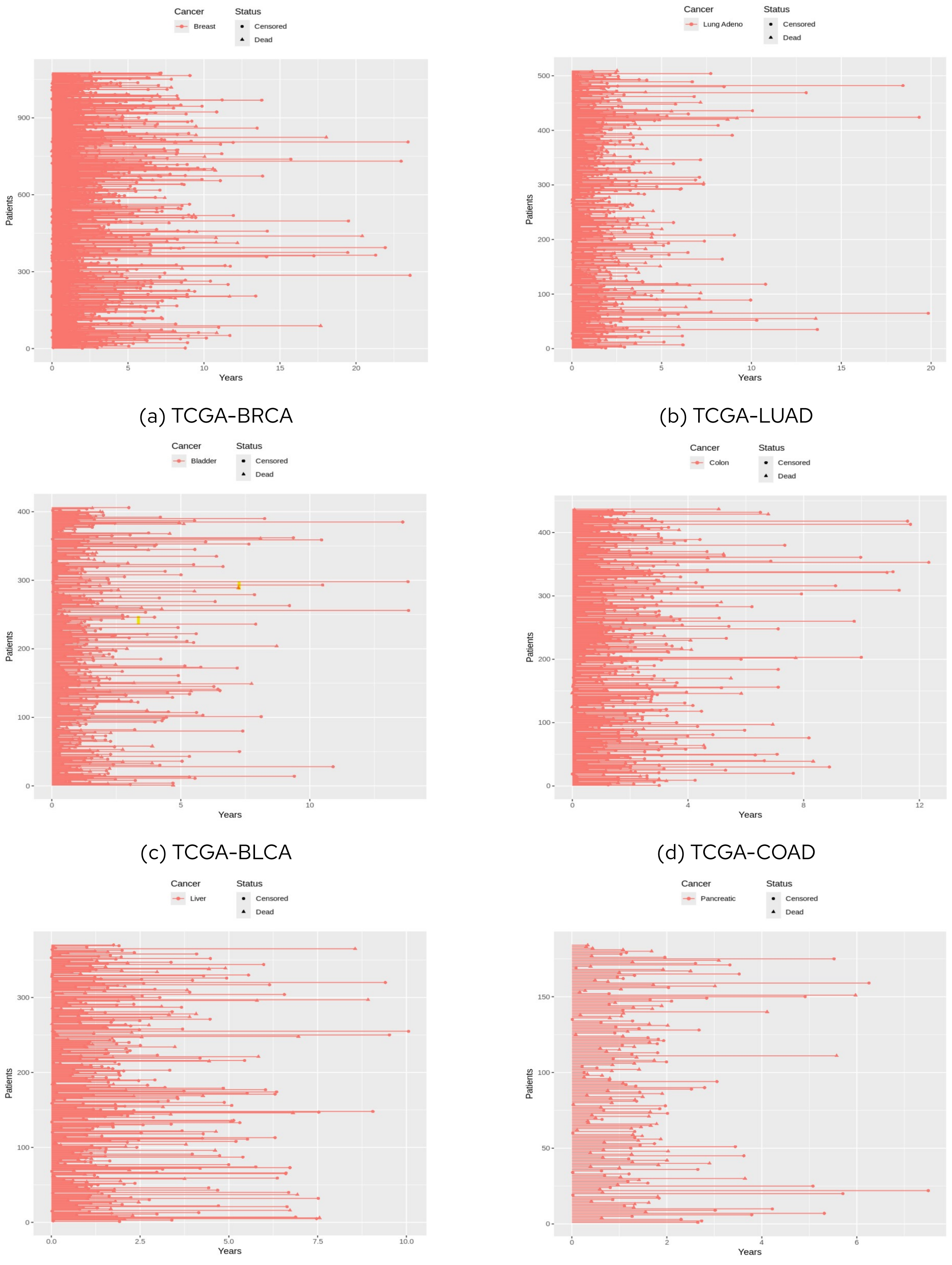
Table 3.5: Details of TCGA-LIHC dataset.

Omics Type	Sample Size	Feature Size
RNA	371	60660
miRNA	373	3762
Proteome	184	458
Clinical	377	69

- TCGA-PAAD: TCGA-PAAD refers to Pancreatic Adenocarcinoma cancer from exocrine pancreas cells that produce digestive enzymes. Common genetic mutations are KRAS, TP53, CDKN2A, SMAD4.

Table 3.6: Details of TCGA-PAAD dataset.

Omics Type	Sample Size	Feature Size
RNA	178	60660
miRNA	178	3762
Proteome	120	217
Clinical	185	70



(a) TCGA-BRCA

(b) TCGA-LUAD

(c) TCGA-BLCA

(d) TCGA-COAD

(e) TCGA-LIHC

(f) TCGA-PAAD

Figure 3.1: Overall survival times and status of six cancer patients from TCGA

4 Experiments and Results

4.1 Environment and Packages

The experiment is carried out on Google Colab Pro with 51GB RAM and 225GB HDD. All code is written and implemented in the R programming language version 4.4.1. There are several packages imported from the Comprehensive R Archive Network (CRAN), the official repository for R that provides R software, packages, and documentation. Utilized packages are listed below:

- BiocManager v1.30.25 [67]: BiocManager is a R package that is designed to access packages from Bioconductor, a repository of bioinformatics that provides packages and tools to access, download, and process Omics data.
- TCGAbiolinks v2.34.0 [68]: TCGAbiolinks aims to retrieve open-access data from TCGA. It simplifies downloading of Omics data such as Gene expression (RNA-Seq), DNA methylation, miRNA expression, Proteome, etc. In addition to that, this package can also help to pre-process the data and carry out some different standard analysis.
- SummarizedExperiment v1.36.0 [69]: Provides one or more assays to represent a matrix-like object where rows represent genomic ranges and columns represent samples. It has three main components, such as assays to hold the main experimental data, rowData to hold metadata about each feature, and colData to provide metadata for each sample like clinical data, treatment information.
- DESeq2 v1.46.0 [70]: Package that counts data from assays often generated from RNA-Seq to identify and normalize differentially expressed genes.
- M3C v1.28.0 [71]: Tool to clustering of high-dimensional data using the Monte Carlo method. It provides ”featurefilter” method for feature selection by variance.
- survival v3.7.0 [72]: Contains core survival analysis methods such as Kaplan-Meier curves, CoxPH to implement and evaluate.
- randomForestSRC v3.3.1 [73]: An R package to implement random forests for survival analysis. It extends classic random forest to handle censored data, making it useful in time-to-event data and survival data.

4.2 Experimental Design

For this thesis experiment, RNA, miRNA, and Proteome, three types of omics data are used. Data is downloaded using TCGAbiolinks package and then converted to matrix using SummarizedExperiment package. As it is recommended to use TMM or DESeq2 normalization for RNA-Seq data prior to further analysis [74], RNA data is normalized using package DESeq2. In addition to that, every omics dataset is pre-processed to remove any missing value, and convert the patient id into a common representation so that the integration of multiple omics data can leverage the patient id to match a patient with their appropriate omics data.

Including non-genetic clinical features such as age, sex, ethnicity, race can improve prediction of survival models [10]. In TCGA cohort, clinical dataset typically has 69 to 70 features. Not all of them are useful as they mostly contain administrative information such as diagnosis_id, treatments_pharmaceutical_treatment_id or contain NA value depending on the cancer type. 10 features are selected from the clinical data that are common to all cancer types.

Mixed integration makes the multi-omics data less interpretable, Hierarchical integration requires prior knowledge and Late integration can technically be called single-omics analysis, we decided to integrate different omics types using Early integration and Intermediate integration. Using Early integration, two multi-omics datasets are created, once with all three omic types and another with only RNA and miRNA omic type. From now on both datasets will be referred to as ”All 3 Omics” and ”RNA and miRNA” respectively. Variance filter is applied to All 3 Omics multi-omics dataset to select 0.2% of the features that is around 130 features except TCGA-LIHC and the dataset with selected feature subset will be referred as ”EI Subset”. It should be noted that the resulting EI Subset may not contain the same number of features from the three omics dataset.

