SurvBoard: Standardised Benchmarking for Multi-omics Cancer Survival Models

David Wissel [1, 3, *] **Nikita Janakarajan** [1, 2, *] **Aayush Grover** [1, 3]

Enrico Toniato [2] **María Rodríguez Martínez** [2, †] **Valentina Boeva** [1, 3, †]

{dwissel, agrover, vboeva}@inf.ethz.ch
{nja, eto, mrm}@zurich.ibm.com
[1] Department of Computer Science, ETH Zurich [2] IBM Research Europe
[3] Swiss Institute for Bioinformatics [*] Equal contribution [†] Equal supervision

Abstract

High-throughput "omics" data, including genomic, transcriptomic, and epigenetic data, have become increasingly produced and have contributed in recent years to the advances in cancer research. In particular, multi-modal omics data get now employed in addition to clinical data to stratify patients according to their clinical outcomes. Despite some recent work on benchmarking multi-modal integration strategies for cancer survival prediction, there is still a need for the standardization of the results of model performances and for the consecutive exploration of the relative performance of statistical and deep learning models. Here, we propose a unique benchmark, SurvBoard, which standardizes several important experimental design choices to enable comparability between cancer survival models that incorporate multi-omics data. By designing several benchmarking scenarios, SurvBoard allows for the comparison of single-cancer models and models trained on pan-cancer data; SurvBoard also makes it possible to investigate the added value of using patient data with missing modalities. Additionally, in this work, we point out several potential pitfalls that might arise during the preprocessing and validation of multi-omics cancer survival models and address them in our benchmark. We compare statistical and deep learning models revealing that statistical models often outperform deep learning models, particularly in terms of model calibration. Finally, we offer a web service that enables quick model evaluation against our benchmark (https://www.survboard.science/). All code and other resources are available on GitHub: https://github.com/BoevaLab/survboard/.

1 Introduction

The prediction of survival-related information based on data with censored observations, commonly termed survival analysis, has recently garnered a lot of attention in both the machine learning and bioinformatics scientific communities. In the context of cancer research, survival analysis facilitates patient risk stratification, and can contribute to an improved treatment choice and ultimately to extend patients survival. With the increased throughput of *omics* data and the availability of a higher number of modalities (*i.e.,* multi-omics), leveraging such data for survival analysis has attracted considerable interest. The hope is that molecular information beyond clinical data can help investigate and discover novel treatment strategies.

With the advent of large-scale cancer projects such as The Cancer Genome Atlas (TCGA) [Tomczak et al., 2015], International Cancer Genome Consortium (ICGC) [Consortium et al., 2010] and

Therapeutically Applicable Research To Generate Effective Treatments (TARGET) [Ma et al., 2018], researchers have begun to incorporate multi-modal omics data into survival models. However, to date most existing works focus exclusively on the TCGA project datasets due to their large size and extensive omics information, which results in an increased risk of overfitting the models to these datasets. Comparison across works is also increasingly difficult due to the many different choices regarding data imputation, cancer types, test splits, and omics types utilised. Furthermore, while a considerable number of benchmarks have explored statistical and regression models for multi-omics integration in the cancer survival context [Herrmann et al., 2021, Zhao et al., 2015, Bøvelstad et al., 2009], no comprehensive benchmark has compared neural and statistical models.

Our benchmark **SurvBoard** aims to remedy these gaps. Our main contributions are: (1) A new benchmark for multi-omics integrative survival models incorporating cancer datasets from three large-scale cancer projects: TCGA, ICGC, and TARGET. SurvBoard benchmarks models in three different settings: a standard multi-omics survival analysis, a multi-omics survival analysis with samples for which some data modalities are missing, and a pan-cancer multi-omics survival analysis, where a model is jointly trained on multiple cancer types; (2) We present several case studies using both raw data and our new benchmark that illustrate potential pitfalls in the multi-omics survival analysis studies. In this context, SurvBoard aims to be a resource for both benchmarking and validation of multi-omics survival models; (3) We provide a web service that provides a leaderboard for SurvBoard, making it easy for other researchers to compare their methods to existing ones.

2 Related work

Herrmann et al. [2021] benchmarked 12 statistical multi-omics models across 18 cancer types from TCGA. They found that taking the multi-modal group structure inherent to multi-omics data into account produced superior results. A main finding of this work is that even the best performing multi-omics models did not significantly outperform a baseline trained only on clinical information. While an important contribution, this benchmark solely focused on TCGA, and did not consider other widely used publicly available cancer datasets such as ICGC and TARGET. Moreover, Herrmann et al. [2021] only considered statistical and classical machine learning models, but not neural models, which are now commonly used in survival analysis [Cheerla and Gevaert, 2019, Vale-Silva and Rohr, 2021]. And while the datasets used by Herrmann et al. [2021] are publicly available, they do not consider samples with missing modalities nor train models in a pan-cancer setting, which is becoming routine in neural networks trained for cancer survival prediction [Cheerla and Gevaert, 2019, Vale-Silva and Rohr, 2021].

Zhao et al. [2015] benchmarked several feature selection and dimensionality reduction methods on four cancer types from TCGA. While the results showed high variance across cancer types, in general, modalities beyond clinical and gene expression did not noticeably improve prediction performance. Furthermore, since the work of Zhao et al. [2015] was quite early in the lifecycle of TCGA, it considered relatively few datasets and only considered dimension reduction or feature selection methods combined with the Cox proportional hazards model. Using similar datasets, Dwivedi et al. [2022] presented an analysis tool to correlate gene sets, pathways, cellular enrichment, and single-cell signatures of clinical outcomes and assist in the development of next-generation prognostic and therapeutic biomarkers. Compared to these works, SurvBoard provides a platform that allows the comprehensive comparison of the predictive performance of new methods.