To make a multi-omics dataset that contains roughly the same number of features from every omics dataset, we apply variance filter to every omics dataset and then integrate them using Intermediate integration. We construct two subsets with all three omics data where one subset contains around 130 features with around 43 features from each omics data, and another subset contains around 70 features with around 23 features from each omics data. These two subsets of multi-omics dataset will be referred to as ”II Subset 1” and ”II Subset 2” respectively. Finally, we apply CoxPH and RSF survival analysis models to every multi-omics and subset of multi-omics datasets.

To assess the performance, a 10-fold cross validation strategy is used where each dataset is randomly split into 10 subsets (folds). 1 subset is used for testing and all the other 9 subsets are used for training. This process is done for every subset and their performance is averaged over all 10 folds in the end [75]. The performance is measured with C-index and IBS as well as their confidence interval. Total time for training and testing for each run is also recorded in seconds.

4.3 Results

In this section, we present the performance measure of both models in terms of C-index, IBS, and completion time. Results are presented separately for each TCGA datasets.

Although different omics data have different number of samples, integrating them based on patient id resulted 846 samples for All 3 Omics with 120 events (death) and 1055 samples for RNA and miRNA with 147 events for BRCA dataset. Table 4.1 and table 4.2 show the performance of CoxPH and RSF with respect to their C-index, confidence interval of C-index, IBS, confidence interval of IBS, and completion time in second.

Table 4.1: CoxPH performance on TCGA-BRCA dataset.

Type	C-index	Confidence Interval	IBS	Confidence Interval	Time (S)
All 3 Omics	0.4482	[0.3963, 0.5001]	0.1259	[0.0712, 0.1805]	27
II Subset 1	0.4209	[0.3497, 0.4921]	0.1244	[0.0628, 0.186]	2
II Subset 2	0.4209	[0.3497, 0.4921]	0.1244	[0.0628, 0.186]	2
EI Subset	0.4209	[0.3497, 0.4921]	0.1244	[0.0628, 0.186]	2
RNA and miRNA	0.4686	[0.4295, 0.5078]	0.1043	[0.064, 0.1445]	30

Table 4.2: RSF performance on TCGA-BRCA dataset.

Type	C-index	Confidence Interval	IBS	Confidence Interval	Time (S)
All 3 Omics	0.4637	[0.4102, 0.5172]	0.1224	[0.0716, 0.1731]	183
II Subset 1	0.457	[0.3906, 0.5235]	0.1327	[0.0724, 0.1929]	5
II Subset 2	0.456	[0.3918, 0.5201]	0.1287	[0.0772, 0.1803]	4
EI Subset	0.4614	[0.3915, 0.5313]	0.1284	[0.0771, 0.1798]	5
RNA and miRNA	0.4826	[0.444, 0.5213]	0.0951	[0.068, 0.1222]	154

TCGA-LUAD has 342 samples with 137 events common for all 3 omics data types and 488 samples with 177 events for RNA and miRNA omics types. Table 4.3 and table 4.4 show the performance of CoxPH and RSF with respect to their C-index, confidence interval of C-index, IBS, confidence interval of IBS, and completion time in second for TCGA-LUAD.

Table 4.3: CoxPH performance on TCGA-LUAD dataset.

Type	C-index	Confidence Interval	IBS	Confidence Interval	Time (S)
All 3 Omics	0.5415	[0.4842, 0.5988]	0.0499	[0.0303, 0.0695]	31
II Subset 1	0.491	[0.4454, 0.5366]	0.0524	[0.0292, 0.0756]	2
II Subset 2	0.491	[0.4454, 0.5366]	0.0524	[0.0292, 0.0756]	2
EI Subset	0.491	[0.4454, 0.5366]	0.0524	[0.0292, 0.0756]	2
RNA and miRNA	0.5293	[0.4606, 0.598]	0.0531	[0.0363, 0.0699]	25

Table 4.4: RSF performance on TCGA-LUAD dataset.

Type	C-index	Confidence Interval	IBS	Confidence Interval	Time (S)
All 3 Omics	0.492	[0.442, 0.542]	0.0491	[0.0243, 0.0738]	132
II Subset 1	0.5258	[0.4812, 0.5704]	0.0487	[0.0241, 0.0733]	4
II Subset 2	0.501	[0.4539, 0.548]	0.0489	[0.0248, 0.0731]	3
EI Subset	0.499	[0.4503, 0.5478]	0.0461	[0.0273, 0.0649]	4
RNA and miRNA	0.4527	[0.3924, 0.513]	0.053	[0.0345, 0.0714]	128

In TCGA-BLCA data, we get 332 samples with 150 events and 397 samples with 174 events for all 3 omics types and RNA and miRNA omics types respectively. Table 4.5 and table 4.6 show the performance of CoxPH and RSF with respect to their C-index, confidence interval of C-index, IBS, confidence interval of IBS, and completion time in second for TCGA-BLCA data.

Table 4.5: CoxPH performance on TCGA-BLCA dataset.

Type	C-index	Confidence Interval	IBS	Confidence Interval	Time (S)
All 3 Omics	0.5037	[0.4542, 0.5533]	0.0612	[0.0458, 0.0766]	25
II Subset 1	0.4286	[0.3858, 0.4714]	0.0647	[0.0473, 0.0821]	2
II Subset 2	0.4286	[0.3858, 0.4714]	0.0647	[0.0473, 0.0821]	2
EI Subset	0.4286	[0.3858, 0.4714]	0.0647	[0.0473, 0.0821]	2
RNA and miRNA	0.4918	[0.4395, 0.5442]	0.0645	[0.0521, 0.0768]	30

Table 4.6: RSF performance on TCGA-BLCA dataset.

Type	C-index	Confidence Interval	IBS	Confidence Interval	Time (S)
All 3 Omics	0.5133	[0.4525, 0.5741]	0.0667	[0.0517, 0.0817]	117
II Subset 1	0.4264	[0.379, 0.4738]	0.0713	[0.0516, 0.091]	3
II Subset 2	0.4211	[0.3724, 0.4699]	0.0715	[0.0515, 0.0915]	3
EI Subset	0.4243	[0.3781, 0.4704]	0.0718	[0.052, 0.0916]	3
RNA and miRNA	0.4785	[0.4376, 0.5193]	0.0641	[0.0478, 0.0804]	128

TCGA-COAD has 325 samples common for all 3 omics types with 70 events and 419 samples common for RNA and miRNA omics types with 94 events. Table 4.7 and table 4.8 show the performance of CoxPH and RSF for TCGA-COAD data with respect to their C-index, confidence interval of C-index, IBS, confidence interval of IBS, and completion time in second.

Table 4.7: CoxPH performance on TCGA-COAD dataset.

Type	C-index	Confidence Interval	IBS	Confidence Interval	Time (S)
All 3 Omics	0.4528	[0.3808, 0.5247]	0.1354	[0.095, 0.1757]	22
II Subset 1	0.5917	[0.5355, 0.6479]	0.1176	[0.0802, 0.1549]	2
II Subset 2	0.5917	[0.5355, 0.6479]	0.1176	[0.0802, 0.1549]	2
EI Subset	0.5917	[0.5355, 0.6479]	0.1176	[0.0802, 0.1549]	2
RNA and miRNA	0.46	[0.4234, 0.4965]	0.1316	[0.0856, 0.1777]	28

Table 4.8: RSF performance on TCGA-COAD dataset.

Type	C-index	Confidence Interval	IBS	Confidence Interval	Time (S)
All 3 Omics	0.4857	[0.4085, 0.563]	0.1205	[0.0894, 0.1517]	96
II Subset 1	0.5776	[0.4831, 0.6721]	0.097	[0.0703, 0.1237]	3
II Subset 2	0.5711	[0.472, 0.6702]	0.0976	[0.0703, 0.1248]	2
EI Subset	0.5767	[0.4782, 0.6752]	0.0974	[0.0702, 0.1246]	2
RNA and miRNA	0.4464	[0.372, 0.5208]	0.1116	[0.0681, 0.155]	110

TCGA-LIHC data has 175 samples with 90 events and 361 samples with 128 events for all 3 types omics data and RNA and miRNA data respectively. Table 4.9 and table 4.10 show the performance of CoxPH and RSF with respect to their C-index, confidence interval of C-index, IBS, confidence interval of IBS, and completion time in second for TCGA-LIHC data.

Table 4.9: CoxPH performance on TCGA-LIHC dataset.

Type	C-index	Confidence Interval	IBS	Confidence Interval	Time (S)
All 3 Omics	0.4862	[0.4285, 0.5439]	0.062	[0.0283, 0.0957]	22
II Subset 1	0.4804	[0.4227, 0.5381]	0.0674	[0.0337, 0.1011]	2
II Subset 2	0.4804	[0.4227, 0.5381]	0.0674	[0.0337, 0.1011]	2
EI Subset	0.4804	[0.4227, 0.5381]	0.0674	[0.0337, 0.1011]	2
RNA and miRNA	0.4968	[0.4704, 0.5232]	0.0725	[0.0484, 0.0966]	24

Table 4.10: RSF performance on TCGA-LIHC dataset.

Type	C-index	Confidence Interval	IBS	Confidence Interval	Time (S)
All 3 Omics	0.482	[0.4229, 0.5411]	0.0628	[0.0277, 0.0979]	84
II Subset 1	0.4858	[0.434, 0.5377]	0.0625	[0.0278, 0.0973]	3
II Subset 2	0.4783	[0.419, 0.5375]	0.0624	[0.028, 0.0968]	2
EI Subset	0.4849	[0.4214, 0.5484]	0.0625	[0.0279, 0.0972]	3
RNA and miRNA	0.5011	[0.471, 0.5312]	0.0665	[0.0424, 0.0907]	108

TCGA-PAAD has 113 samples common for all 3 omics types with 63 events and 177 samples common for RNA and miRNA omics types with 93 events. Table 4.11 and table 4.12 show the performance of CoxPH and RSF with respect to their C-index, confidence interval of C-index, IBS, confidence interval of IBS, and completion time in second for TCGA-PAAD.

Table 4.11: CoxPH performance on TCGA-PAAD dataset.

Type	C-index	Confidence Interval	IBS	Confidence Interval	Time (S)
All 3 Omics	0.4705	[0.3879, 0.553]	0.0774	[0.0392, 0.1155]	25
II Subset 1	0.4807	[0.3981, 0.5633]	0.081	[0.0428, 0.1192]	2
II Subset 2	0.4807	[0.3981, 0.5633]	0.081	[0.0428, 0.1192]	2
EI Subset	0.4807	[0.3981, 0.5633]	0.081	[0.0428, 0.1192]	2
RNA and miRNA	0.4564	[0.4125, 0.5004]	0.0936	[0.0348, 0.1524]	24

Table 4.12: RSF performance on TCGA-PAAD dataset.

Type	C-index	Confidence Interval	IBS	Confidence Interval	Time (S)
All 3 Omics	0.4741	[0.3924, 0.5557]	0.0861	[0.0496, 0.1226]	83
II Subset 1	0.4804	[0.399, 0.5619]	0.0861	[0.0502, 0.122]	3
II Subset 2	0.4809	[0.3974, 0.5643]	0.0861	[0.0498, 0.1224]	2
EI Subset	0.4659	[0.3829, 0.5489]	0.0862	[0.0503, 0.1222]	3
RNA and miRNA	0.4595	[0.4182, 0.5008]	0.0867	[0.0316, 0.1418]	88

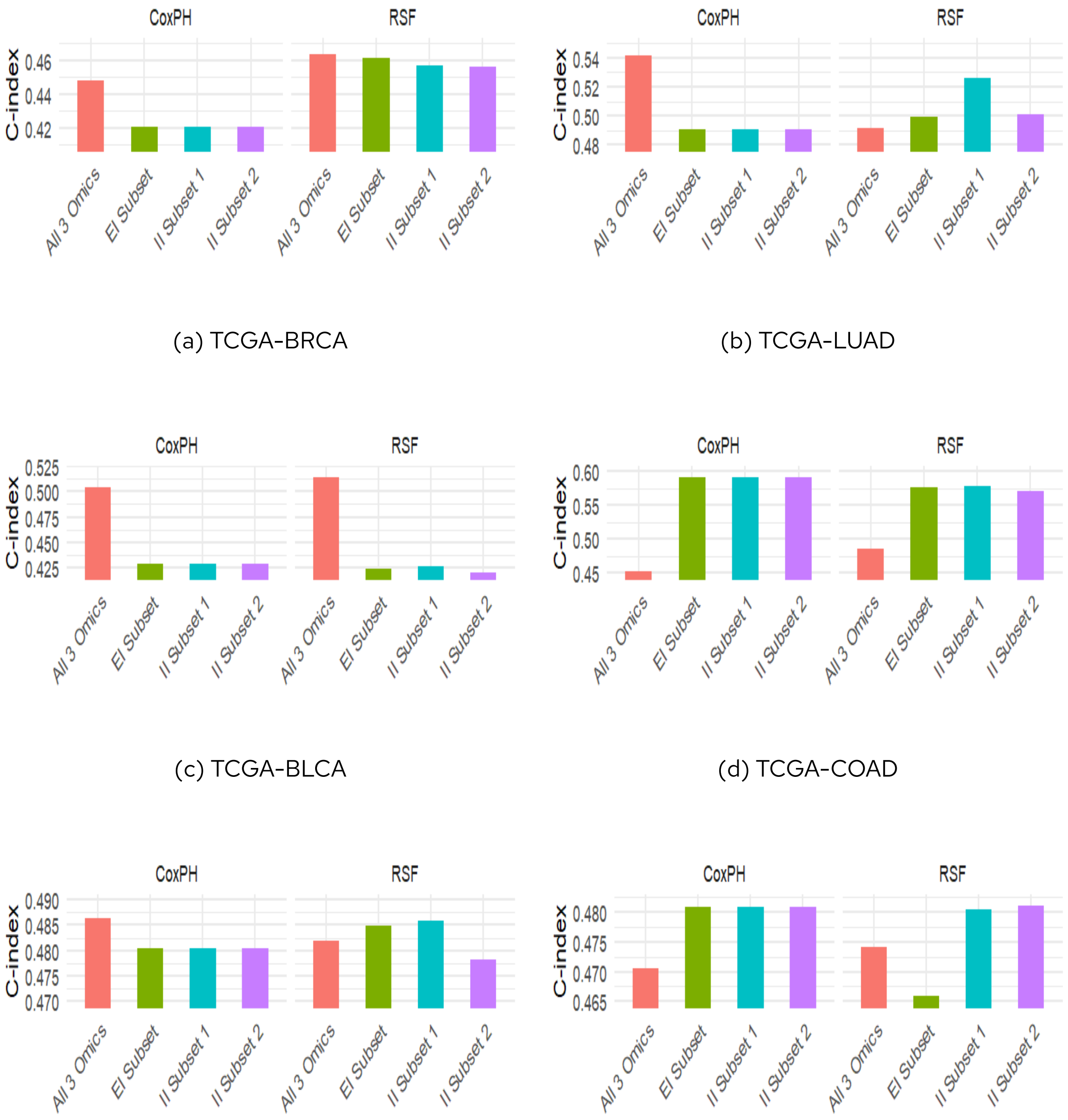
5 Discussion

5.1 Performance Based on Feature Selection

Models predictive performance is measured in both C-index and IBS. First, we look at the predictive performance in C-index for both model across all 6 multi-omics datasets with and without feature selection to discuss about the effect of feature selection on multi-omics data and Figure 5.1 shows models' performance in C-index for different feature selections. On BRCA dataset, both the CoxPH and RSF models achieve higher C-index when no features are selected. RSF achieves higher C-index for LUAD data when feature selection is applied for both EI and II multi-omics datasets. CoxPH does not perform better when features are selected for the same dataset. Similar to BRCA, both models perform poorly for feature selected subsets of BLCA data. CoxPH do not perform well for feature selection on LIHC data, while RSF does except II Subset 2 dataset. For COAD and PAAD datasets, both models perform better when features are selected prior to survival analysis. However, RSF EI Subset do not perform well compared to both II subsets.

Now we look at IBS performance for both models where a lower IBS score means better performance. CoxPH IBS is lower for all 3 feature selected subsets for BRCA, while RSF IBS is higher. For LUAD dataset, we can see the different scenario where RSF IBS is lower for all 3 feature selected subsets, but CoxPH IBS is higher. On the other hand, both models fail to achieve a lower IBS on all 3 feature selected subsets for BLCA data. Both models' performance on COAD data are quite opposite than the performance on BLCA data. For COAD, both models achieve a lower IBS score for all 3 subsets. On the other hand, only RSF performs better in terms of IBS for all 3 subsets of LIHC than original dataset, while CoxPH does not outperform the prediction of the original dataset for the subsets. Lastly, both models' performance are superior on the original dataset than all 3 feature selected subsets for PAAD data. Models performance in IBS for different features selection subsets and the original multi-omics dataset can be observed in Figure 5.2.