It is also worth mentioning various packages and webservices that simplify the acquisition and preprocessing of TCGA data [Ramos et al., 2020, Gao et al., 2013, Goldman et al., 2020, Colaprico et al., 2016]. As these serve a general-purpose goal, they do not significantly conflict with SurvBoard, whose goal is not only to simplify but to effectively remove the need of data preprocessing when benchmarking multi-omics survival models.

3 Benchmark design

3.1 Settings for Multi-Omics Survival Analysis

As discussed, we believe there are certain pitfalls in the validation of multi-omics survival analysis model that ought to be addressed. We demonstrate those in three different settings related to training, validation and comparability of survival models:

1. **Standard.** Our first setting implements standard multi-omics survival analysis. This means that each model is trained and tested only on samples of the same cancer type.

2. **Missing data modality.** Typically, not all omics information is available for all patients, and many patients have missing omics modalities. Exclusion of samples exhibiting *missing modalities* can noticeably decrease cohort size, and methods have been developed to train models that can handle samples with missing modalities. Methods that can handle missing modalities have two advantages over methods that cannot: (1) They benefit from an increased training set size due to their ability to train on samples with incomplete observations. The larger dataset size can potentially lead to improved performance. (2) These models can make predictions for patients with both complete and incomplete modalities, allowing a more thorough evaluation of a particular model

In our missing data modality setting, we only exploit the first advantage to ensure comparability with other models by using a consistent test set, *i.e.,* methods including samples with missing modalities are tested on the exact same samples as standard methods (since missing modality samples were only used as additional training data).

Throughout the remainder of the text, we abbreviate the missing data modality setting as merely the *missing* setting.

3. **Pan-cancer.** In our third setting, we jointly train on datasets from multiple cancer types. A general expectation is that these models generalise better. In practice, pan-cancer training can only be achieved by models capable of handling missing modalities, since not all datasets share identical modalities.

3.2 Preprocessing

While several packages (like Cerami et al. [2012]) and data sources (like Weinstein et al. [2013]) allow the acquisition and exploitation of TCGA, ICGC, and TARGET datasets, preprocessing choices are left to the user, which leads to confusion and inconsistency across experiments. To enable a fair comparison of existing and new methods, we standardize as many preprocessing choices as possible. In this section, we outline the general practices that SurvBoard follows.

Endpoint choice. While all of our considered cancer datasets provide multiple endpoints (*e.g.,* overall survival (OS), Disease Free Survival (DFS) and others), it is relatively common in the survival literature to utilize OS, as it is ubiquitously available compared to other endpoints.We follow this broader convention and use OS as the endpoint of all datasets within our benchmark.

Patient cohorts. We restricted our datasets to *primary* tumor tissue samples. Additionally, we also excluded those patients for whom either the event indicator or the event time was missing.However, users still have a possibility of using this data through a command line argument (see GitHub).

Dataset selection. We only considered datasets that included a minimum of 100 samples and a minimum event ratio of 5% or 10 total events (whichever was larger) to ensure that meaningful performance metrics could be computed. We note that we only counted samples with complete modalities when considering the dataset selection, as only these were part of the test splits (missing modality samples serving only as additional training data, refer to Section 3.1 for details).

Modality selection. We selected the maximum number of modalities (including both clinical data and omics modalities) available for each dataset in each project, subject to the dataset selection criteria stated in **Dataset selection**. We excluded datasets that did not possess at least two omic modalities (*e.g.,* gene expression and mutation) to stay true to the name of **multi**-omics. For example, if one dataset had four omics modalities available, but only 70 samples that had all those modalities present, we used only three modalities if they met our criteria for sample size. We excluded datasets that did not meet the minimum number of modalities required. We note that the minimum number of modalities did not apply to missing modality samples; *i.e.,* some missing modality samples may only have one modality if they were missing all other modalities that were available for that particular cancer dataset.

Clinical variables. To ensure a fair comparison across projects and cancer types, we only included standard and readily (at time of diagnosis) available clinical information, such as demographic (*e.g.,* age and gender) and staging (*e.g.,* clinical stage) information. While there are other clinical features that are significantly associated with cancer progression and survival, we chose to ignore those as they can be cancer type-specific, such as smoking-history for lung cancer. Furthermore, using a common set of clinical variables enables a fair comparison between single-cancer and pan-cancer

models. Categorical variables were left as strings to enable each model to determine its handling of categorical variables. Appendix A provides an exhaustive overview of which clinical variables were included for each dataset in each project, as sometimes slightly different variables were available.

Missing values (within modalities). To handle missing values, we followed a three-step procedure. First, for categorical variables, we created a separate category corresponding to NA called *MISSING*. Specifically, we assume the information is Missing Not At Random (MNAR) [Van Buuren, 2018], *i.e.,* the missingness of a categorical variable might be correlated with either the target variable or other covariates. Our aim here is to avoid mixing potentially unrelated categories by, for example, using mode imputation. Second, non-categorical variables missing in more than 10% of samples in a specific dataset were excluded (only) from that dataset. Third, non-categorical variables with missing rates less than 10% in a dataset were imputed using the median of the available samples on the full dataset. While imputation in the full dataset (*i.e.,* before splitting) might lead to some information leakage, research has found that it does not incur significant bias [Hornung et al., 2015]. Therefore, we chose this approach in order to eliminate all non-model-specific preprocessing from researchers. Further, rather that aiming to construct a *perfectly* preprocessed dataset, which would be impossible to agree on due to the many different preprocessing choices, SurvBoard ensures the usage of common datasets throughout the literature. [1]

Missing modalities. We created two sets of samples for each dataset in each project - one "complete" dataset, used in the "standard" setting and to create the dataset splits, and an "incomplete" dataset that includes samples with one or more missing modalities (from the modalities used in the complete dataset for that cancer type). Importantly, the incomplete dataset is intended *only* as additional training data in our benchmark, as noted in Section 3.1. Within the incomplete datasets, NA values indicate that a particular modality is missing in a specific sample. NA values for features within available modalities were no longer present in the incomplete dataset as they had been imputed or removed as detailed in the previous paragraph.

Pan-cancer Since only TCGA provided pan-cancer normalised data, we only created a pan-cancer dataset for TCGA and not on ICGC or TARGET. We combined variables across all TCGA datasets in the pan-cancer data, excluding variables that were not available for all cancer types. If a particular dataset did not have a modality available, we did not remove this modality from the pan-cancer dataset, but instead set it to missing for the samples from that cancer type only.

3.3 Datasets

TCGA project We included 21 cancer datasets from TCGA, which is arguably the largest and most commonly used database for multi-omics cancer survival analysis. TCGA is also the only project usable in the pan-cancer setting since other projects do not provide data normalised across cancer types.

ICGC project We also included four cancer datasets from ICGC based on the selection criteria highlighted in Section 3.2. Of note, generally, TCGA datasets are also included in ICGC. To avoid duplications, we included only ICGC datasets that not already part of TCGA.

TARGET project We incorporated three datasets from TARGET, a pediatric cancer database, similarly based on the selection criteria highlighted in Section 3.2.

Thus, our benchmark includes a total of 28 cancer datasets. Table S1 provides an overview of the number of samples, modalities, and features across all datasets. Appendix A provides additional information of all datasets from all projects. Table S2 lists all cancer type name abbreviations.

3.4 Performance metrics

Throughout our benchmarks, we calculated three performance metrics, all of which evaluate the survival function. First, we used the **Antolini Concordance** [Antolini et al., 2005] to measure the ability of each survival model to discriminate low-risk patients from high-risk patients across time. Second, the **Integrated Brier score** [Graf et al., 1999], a relatively standard measure in survival benchmarks, takes both discrimination and calibration accuracy into account. Third, the recently proposed **D-Calibration (D-CAL)** [Haider et al., 2020] measures the distributional calibration of

[1] Recently, label errors have been highlighted in multiple computer vision datasets [Northcutt et al., 2021]; despite this, these datasets have arguably greatly accelerated progress in computer vision research.

each multi-omics survival model. For D-CAL, we evaluated the test statistic where lower values corresponded to a better fit. We provide further details on all considered performance metrics in Appendix B.

3.5 Validation

As validation, we performed five-fold cross-validation, five times repeated, yielding a total of 25 test splits per cancer. Splits were created by stratifying by the event indicator, OS. This means that, for each train and test fold, we kept the same event ratio as in the original, unsplit data.[2] As mentioned previously, the incomplete modality samples were never part of the test set, serving instead only as additional training data. In the pan-cancer setting, training data from all cancer types were included in each training split. Since large-scale benchmarks often encounter model failures due to the large number of models being trained (see *e.g.,* Herrmann et al. [2021]), our benchmark was required to handle these failing iterations.We rely on a simple fallback learner, replacing failing model iterations with a Kaplan-Meier estimator [Kaplan and Meier, 1958].

3.6 Web service

We provided a web service that allows interested researchers to submit predictions on our benchmark. The web service automatically evaluates all submitted predictions and shows all performance metrics for all datasets within our benchmark, allowing for easy comparison to other models. We show an example submission on our web service in Appendix C. A sample submission file and links to our web service are available within our Github repository.

4 Case studies

Here, we illustrate several pitfalls associated with the validation of (multi-) omics survival analysis models and present our approach using our benchmark, SurvBoard. Our case studies emphasise the importance of some of the design choices we made in our benchmark SurvBoard, and may be useful for other researchers to validate their models on small n and large p omics data size regimes.

We evaluated a selection of statistical models, including ℓ_1 penalised Cox model (Lasso) [Friedman et al., 2010, Simon et al., 2011], a random survival forest (Random Survival Forest (RSF)) [Ishwaran et al., 2008, Wright and Ziegler, 2017], and a group-aware random survival forest called BlockForest [Hornung and Wright, 2019], which was also the best performing statistical model from the study of Herrmann et al. [2021]. We also included two neural models, one based on the mean-pooling architecture of Cheerla and Gevaert [2019], and the other, a fully connected architecture that disregards the group structure of the multi-omics data. Further details on statistical and neural models can be found in Appendix D and Appendix E respectively.

Clinical variable choice. The choice of clinical variables is key to both performance and potential pitfalls. Indeed, while it might be alluring to use treatment related variables as predictive features, some treatment-related variables such as "non-therapy" may be associated with disease severity. For instance, a clinician might decide against starting radiation therapy if the patient is expected to die soon of their disease or other causes [Arenas et al., 2014]. Indeed, on TCGA, we observed cancer types in which treatment related variables such as "radiation therapy" were associated with the outcome to a degree that made it unlikely to be solely due to the treatment effect (Fig. 1). This strong association suggests instead a potential leakage of the target (Fig. 1). As treatment related variables may not accurately capture the true treatment effect due to confounder effects, *e.g.,* sufficient disease progression warranting palliative care, the choice of which clinical variables to use for training should be carefully investigated.

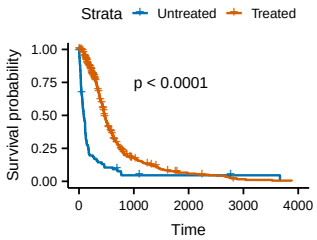

Figure 1: Kaplan-Meier estimator on glioblastoma multiforme (TCGA-GBM) stratified by whether a patient received radiation therapy. Patients with an unknown value for radiation therapy were excluded.

[2]Practically, we used the *sklearn.model_selection.RepeatedStratifiedKFold* from *scikit learn* [Pedregosa et al., 2011].