With respect to our first thesis question, we can see that feature selected subsets do not always perform better than the original multi-omics dataset, nor the original multi-omics dataset always performs better than the feature selected subsets. Mostly the survival prediction performance depends on the selected dataset, analysis model, and performance measure metric.



(a) TCGA-BRCA

(b) TCGA-LUAD

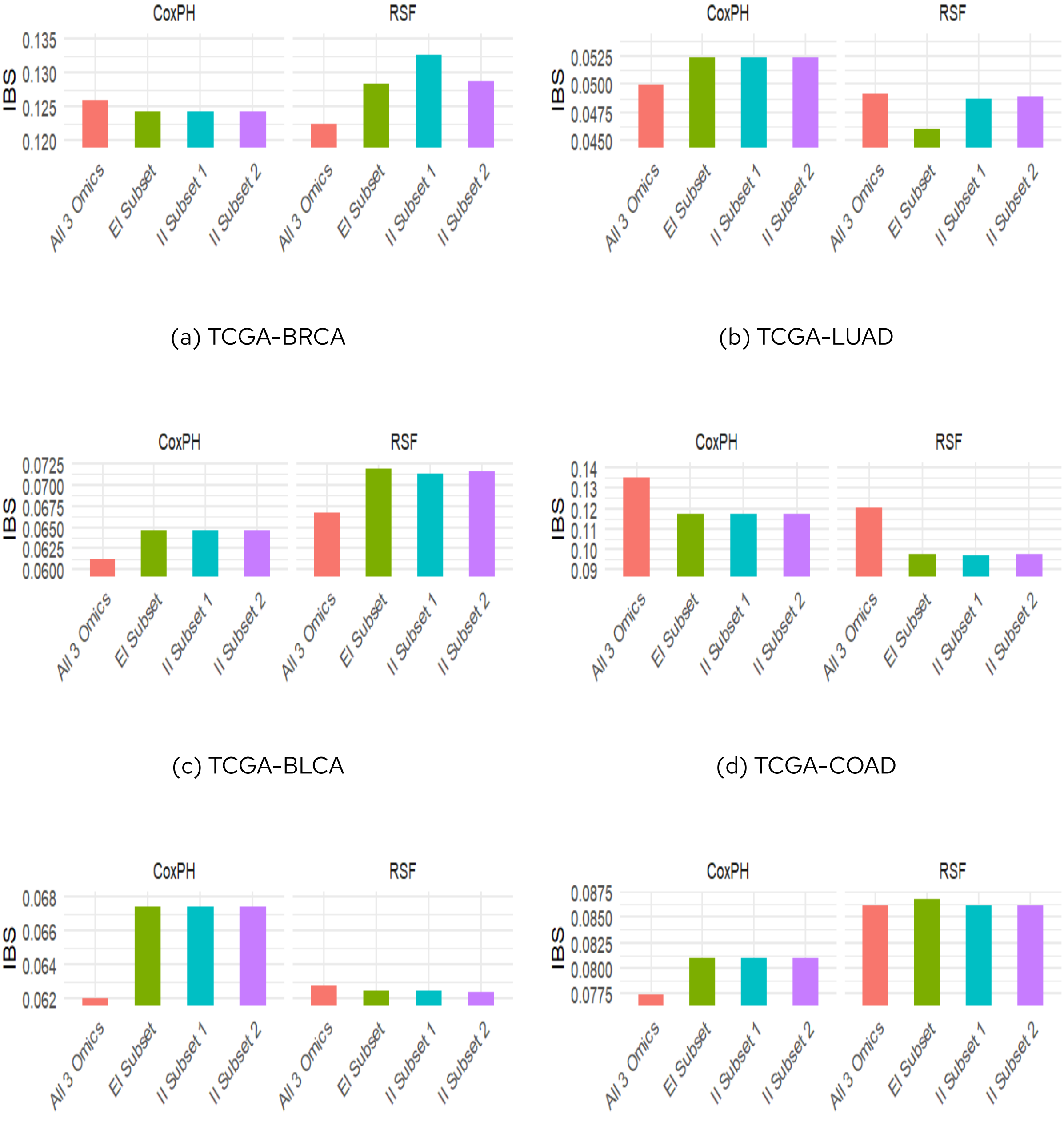
(c) TCGA-BLCA

(d) TCGA-COAD

(e) TCGA-LIHC

(f) TCGA-PAAD

Figure 5.1: Models performance in C-index for different features selection.



(a) TCGA-BRCA

(b) TCGA-LUAD

(c) TCGA-BLCA

(d) TCGA-COAD

(e) TCGA-LIHC

(f) TCGA-PAAD

Figure 5.2: Models performance in IBS for different features selection.

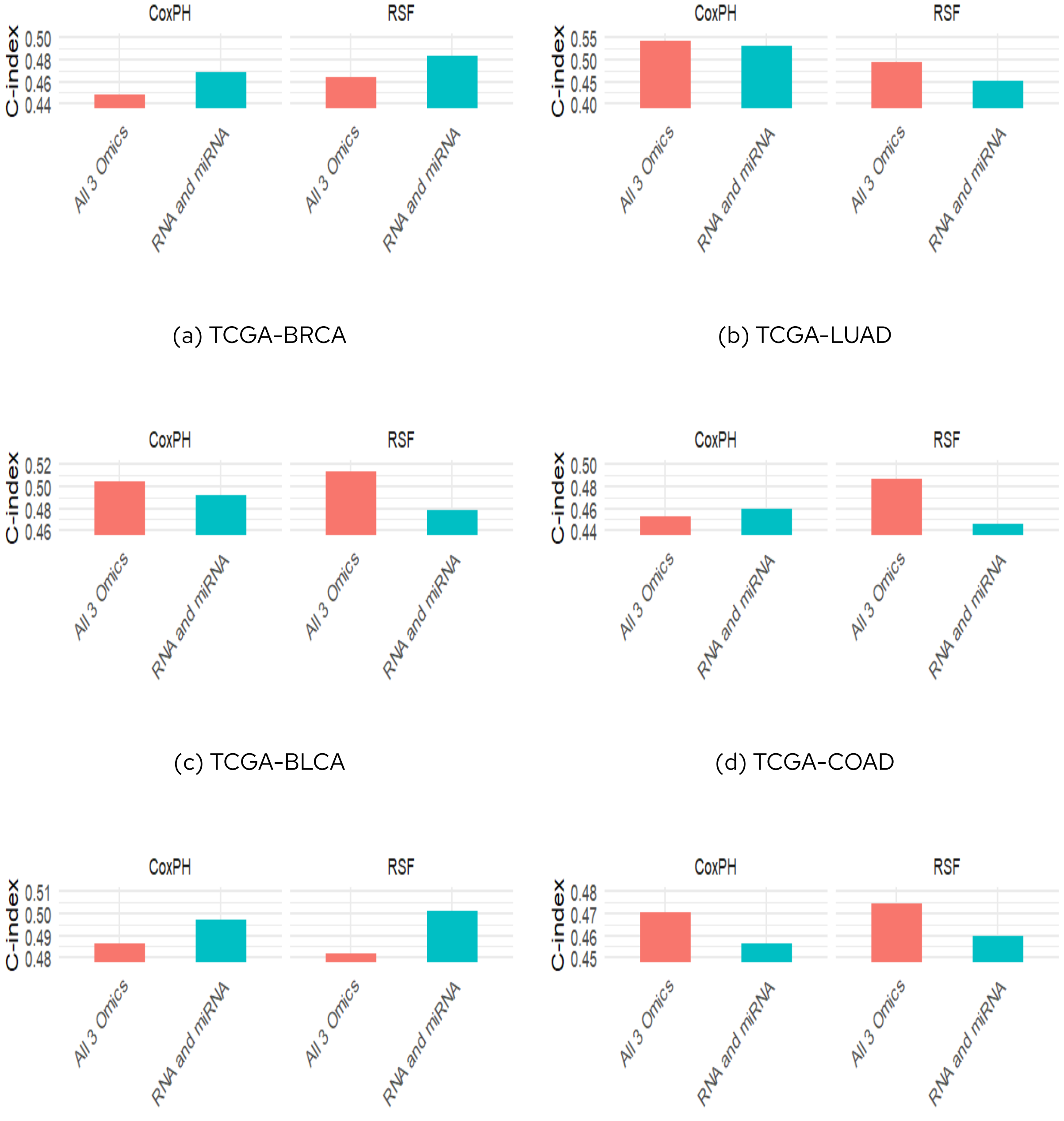
5.2 Performance Based on Omics Selection

There are many omics types such as gene expression (RNA), miRNA expression, copy number segment, protein expression, and DNA methylation available for each TCGA cohorts for integration to construct multi-omics data. Because of very high dimensionality nature of omics data, it can be quite overwhelming for any model to successfully analyze when all available omics types are integrated to make a multi-omics data. We construct two multi-omics datasets by integrating 3 (RNA, miRNA, Proteome) omics type and 2 (RNA, miRNA) omics type to find out the performance of different combinations of omics within the same cohorts. As before, first we discuss models performance in terms of C-index and then in terms of IBS.