Not performing a sufficient number of repetitions of cross-validation. While empirical studies have emphasised and practiced the use of repeated cross-validation [Herrmann et al., 2021], it is still relatively common for multi-omics survival analysis method papers to validate their method with unrepeated cross-validation, and in some cases with just a single split. We found a large variation in model ranking with respect to Antolini concordance across cross-validation repetitions for selected models on SurvBoard (Fig. 2). Smaller datasets such as the Acute Lymphoblastic Leukemia (TARGET-ALL) dataset especially exemplify this issue. Larger datasets, such as breast cancer (TCGA-BRCA), showed comparatively greater consistency, although, they still suffered from some variability across cross-validation repetitions. Similar patterns were observed when using the Integrated Brier score (IBS) (Fig. S2) and D-CAL (Fig. S3), although we found the IBS to be noticeably more stable across cross-validation repetitions than our other two metrics. When pooling across datasets, variability in terms of ranks decreased noticeably for all metrics as expected (Fig. S4 - Fig. S6). Thus, it may be advantageous to either perform several repetitions of cross-validation when evaluating performances on singular datasets or pool performances across datasets, combined with an appropriate statistical test, such as the Wilcoxon ranked sign test [Demšar, 2006].

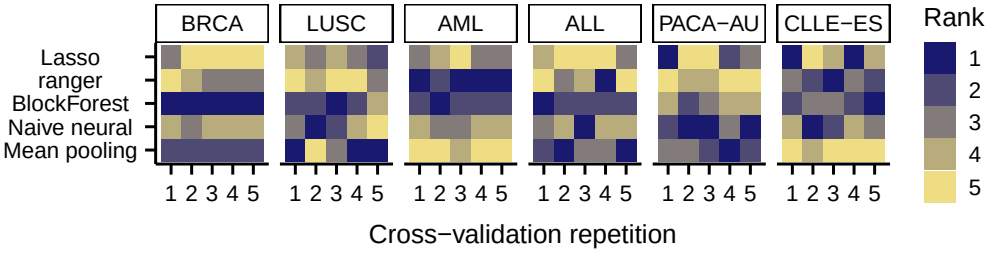

Figure 2: Antolini concordance rank of all considered models across cross-validation repetitions on all cancer types; one signifies best rank, five worst rank.

Reporting results using a single metric.

While discriminative metrics, such as Harrell's concordance or Antolini's concordance have been commonly employed for evaluating survival models, recent works have also increasingly included calibration metrics such as the IBS.We emphasise that reporting *at least* one discriminative metric and one calibration metric is paramount, as these are not necessarily correlated (see Appendix B for further details). In particular, we found that on various datasets included in SurvBoard, one could easily report only one of the metrics which might favour a particular model over all others (*e.g.,* if one wanted to favour neural models, one could only report discriminative metrics such as concordance). On bladder cancer (TCGA-BLCA), for example, we found both neural models to perform as good or better than all other models in terms of discrimination (Fig. 3A), while their calibration was generally worse and had much larger variance (Fig. 3B-C). Our point here stands especially since measures of distributional calibration (especially D-CAL) have only recently been proposed and employed for survival analysis, and, to the best of our knowledge, they have not been widely used for (multi-) omics survival analysis models.

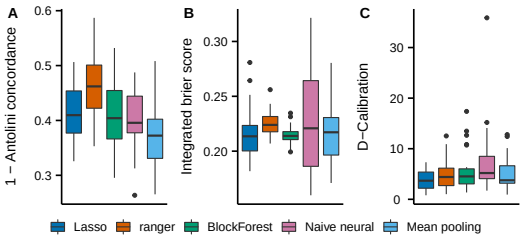

Figure 3: Performance of selected multi-omics survival methods on bladder cancer (TCGA-BLCA). Lower values are better. A. 1 - Antolini concordance. B. IBS. C. D-CAL. Lower values are better.

5 Benchmark results

Throughout our benchmark results, we test for significance when pooling by project only on TCGA, since ICGC and TARGET contain a small number of datasets (four and three respectively), making statistical tests for testing performance differences across datasets underpowered. We present results for our benchmark for the same models that we used throughout our case studies, Priority-Lasso [Klau et al., 2018] and CoxBoost [Binder and Schumacher, 2008]. Table S3 contains an overview

of the number of model failures during our study. Consistent with Herrmann et al. [2021], the only model showing failures is Lasso, failing in 36% of its runs.

Benefits from including missing modalities and pan-cancer training are dataset and metric-dependent. In our first experiment, we investigated the performance impact of training a specified neural model in different settings (see Section 3.1 for details). We found that on both TCGA and ICGC, using samples with some missing modalities improved our mean pooling models' performance by 0.021 (TCGA) and 0.007 (ICGC) (Table 1). For the calibration metrics IBS and D-CAL, the results were more nuanced. On TCGA the IBS slightly worsened, while D-CAL improved on including samples with missing modalities. On ICGC, both the IBS and D-CAL improved in the missing setting, relative to the standard setting. On TARGET, the missing setting worsened performance in all metrics relative to the standard setting. This is attributed to a large presence of clinical-only patients in the TARGET datasets. For example, TARGET-ALL had 59% of patients with only clinical data available. These results highlight that the potential improvements to be gained by including samples with missing modalities are heavily dataset- and cancer-type-dependent.

Table 1: Pooled mean performance metrics stratified by project for the mean pooling neural architecture in all settings. Standard deviation is shown in parentheses. On TCGA, values not significantly worse than best are bold-faced based on Wilcoxon signed-rank test (p $\geq$ 0.05). $\uparrow$ signifies a metric for which higher is better, $\downarrow$ signifies a metric for which lower is better.