Both models achieve higher C-index for RNA and miRNA multi-omics data than RNA, miRNA and Proteome for BRCA data. Unlike BRCA data, CoxPH and RSF both have higher C-index for RNA, miRNA and Proteome multi-omics data than RNA and miRNA multi-omics data. For COAD data, only CoxPH has higher C-index for RNA and miRNA data than RNA, miRNA and Proteome data, while RSF output for RNA and miRNA data is lower than RNA, miRNA and Proteome data. Both models achieve higher C-index for RNA and miRNA multi-omics data than RNA, miRNA and Proteome multi-omics data of LIHC. On the other hand, both models have lower C-index for RNA and miRNA multi-omics data than RNA, miRNA and Proteome multi-omics data of PAAD. The performance of the models in C-index for different omics selection is presented in Figure 5.3.

Looking at IBS we see that CoxPH and RSF both perform well for RNA and miRNA multi-omics data than RNA, miRNA and Proteome multi-omics data on BRCA cohorts. Unlike BRCA, both models perform poorly for RNA and miRNA multi-omics than RNA, miRNA and Proteome on LUAD data. While RSF IBS is lower for RNA and miRNA multi-omics on BLCA data, CoxPH IBS is higher for RNA, miRNA and Proteome multi-omics. Similar to BRCA, both models achieve lower IBS for RNA and miRNA multi-omics than RNA, miRNA and Proteome multi-omics dataset on COAD data. For both LIHC and PAAD data, CoxPH and RSF both achieve lower IBS for RNA, miRNA and Proteome multi-omics data than RNA and miRNA multi-omics data. Models performance in IBS for RNA, miRNA and Proteome multi-omics data and RNA and miRNA multi-omics data can be observed in Figure 5.4.

With respect to our second thesis question, we can see that the effects of selecting different omics types vary among dataset and models, where sometimes selecting few omics types perform better than selecting more omics types and sometimes the other way around.



(a) TCGA-BRCA

(b) TCGA-LUAD

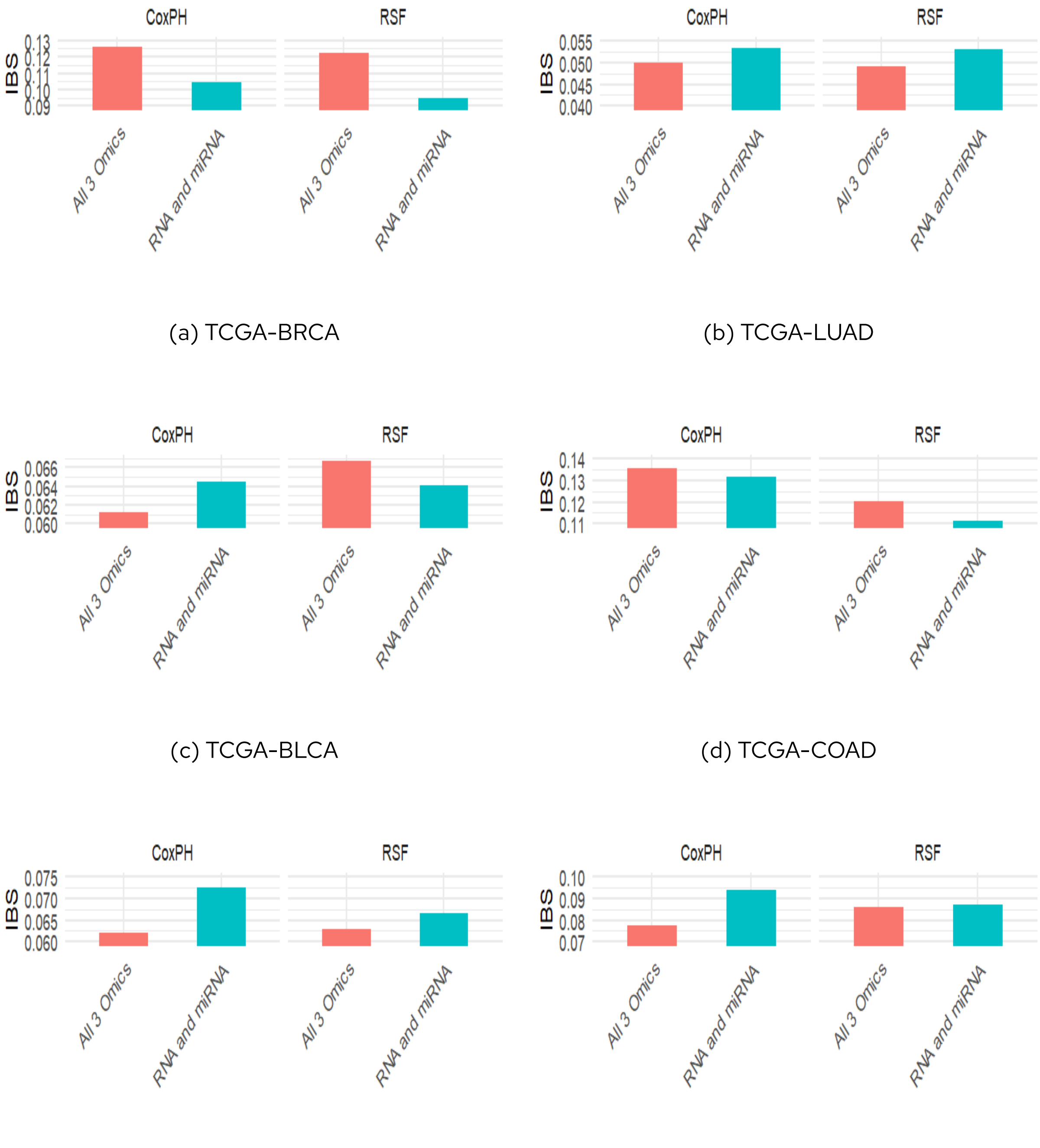
(c) TCGA-BLCA

(d) TCGA-COAD

(e) TCGA-LIHC

(f) TCGA-PAAD

Figure 5.3: Models performance in C-index for different omics selection.



(a) TCGA-BRCA

(b) TCGA-LUAD

(c) TCGA-BLCA

(d) TCGA-COAD

(e) TCGA-LIHC

(f) TCGA-PAAD

Figure 5.4: Models performance in IBS for different omics selection.