Model	Setting	Project	Antolini concordance ($\uparrow$)	IBS ($\downarrow$)	D-CAL ($\downarrow$)
Mean pooling	Standard	TCGA	0.602 (0.129)	**0.192 (0.042)**	**4.655 (4.062)**
Mean pooling	Missing	TCGA	**0.623 (0.125)**	**0.194 (0.05)**	**4.4 (3.921)**
Mean pooling	Pan-cancer	TCGA	**0.631 (0.13)**	**0.191 (0.047)**	5.92 (9.095)
Mean pooling	Standard	ICGC	0.623 (0.102)	0.199 (0.072)	10.792 (13.657)
Mean pooling	Missing	ICGC	0.63 (0.105)	0.193 (0.07)	10.11 (13.339)
Mean pooling	Standard	TARGET	0.614 (0.099)	0.207 (0.026)	5.217 (5.385)
Mean pooling	Missing	TARGET	0.585 (0.086)	0.227 (0.035)	6.036 (4.675)

Pan-cancer training (see Section 3.1) further improved concordance and IBS on TCGA, while D-CAL got slightly worse relative to the missing and standard settings (Table 1). When stratified by cancer type on TCGA, it became clear that potential gains from both the pan-cancer setting and the missing modality setting were dataset and cancer type-specific for all metrics (Table S4 - Table S6). Thus, researchers should carefully evaluate whether including samples with some missing modalities or training on multiple cancer types might be beneficial for their application.

Statistical models tend to outperform neural models in the standard setting. Next, we compared neural and statistical models. Most neural studies do not compare to group-aware statistical models, such as BlockForest, despite Herrmann et al. [2021] finding these statistical models to perform quite well. We considered neural models only in the standard setting since statistical models are only trainable in this setting (refer to Section 3.1 for details).

For all projects and all metrics, we found that our best neural models underperform slightly compared to our best statistical models (Table 2). While neural models were competitive in terms of concordance, their calibration tended to be generally worse, as evidenced by higher IBS and D-CAL test statistics (*e.g.,* on TCGA, the mean pooling model is not significantly worse than the best statistical models concordance, but both its IBS and D-CAL were significantly worse). On TCGA, we found that using models which were aware of the group-structure of the multi-omics data generally performed better. For instance, BlockForest outperformed the group-naive but otherwise equivalent RSF. A similar observation holds for the other model pairings that share model class but not naivety about the group structure (see Appendix D and Appendix E for details about which models incorporate information about the group structure). CoxBoost, another linear Cox-based model naive to the group structure of the data, performed quite similar to the Lasso on TCGA in terms of IBS and D-CAL, and outperformed the Lasso by margins of roughly 0.03 and 0.02 in terms of Antolini's concordance on ICGC and TARGET. On TCGA, the two models' discriminative performance was similar, with CoxBoost outperforming the Lasso by only 0.01 (Table 2). Priority-Lasso showed the highest concordance on TCGA out of all models (Table 2), while it achieved a slightly higher (that is, worse) IBS compared to BlockForest and a slightly lower (*i.e.,* better) D-CAL test statistic. We generally found linear models to perform relatively better compared to Herrmann et al. [2021] on

Table 2: Pooled mean performance metrics stratified by project for neural and statistical models in the standard setting. Standard deviation is shown in parentheses. On TCGA, values not significantly worse than best are bold-faced based on Wilcoxon signed-rank test ($p \geq 0.05$). ↑ signifies a metric for which higher is better, ↓ signifies a metric for which lower is better.

Model	Setting	Project	Antolini concordance (↑)	IBS (↓)	D-CAL (↓)
Lasso	Standard	TCGA	0.577 (0.117)	**0.187 (0.036)**	**3.876 (3.151)**
RSF	Standard	TCGA	0.57 (0.123)	0.191 (0.033)	4.321 (3.753)
BlockForest	Standard	TCGA	0.611 (0.128)	**0.186 (0.032)**	4.396 (3.562)
Fully connected	Standard	TCGA	0.582 (0.126)	0.206 (0.056)	6.191 (6.536)
Mean pooling	Standard	TCGA	0.602 (0.129)	0.192 (0.042)	4.655 (4.062)
CoxBoost	Standard	TCGA	0.583 (0.122)	**0.187 (0.037)**	**3.873 (3.176)**
Priority-Lasso	Standard	TCGA	**0.629 (0.127)**	**0.191 (0.055)**	4.078 (3.331)
Lasso	Standard	ICGC	0.615 (0.092)	0.165 (0.03)	4.816 (4.646)
RSF	Standard	ICGC	0.633 (0.114)	0.161 (0.032)	5.89 (5.315)
BlockForest	Standard	ICGC	0.645 (0.11)	0.16 (0.03)	4.54 (4.671)
Fully connected	Standard	ICGC	0.635 (0.104)	0.206 (0.082)	12.97 (16.281)
Mean pooling	Standard	ICGC	0.623 (0.102)	0.199 (0.072)	10.792 (13.657)
CoxBoost	Standard	ICGC	0.63 (0.094)	0.166 (0.033)	4.899 (4.322)
Priority-Lasso	Standard	ICGC	0.629 (0.107)	0.164 (0.034)	4.732 (4.342)
Lasso	Standard	TARGET	0.609 (0.087)	0.21 (0.025)	4.252 (4.325)
RSF	Standard	TARGET	0.641 (0.078)	0.202 (0.027)	4.547 (3.297)
BlockForest	Standard	TARGET	0.639 (0.086)	0.202 (0.024)	4.573 (3.497)
Fully connected	Standard	TARGET	0.624 (0.093)	0.213 (0.036)	6.695 (5.868)
Mean pooling	Standard	TARGET	0.614 (0.099)	0.207 (0.026)	5.217 (5.385)
CoxBoost	Standard	TARGET	0.622 (0.08)	0.208 (0.027)	4.058 (3.334)
Priority-Lasso	Standard	TARGET	0.614 (0.099)	0.207 (0.027)	4.246 (3.891)

TCGA, who found much larger gaps between BlockForest and both Priority-Lasso and the Lasso. That said, we believe these results are explainable by a difference in the considered cancer types and a larger number of modalities.

Interestingly, we found that on ICGC and TARGET, incorporating the group structure into the models seemed considerably less important, with some of the group naive models outperforming their group-aware but otherwise equivalent models (*e.g.,* the fully connected model outperformed the mean-pooling neural model in terms of concordance on both ICGC and TARGET). Both TARGET and ICGC have a lower dimensionality and a lower number of modalities than TCGA (Table S1), which likely contributed to this pattern. On ICGC and TARGET, Priority-Lasso also performed well, but was not able to edge out RSF or BlockForest in terms of discriminative performance, although it continued to outperform the Lasso in terms of concordance across projects.

Omics data offer predictive value beyond clinical on some but not all projects. Lastly, we investigated the importance of clinical data on all projects. Herrmann et al. [2021] found that none of their considered statistical models were able to outperform a clinical-only Cox baseline on TCGA. We aimed to reproduce and extend these experiments also to TARGET and ICGC. Specifically, we compared the best multi-omics model in terms of Antolini's concordance in each project, as well as a baseline multi-omics Lasso model, to a clinical-only Cox baseline on TCGA, ICGC and TARGET.

Despite using considerably fewer clinical variables than Herrmann et al. [2021], the clinical-only Cox model still outperformed the best multi-omics model on TCGA in terms of concordance (Table 3). However, we found that the clinical-only Cox model had a worse calibration than the multi-omics models, attaining a higher IBS and D-CAL test statistics. This is in contrast to the findings of Herrmann et al. [2021] where the clinical-only Cox model was on par with the best multi-omics models in terms of the IBS. Since Herrmann et al. [2021] used considerably more clinical variables than we did, we believe this difference in calibration is likely due to the differing clinical variables.

On TARGET and ICGC, the situation was considerably different compared to TCGA. The clinical-only Cox model was outperformed even by a group-naive Lasso model, which achieved better metrics in all cases but D-CAL on ICGC. Further, the best multi-omics models on ICGC and TARGET outperformed the clinical-only model by fairly large margins, suggesting that the clinical information

Table 3: Pooled mean performance metrics stratified by project for clinical-only Cox model, Lasso multi-omics baseline, and the best multi-omics model within each setting. Standard deviation shown in parentheses. On TCGA, values not significantly worse than best are bold-faced per performance metric based on Wilcoxon signed-rank tests ($p \geq 0.05$). ↑ signifies a metric for which higher is better, ↓ signifies a metric for which lower is better.

Model	Setting	Project	Antolini concordance (↑)	IBS (↓)	D-CAL (↓)
Lasso	Standard	TCGA	0.577 (0.117)	**0.187 (0.036)**	**3.876 (3.151)**
Priority-Lasso	Standard	TCGA	**0.629 (0.127)**	**0.191 (0.055)**	**4.078 (3.331)**
Cox	Standard	TCGA	**0.624 (0.111)**	0.205 (0.071)	4.922 (4.291)
Lasso	Standard	ICGC	0.615 (0.092)	0.165 (0.03)	4.816 (4.646)
BlockForest	Standard	ICGC	0.645 (0.11)	0.16 (0.03)	4.54 (4.671)
Cox	Standard	ICGC	0.595 (0.105)	0.172 (0.04)	4.818 (4.592)
Lasso	Standard	TARGET	0.609 (0.087)	0.21 (0.025)	4.252 (4.325)
RSF	Standard	TARGET	0.641 (0.078)	0.202 (0.027)	4.547 (3.297)
Cox	Standard	TARGET	0.591 (0.091)	0.226 (0.041)	5.402 (4.339)

on TARGET and ICGC was much more limited compared to TCGA (as mentioned previously, statistical tests are severely underpowered on TARGET and ICGC, due to them only having three and four datasets respectively, thus we do not show test results). More broadly, we recommend practitioners and researchers to carefully consider their research question or application in order to decide whether multi-omics data might offer increased performance relative to a clinical-only model.

6 Discussion

In this work, we have presented SurvBoard, a rigorous benchmark and validation of (multi-) omics survival models. SurvBoard enables the rigorous comparisons of models across 28 datasets from 3 projects. We also focused on model comparability by ensuring that models utilising pan-cancer data or samples with missing modalities can be compared to models trained on single datasets. Additionally, we provided a simple web service that allows researchers to easily evaluate their models on our new benchmark. We have also illustrated potential pitfalls during the validation of (multi-) omics survival models, highlighting the importance of the choice of clinical variables, the use of repeated cross validation, and the display of results using several relevant metrics. We summed up our recommendations for validating and training multi-omics survival models in Appendix F.

On benchmarking several statistical and neural models on SurvBoard, we found that neural models' performance was not significantly worse than statistical models in terms of discriminative power whereas their calibration tended to be worse. Similar to Herrmann et al. [2021], we also found that the use of multi-omics did not improve performance beyond a clinical-only baseline on TCGA. On ICGC and TARGET, we observed the opposite, suggesting that the predictive power of clinical-only models depended strongly on both cancer types and the specific clinical variables considered.