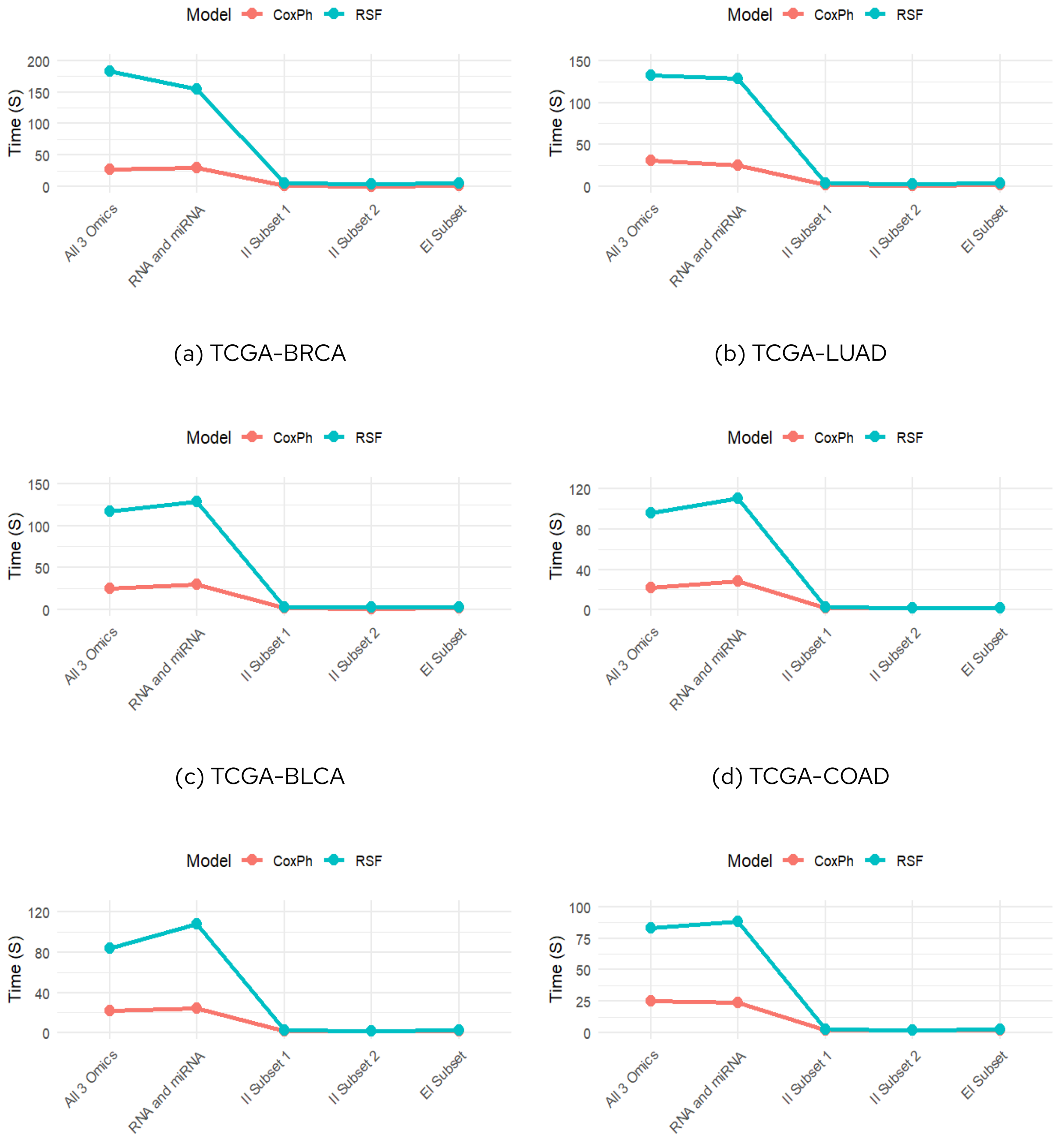
5.3 Computation Time

The models' computation times are measured in seconds based on the time required for models to train and test. Of course, the computation times heavily depend on the sample and features size of multi-omics datasets. In general, both models require huge amounts of time for full RNA, miRNA and Proteome multi-omics dataset compared to the feature selected subsets.

CoxPH takes roughly 11 to 15 times more time for RNA, miRNA and Proteome multi-omics dataset than the subsets. On the other hand, RSF takes around 33 to 39 times more time for RNA, miRNA and Proteome multi-omics dataset than the subsets.

Moreover, require time for both models running on RNA, miRNA and Proteome multi-omics dataset and RNA and miRNA multi-omics dataset fluctuate among them and cannot be determined whether RNA, miRNA and Proteome multi-omics will take more time or RNA and miRNA multi-omics will take more time for the same TCGA cohorts. One reason for this fluctuation may be that integrating more omics type increases the number of features, but can decrease the number of samples and integrating less omics type decreases the number of features, but can increase the number of samples.

Last but not least, RSF for multi-omics original dataset takes 4 to 6 times more time than CoxPH. Figure 5.5 shows computation time required for each model for full multi-omics dataset and subsets of the full multi-omics dataset of different TCGA cohorts.



(a) TCGA-BRCA

(b) TCGA-LUAD

(c) TCGA-BLCA

(d) TCGA-COAD

(e) TCGA-LIHC

(f) TCGA-PAAD

Figure 5.5: Models performance in computation time.

6 Conclusion

This thesis explores the survival analysis in multi-omics settings with the aim to find out the effect of feature selection on multi-omics data for further survival analysis and whether integrating more available omics type to build a multi-omics dataset is a better approach than carefully selecting few from available omics type to build a multi-omics dataset and vice versa.

To answer the first question of the thesis, this study evaluate six dataset from TCGA using two survival analysis methods such as CoxPH, a state-of-the-art classical survival model, and RSF, a state-of-the-art machine learning survival model [37]. RNA, miRNA and Proteome omics type are chosen from every TCGA dataset and integrate them using early integration (EI) and intermediate integration (II) techniques from five multi-omics integration techniques as other techniques either transform the original data to make a multi-omics dataset or analyze single omics to later integrate for multi-omics that cannot technically be called multi-omics dataset [5]. Variance filter method is adopted to select features to make three subsets of the full multi-omics dataset because variance filter method is proved to be affective when selecting features of omics data for independently analyze using survival models [10].

Among the three subsets, two are constructed when II is applied with different number of features such as 130 and 70. To make the subset constructed when EI is applied comparable to the subset constructed when II is applied, the features of the subset of EI are kept around 130 as well. Applying both survival models to the full multi-omics dataset and it's three subsets reveal the effect of feature selection on multi-omics data for further survival analysis.

Two of the six selected TCGA data perform better when feature selection is applied in terms of C-index for the CoxPH model. Additionally, in terms of IBS two TCGA data performs better for the subsets compared to the full multi-omics data when CoxPH is applied. Only one TCGA data are common for both C-index and IBS and other two data either performed well in terms of C-index or IBS. On the other hand, four among six TCGA data perform better on feature selected subsets in terms of C-index when RSF is applied with the exception of II Subset 2 of BLCA and EI Subset of COAD. In terms of IBS, three TCGA data perform better on all subsets than the full multi-omics dataset. Although, one more dataset, COAD produce the same IBS for II Subset 2 and II Subset 2. It is to be noted that, three of the performant TCGA data in terms of IBS are also among the four data in terms of C-index which indicates that RSF is more robust than CoxPH when dealing with multi-omics data.

Survival analysis is also done with RNA and miRNA multi-omics data to compare it with RNA, miRNA and Proteome multi-omics data to observe and answer the second thesis question regarding the effect of multi-omics data integrated with different omics type.

Here, three of the six TCGA data performed better in terms of C-index and two of the six TCGA data performed better in terms of IBS for RNA and miRNA multi-omics data compared to its counterpart RNA, miRNA and Proteome multi-omics data when survival analysis is done with CoxPH. Two of the TCGA data are found in both C-index and IBS evaluation measure metric. Additionally, when RSF model is applied, two of the six TCGA data are performant in terms of C-index and three of the six are performant in terms of IBS for RNA and miRNA multi-omics than RNA, miRNA and Proteome multi-omics data. One of the data is common for both C-index and IBS, while rest of them are for either C-index or IBS.

In addition to the above output, it can be observed that CoxPH takes roughly 11 to 15 times more time for RNA, miRNA and Proteome multi-omics dataset than the subsets and RSF takes around 33 to 39 times more time for RNA, miRNA and Proteome multi-omics dataset than the subsets. Moreover, require time for both models running on RNA, miRNA and Proteome multi-omics dataset and RNA and miRNA multi-omics dataset fluctuate among them.

It is evident that survival analysis prediction performance heavily depends on the chosen multi-omics dataset, integration technique of creating multi-omics dataset, selected survival analysis model and it's configuration and the considered prediction performance measure [7]. Our study also supports this claim as it cannot definitely be said whether or not feature selected subset perform better than the full multi-omics dataset and whether or not integrating more available omics type to build a multi-omics dataset is a better approach than carefully selecting few from available omics type to build a multi-omics dataset.