Future work and limitations. Future work investigating the role of using different loss functions for neural multi-omics models would be useful as all of our results are based on the Cox loss function. Analysis of patient similarity between training and test sets would be beneficial to both SurvBoard and related studies, as it can identify and prevent potential quasi-leakage due to high patient similarity between training and test samples. In addition, future work is needed to investigate the computational requirements of different multi-omics survival models, building on the work of Herrmann et al. [2021]. Our benchmark contains a fairly high number of modalities, which may affect some models negatively (*e.g.,* [Cheerla and Gevaert, 2019] and [Vale-Silva and Rohr, 2021] both found that their model performed best with only two modalities). Further investigation is required to identify additional pitfalls and to determine whether they are generalisable to all datasets. In this study, we evaluated two neural architectures. We also investigated more sophisticated neural models based on the product of experts approach of Lee and van der Schaar [2021], however, we found them exceedingly difficult to train, presumably due to the low sample size in the standard setting. Future work should consider the potential benefit of using more sophisticated neural architectures considering the limited sample sizes of omics datasets. Moroever, our standard choice of clinical

variables to ensure model comparisons in the pan-cancer setting may be a disadvantage to other models. Lastly, even though SurvBoard contains several projects, the TCGA project contributed considerably more datasets than the other projects. Thus, potential inference drawn based on the smaller projects might be less reliable relative to TCGA. In the future, we look forward to extending SurvBoard with more datasets, depending on the feedback of the community.

Societal impact. Providing both a consistent preprocessing pipeline as well as an online resource that makes the evaluation of multi-omics survival models possible should improve and motivate further research that may ultimately lead to better patient outcomes in the future. In addition, our benchmark places high importance on clinical applicability (*e.g.,* through the usage of only primary tissue samples) and on robust validation, producing more reliable conclusions about the superiority of different models. That said, some users may try to overfit to SurvBoard, since all data and splits are publicly available, which may cause researchers or clinicians to erroneosuly place trust in particular survival models. To discourage this practice, we require code submission in order to participate in SurvBoard; this aims at curtailing overfitting. Structured code with callable models can be verified and will have an indication for the same. All projects we use (ICGC, TCGA and TARGET) have gone to great lengths to make sure their data is non-identifiable with respect to original patient identity.

Acknowledgments and Disclosure of Funding

The results shown here are in whole or part based upon data generated by the TCGA Research Network: https://www.cancer.gov/tcga. The results shown here are in whole or part based upon data generated by the ICGC: https://dcc.icgc.org. The results published here are in whole or part based upon data generated by the Therapeutically Applicable Research to Generate Effective Treatments (https://ocg.cancer.gov/programs/target) initiative, phs000218. The data used for this analysis are available at https://portal.gdc.cancer.gov/projects.

References

Katarzyna Tomczak, Patrycja Czerwińska, and Maciej Wiznerowicz. The cancer genome atlas (tcga): an immeasurable source of knowledge. *Contemporary oncology*, 19(1A):A68, 2015.

International Cancer Genome Consortium et al. International network of cancer genome projects. *Nature*, 464(7291):993, 2010.

Xiaotu Ma, YU Liu, Yanling Liu, Ludmil B Alexandrov, Michael N Edmonson, Charles Gawad, Xin Zhou, Yongjin Li, Michael C Rusch, John Easton, et al. Pan-cancer genome and transcriptome analyses of 1,699 paediatric leukaemias and solid tumours. *Nature*, 555(7696):371–376, 2018.

Moritz Herrmann, Philipp Probst, Roman Hornung, Vindi Jurinovic, and Anne-Laure Boulesteix. Large-scale benchmark study of survival prediction methods using multi-omics data. *Briefings in bioinformatics*, 22(3):bbaa167, 2021.

Qing Zhao, Xingjie Shi, Yang Xie, Jian Huang, BenChang Shia, and Shuangge Ma. Combining multidimensional genomic measurements for predicting cancer prognosis: observations from tcga. *Briefings in bioinformatics*, 16(2):291–303, 2015.

Hege M Bøvelstad, Ståle Nygård, and Ørnulf Borgan. Survival prediction from clinico-genomic models-a comparative study. *BMC bioinformatics*, 10(1):1–9, 2009.

Anika Cheerla and Olivier Gevaert. Deep learning with multimodal representation for pancancer prognosis prediction. *Bioinformatics*, 35(14):i446–i454, 2019.

Luís A Vale-Silva and Karl Rohr. Long-term cancer survival prediction using multimodal deep learning. *Scientific Reports*, 11(1):1–12, 2021.

Bhakti Dwivedi, Hope Mumme, Sarthak Satpathy, Swati S Bhasin, and Manoj Bhasin. Survival genie, a web platform for survival analysis across pediatric and adult cancers. *Scientific Reports*, 12(1): 1–10, 2022.

Marcel Ramos, Ludwig Geistlinger, Sehyun Oh, Lucas Schiffer, Rimsha Azhar, Hanish Kodali, Ino de Bruijn, Jianjiong Gao, Vincent J Carey, Martin Morgan, et al. Multiomic integration of public oncology databases in bioconductor. *JCO Clinical Cancer Informatics*, 1:958–971, 2020.

Jianjiong Gao, Bülent Arman Aksoy, Ugur Dogrusoz, Gideon Dresdner, Benjamin Gross, S Onur Sumer, Yichao Sun, Anders Jacobsen, Rileen Sinha, Erik Larsson, et al. Integrative analysis of complex cancer genomics and clinical profiles using the cbioportal. *Science signaling*, 6(269): pl1–pl1, 2013.

Mary J Goldman, Brian Craft, Mim Hastie, Kristupas Repečka, Fran McDade, Akhil Kamath, Ayan Banerjee, Yunhai Luo, Dave Rogers, Angela N Brooks, et al. Visualizing and interpreting cancer genomics data via the xena platform. *Nature biotechnology*, 38(6):675–678, 2020.

Antonio Colaprico, Tiago C Silva, Catharina Olsen, Luciano Garofano, Claudia Cava, Davide Garolini, Thais S Sabedot, Tathiane M Malta, Stefano M Pagnotta, Isabella Castiglioni, et al. Tcgabiolinks: an r/bioconductor package for integrative analysis of tcga data. *Nucleic acids research*, 44(8):e71–e71, 2016.

Ethan Cerami, Jianjiong Gao, Ugur Dogrusoz, Benjamin E Gross, Selcuk Onur Sumer, Bülent Arman Aksoy, Anders Jacobsen, Caitlin J Byrne, Michael L Heuer, Erik Larsson, et al. The cbio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data, 2012.