In general, survival performance depends on the selected dataset, analysis model and performance measure metric. As it is observed that analysis of smaller feature selected subsets takes significantly less time than analysis of full multi-omics dataset irrespective of chosen model and the performance difference for both settings are not that statistically significant, researchers can apply feature selection before applying survival model when lower computation time is the primary concern. On the other hand, if the prediction performance is the primary goal, then researchers are advised to try with and without feature selection to see which works better for their chosen dataset and survival models.

Unlike feature selection, it is not definitive of computation time to confirm whether selecting more or less omics type from all available omics type to construct multi-omics dataset takes more or less time to finish the survival analysis as selecting different omics type can influence the computation time in both negative and positive way depending on both the number of sample and the number of features. Here, researchers are advised to try out different settings for integrating omics to construct multi-omics dataset depending on the dataset and survival model and see which one works best for them.

By providing comprehensive answers through rigorous analysis and evidence to the research questions the thesis set out to address, we can conclude that it has successfully achieved its goal. One limitation of this study is that only two survival models are implemented on six TCGA dataset. In future, more dataset from TCGA, NCI Genomics Data Commons (GDC) and the Gene Expression Omnibus (GEO) can be included with other survival analysis models. Moreover, different numbers of features can be considered while building the multi-omics subsets and different type of omics can be included to observe the different predictive performance based on selected omics types.

bibliography

[1] Y. Hasin, M. Seldin, and A. Lusis, "Multi-omics approaches to disease," *Genome biology*, vol. 18, pp. 1–15, 2017.

[2] A. F. Abbasi, M. N. Asim, S. Ahmed, S. Vollmer, and A. Dengel, "Survival prediction landscape: An in-depth systematic literature review on activities, methods, tools, diseases, and databases," *medRxiv*, pp. 2024–01, 2024.

[3] A.-L. Boulesteix and W. Sauerbrei, "Added predictive value of high-throughput molecular data to clinical data and its validation," *Briefings in bioinformatics*, vol. 12, no. 3, pp. 215–229, 2011.

[4] M. Herrmann, "Large-scale benchmark study of prediction methods using multi-omics data," Ph.D. dissertation, 2019.

[5] M. Picard, M.-P. Scott-Boyer, A. Bodein, O. Périn, and A. Droit, "Integration strategies of multi-omics data for machine learning analysis," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 3735–3746, 2021.

[6] R. Couronné, P. Probst, and A.-L. Boulesteix, "Random forest versus logistic regression: a large-scale benchmark experiment," *BMC bioinformatics*, vol. 19, pp. 1–14, 2018.

[7] M. Herrmann, P. Probst, R. Hornung, V. Jurinovic, and A.-L. Boulesteix, "Large-scale benchmark study of survival prediction methods using multi-omics data," *Briefings in bioinformatics*, vol. 22, no. 3, p. bbaa167, 2021.

[8] G. L. Libralon, A. C. P. d. L. F. de Carvalho, and A. C. Lorena, "Pre-processing for noise detection in gene expression classification data," *Journal of the Brazilian Computer Society*, vol. 15, pp. 3–11, 2009.

[9] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[10] A. Bommert, T. Welchowski, M. Schmid, and J. Rahnenführer, "Benchmark of filter methods for feature selection in high-dimensional gene expression survival data," *Briefings in Bioinformatics*, vol. 23, no. 1, p. bbab354, 2022.

[11] K. Tomczak, P. Czerwińska, and M. Wiznerowicz, "Review the cancer genome atlas (tcga): an immeasurable source of knowledge," *Contemporary Oncology/Współczesna Onkologia*, vol. 2015, no. 1, pp. 68–77, 2015.

[12] B. B. Misra, C. Langefeld, M. Olivier, and L. A. Cox, "Integrated omics: tools, advances and future approaches," *Journal of molecular endocrinology*, vol. 62, no. 1, pp. R21–R45, 2019.

[13] I. Gaynanova and G. Li, "Structural learning and integrative decomposition of multi-view data," *Biometrics*, vol. 75, no. 4, pp. 1121–1132, 2019.

[14] Y. El-Manzalawy, T.-Y. Hsieh, M. Shivakumar, D. Kim, and V. Honavar, "Min-redundancy and max-relevance multi-view feature selection for predicting ovarian cancer survival using multi-omics data," *BMC medical genomics*, vol. 11, pp. 19–31, 2018.

[15] T. Bhadra, S. Mallik, N. Hasan, and Z. Zhao, "Comparison of five supervised feature selection algorithms leading to top features and gene signatures from multi-omics data in cancer," *BMC bioinformatics*, vol. 23, no. Suppl 3, p. 153, 2022.

[16] S. K. Pal and P. Mitra, *Pattern recognition algorithms for data mining*. Chapman and Hall/CRC, 2004.

[17] T. Bhadra and S. Bandyopadhyay, "Supervised feature selection using integration of densest subgraph finding with floating forward–backward search," *Information Sciences*, vol. 566, pp. 1–18, 2021.

[18] Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," *bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[19] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[20] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proceedings of the 20th international conference on machine learning (ICML-03)*, 2003, pp. 856–863.

[21] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, The University of Waikato, 1999.

[22] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[23] I. Kononenko, E. Šimec, and M. Robnik-Šikonja, “Overcoming the myopia of inductive learning algorithms with relieff,” *Applied Intelligence*, vol. 7, pp. 39–55, 1997.

[24] A. Unler, A. Murat, and R. B. Chinnam, “mr2pso: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification,” *Information Sciences*, vol. 181, no. 20, pp. 4625–4641, 2011.

[25] T. M. Cover, “The best two independent measurements are not the two best,” *IEEE Transactions on Systems, Man, and Cybernetics*, no. 1, pp. 116–117, 1974.

[26] I. A. Gheyas and L. S. Smith, “Feature subset selection in large dimensionality domains,” *Pattern recognition*, vol. 43, no. 1, pp. 5–13, 2010.

[27] A. Vince, “A framework for the greedy algorithm,” *Discrete Applied Mathematics*, vol. 121, no. 1-3, pp. 247–260, 2002.

[28] M. Rostami and P. Moradi, “A clustering based genetic algorithm for feature selection,” in *2014 6th Conference on Information and Knowledge Technology (IKT)*. IEEE, 2014, pp. 112–116.

[29] B. Xue, M. Zhang, and W. N. Browne, “Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms,” *Applied soft computing*, vol. 18, pp. 261–276, 2014.

[30] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, “A review of feature selection methods on synthetic data,” *Knowledge and information systems*, vol. 34, pp. 483–519, 2013.

[31] R. Tibshirani, “The lasso method for variable selection in the cox model,” *Statistics in medicine*, vol. 16, no. 4, pp. 385–395, 1997.

[32] A.-C. Haury, P. Gestraud, and J.-P. Vert, “The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures,” *PloS one*, vol. 6, no. 12, p. e28210, 2011.

[33] L. Breiman, “Random forests,” *Machine learning*, vol. 45, pp. 5–32, 2001.

[34] V. Bolón-Canedo, N. Sánchez-Marono, A. Alonso-Betanzos, J. M. Benítez, and F. Herrera, "A review of microarray datasets and applied feature selection methods," *Information sciences*, vol. 282, pp. 111–135, 2014.

[35] L. Gidskehaug, E. Anderssen, A. Flatberg, and B. K. Alsberg, "A framework for significance analysis of gene expression data using dimension reduction methods," *BMC bioinformatics*, vol. 8, pp. 1–14, 2007.

[36] Y. Li, U. Mansmann, S. Du, and R. Hornung, "Benchmark study of feature selection strategies for multi-omics data," *BMC bioinformatics*, vol. 23, no. 1, p. 412, 2022.

[37] R. E. B. Sonabend, "A theoretical and methodological framework for machine learning in survival analysis: Enabling transparent and accessible predictive modelling on right-censored time-to-event data," Ph.D. dissertation, UCL (University College London), 2021.

[38] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *Journal of the American statistical association*, vol. 53, no. 282, pp. 457–481, 1958.

[39] W. Nelson, "Theory and applications of hazard plotting for censored failure data," *Technometrics*, vol. 14, no. 4, pp. 945–966, 1972.

[40] O. Aalen, "Nonparametric inference for a family of counting processes," *The Annals of Statistics*, pp. 701–726, 1978.

[41] M. G. Akritas, "Nearest neighbor estimation of a bivariate distribution under random censoring," *The Annals of Statistics*, pp. 1299–1327, 1994.

[42] D. R. Cox, "Regression models and life-tables," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 34, no. 2, pp. 187–202, 1972.

[43] M. LeBlanc and J. Crowley, "Relative risk trees for censored survival data," *Biometrics*, pp. 411–425, 1992.

[44] H. Ishwaran, E. H. Blackstone, C. E. Pothier, and M. S. Lauer, "Relative risk forests for exercise heart rate recovery as a predictor of mortality," *Journal of the American Statistical Association*, vol. 99, no. 467, pp. 591–600, 2004.

[45] T. Hothorn, B. Lausen, A. Benner, and M. Radespiel-Tröger, "Bagging survival trees," *Statistics in medicine*, vol. 23, no. 1, pp. 77–91, 2004.

[46] T. Hothorn, P. Bühlmann, S. Dudoit, A. Molinaro, and M. J. Van Der Laan, "Survival ensembles," *Biostatistics*, vol. 7, no. 3, pp. 355–373, 2006.

[47] H. Ishwaran, U. B. Kogalur, E. H. Blackstone, and M. S. Lauer, "Random survival forests," 2008.

[48] G. Ridgeway, "The state of boosting," *Computing science and statistics*, pp. 172–181, 1999.

[49] H. Binder and M. Schumacher, "Allowing for mandatory covariates in boosting estimation of sparse high-dimensional survival models," *BMC bioinformatics*, vol. 9, pp. 1–10, 2008.

[50] H. Binder, "Coxboost: Cox models by likelihood based boosting for a single survival endpoint or competing risks," *R package version*, vol. 1, no. 4, 2013.

[51] M. Schmid and T. Hothorn, "Flexible boosting of accelerated failure time models," *BMC bioinformatics*, vol. 9, pp. 1–13, 2008.

[52] D. Faraggi and R. Simon, "A neural network model for survival data," *Statistics in medicine*, vol. 14, no. 1, pp. 73–82, 1995.

[53] T. Ching, X. Zhu, and L. X. Garmire, "Cox-nnet: an artificial neural network method for prognosis prediction of high-throughput omics data," *PLoS computational biology*, vol. 14, no. 4, p. e1006076, 2018.

[54] J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, and Y. Kluger, "Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network," *BMC medical research methodology*, vol. 18, pp. 1–12, 2018.

[55] L. Zhao and D. Feng, "Deep neural networks for survival analysis using pseudo values," *IEEE journal of biomedical and health informatics*, vol. 24, no. 11, pp. 3308–3314, 2020.

[56] C. Lee, W. Zame, J. Yoon, and M. Van Der Schaar, "Deephit: A deep learning approach to survival analysis with competing risks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[57] H. Kvamme, Ø. Borgan, and I. Scheel, "Time-to-event prediction with neural networks and cox regression," *Journal of machine learning research*, vol. 20, no. 129, pp. 1–30, 2019.

[58] B. Jing, T. Zhang, Z. Wang, Y. Jin, K. Liu, W. Qiu, L. Ke, Y. Sun, C. He, D. Hou *et al.*, "A deep survival analysis method based on ranking," *Artificial intelligence in medicine*, vol. 98, pp. 1–9, 2019.

[59] L. Zhao, Q. Dong, C. Luo, Y. Wu, D. Bu, X. Qi, Y. Luo, and Y. Zhao, "Deepomix: a scalable and interpretable multi-omics deep learning framework and application in cancer survival analysis," *Computational and structural biotechnology journal*, vol. 19, pp. 2719–2725, 2021.

[60] F. E. Harrell Jr, K. L. Lee, R. M. Califf, D. B. Pryor, and R. A. Rosati, "Regression modelling strategies for improved prognostic prediction," *Statistics in medicine*, vol. 3, no. 2, pp. 143–152, 1984.

[61] H. Uno, T. Cai, M. J. Pencina, R. B. D'Agostino, and L.-J. Wei, "On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data," *Statistics in medicine*, vol. 30, no. 10, pp. 1105–1117, 2011.

[62] M. Schmid and S. Potapov, "A comparison of estimators to evaluate the discriminatory power of time-to-event models," *Statistics in medicine*, vol. 31, no. 23, pp. 2588–2609, 2012.

[63] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthly weather review*, vol. 78, no. 1, pp. 1–3, 1950.

[64] J. D. Kalbfleisch and R. L. Prentice, *The statistical analysis of failure time data*. John Wiley & Sons, 2011.

[65] D. R. Cox, "Partial likelihood," *Biometrika*, vol. 62, no. 2, pp. 269–276, 1975.

[66] F. E. Harrell, R. M. Califf, D. B. Pryor, K. L. Lee, and R. A. Rosati, "Evaluating the yield of medical tests," *Jama*, vol. 247, no. 18, pp. 2543–2546, 1982.

[67] M. Morgan, "Access the bioconductor project package repository [r package biocmanager version 1.30. 20]," *Comprehensive R Archive Network (CRAN)*, 2023.

[68] A. Colaprico, T. C. Silva, C. Olsen, L. Garofano, C. Cava, D. Garolini, T. S. Sabedot, T. M. Malta, S. M. Pagnotta, I. Castiglioni *et al.*, "Tcgabiolinks: an r/bioconductor package for integrative analysis of tcga data," *Nucleic acids research*, vol. 44, no. 8, pp. e71–e71, 2016.

[69] M. Morgan, V. Obenchain, J. Hester, and H. Pagès, *SummarizedExperiment: A container (S4 class) for matrix-like assays*, 2024, r package version 1.36.0. [Online]. Available: https://bioconductor.org/packages/SummarizedExperiment

[70] M. I. Love, W. Huber, and S. Anders, "Moderated estimation of fold change and dispersion for rna-seq data with deseq2," *Genome biology*, vol. 15, pp. 1–21, 2014.

[71] C. R. John, D. Watson, D. Russ, K. Goldmann, M. Ehrenstein, C. Pitzalis, M. Lewis, and M. Barnes, "M3c: Monte carlo reference-based consensus clustering," *Scientific reports*, vol. 10, no. 1, p. 1816, 2020.

[72] T. M. Therneau, *A Package for Survival Analysis in R*, 2024, r package version 3.7.0. [Online]. Available: https://CRAN.R-project.org/package=survival

[73] H. Ishwaran and U. Kogalur, *Fast Unified Random Forests for Survival, Regression, and Classification (RF-SRC)*, 2023, r package version 3.3.1. [Online]. Available: https://cran.r-project.org/package=randomForestSRC

[74] Y. Zhao, M.-C. Li, M. M. Konaté, L. Chen, B. Das, C. Karlovich, P. M. Williams, Y. A. Evrard, J. H. Doroshow, and L. M. McShane, "Tpm, fpkm, or normalized counts? a comparative study of quantification measures for the analysis of rna-seq data from the nci patient-derived models repository," *Journal of translational medicine*, vol. 19, no. 1, p. 269, 2021.

[75] R. M. Simon, J. Subramanian, M.-C. Li, and S. Menezes, "Using cross-validation to evaluate predictive accuracy of survival risk classifiers based on high-dimensional data," *Briefings in bioinformatics*, vol. 12, no. 3, pp. 203–214, 2011.

List of Abbreviations

Abbreviation	Description
All 3 Omics	Multi-omics dataset with RNA, miRNA and Proteome
ANN	Artificial Neural Network
C-index	Concordance Index
CoxPH	Cox Proportional Hazards Model
DFKI	Deutsches Forschungszentrum für Künstliche Intelligenz
DSA	Data Science and its Applications
EI	Early integration
EI Subset	Early integration subset of multi-omics dataset with RNA, miRNA and Proteome
GBM	Gradient Boosting Machine
II	Intermediate integration
II Subset 1	Intermediate integration subset of around 130 features of multi-omics dataset with RNA, miRNA and Proteome
II Subset 2	Intermediate integration subset of around 70 features of multi-omics dataset with RNA, miRNA and Proteome
IBS	Integrated Brier Score
mRMR	Minimum Redundancy Maximum Relevance
miRNA	miRNA Expression Quantification
NN	Neural Network
Proteome	Protein Expression Quantification
RNA	Gene Expression RNA Sequencing
RNA and miRNA	Multi-omics dataset with RNA and miRNA
RNA, miRNA and Proteome	RNA, miRNA and Proteome
RSF	Random Survival Forest
TCGA	The Cancer Genome Atlas

List of Tables

List of Figures