John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, and Joshua M Stuart. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120, 2013.

Stef Van Buuren. *Flexible imputation of missing data*. CRC press, 2018.

Roman Hornung, Christoph Bernau, Caroline Truntzer, Rory Wilson, Thomas Stadler, and Anne-Laure Boulesteix. A measure of the impact of cv incompleteness on prediction error estimation with application to pca and normalization. *BMC Medical Research Methodology*, 15(1):1–15, 2015.

Curtis Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL `https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/file/f2217062e9a397a1dca429e7d70bc6ca-Paper-round1.pdf`.

Laura Antolini, Patrizia Boracchi, and Elia Biganzoli. A time-dependent discrimination index for survival data. *Statistics in medicine*, 24(24):3927–3944, 2005.

Erika Graf, Claudia Schmoor, Willi Sauerbrei, and Martin Schumacher. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in medicine*, 18(17-18):2529–2545, 1999.

Humza Haider, Bret Hoehn, Sarah Davis, and Russell Greiner. Effective ways to build and evaluate individual survival distributions. *J. Mach. Learn. Res.*, 21(85):1–63, 2020.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Edward L Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481, 1958.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. doi: 10.18637/jss.v033.i01. URL `https://www.jstatsoft.org/v33/i01/`.

Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(5):1–13, 2011. doi: 10.18637/jss.v039.i05. URL `https://www.jstatsoft.org/v39/i05/`.

Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, and Michael S Lauer. Random survival forests. *The annals of applied statistics*, 2(3):841–860, 2008.

Marvin N Wright and Andreas Ziegler. ranger: A fast implementation of random forests for high dimensional data in c++ and r. *Journal of Statistical Software*, 77:1–17, 2017.

Roman Hornung and Marvin N Wright. Block forests: random forests for blocks of clinical and omics covariate data. *BMC bioinformatics*, 20(1):1–17, 2019.

Meritxell Arenas, Sebastià Sabater, Marina Gascón, Ivan Henríquez, M José Bueno, Àngels Rius, Àngels Rovirosa, David Gómez, Anna Lafuerza, Albert Biete, et al. Quality assurance in radiotherapy: analysis of the causes of not starting or early radiotherapy withdrawal. *Radiation Oncology*, 9 (1):1–6, 2014.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

Simon Klau, Vindi Jurinovic, Roman Hornung, Tobias Herold, and Anne-Laure Boulesteix. Priority-lasso: a simple hierarchical approach to the prediction of clinical outcome using multi-omics data. *BMC bioinformatics*, 19(1):1–14, 2018.

Harald Binder and Martin Schumacher. Allowing for mandatory covariates in boosting estimation of sparse high-dimensional survival models. *BMC bioinformatics*, 9(1):1–10, 2008.

Changhee Lee and Mihaela van der Schaar. A variational information bottleneck approach to multi-omics data integration. In *International Conference on Artificial Intelligence and Statistics*, pages 1513–1521. PMLR, 2021.

Travers Ching, Xun Zhu, and Lana X Garmire. Cox-nnet: an artificial neural network method for prognosis prediction of high-throughput omics data. *PLoS computational biology*, 14(4):e1006076, 2018.

Sunkyu Kim, Keonwoo Kim, Junseok Choe, Inggeol Lee, and Jaewoo Kang. Improved survival analysis by learning shared genomic information from pan-cancer data. *Bioinformatics*, 36 (Supplement_1):i389–i398, 2020.

Anand Mayakonda, De-Chen Lin, Yassen Assenov, Christoph Plass, and H Phillip Koeffler. Maftools: efficient and comprehensive analysis of somatic variants in cancer. *Genome research*, 28(11): 1747–1756, 2018.

Craig H Mermel, Steven E Schumacher, Barbara Hill, Matthew L Meyerson, Rameen Beroukhim, and Gad Getz. Gistic2. 0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers. *Genome biology*, 12(4):1–14, 2011.

GenomeOC. Therapeutically applicable research to generate effective treatments, Mar 2021. URL `https://ocg.cancer.gov/programs/target`.

Frank E Harrell, Robert M Califf, David B Pryor, Kerry L Lee, and Robert A Rosati. Evaluating the yield of medical tests. *Jama*, 247(18):2543–2546, 1982.

Hajime Uno, Tianxi Cai, Michael J Pencina, Ralph B D'Agostino, and Lee-Jen Wei. On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in medicine*, 30(10):1105–1117, 2011.

Raphael Sonabend, Andreas Bender, and Sebastian Vollmer. Avoiding c-hacking when evaluating survival distribution predictions with discrimination measures. 2022.

Håvard Kvamme, Ørnulf Borgan, and Ida Scheel. Time-to-event prediction with neural networks and cox regression. *arXiv preprint arXiv:1907.00825*, 2019.

Glenn W Brier et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Qixian Zhong, Jonas W Mueller, and Jane-Ling Wang. Deep extended hazard models for survival analysis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 15111–15124. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/7f6caf1f0ba788cd7953d817724c2b6e-Paper.pdf`.

Michel Lang, Martin Binder, Jakob Richter, Patrick Schratz, Florian Pfisterer, Stefan Coors, Quay Au, Giuseppe Casalicchio, Lars Kotthoff, and Bernd Bischl. mlr3: A modern object-oriented machine learning framework in r. *Journal of Open Source Software*, 4(44):1903, 2019.

Raphael Sonabend, Franz J Király, Andreas Bender, Bernd Bischl, and Michel Lang. mlr3proba: An r package for machine learning in survival analysis. *Bioinformatics*, 37(17):2789–2791, 2021.

Terry M. Therneau and Patricia M. Grambsch. *Modeling Survival Data: Extending the Cox Model*. Springer, New York, 2000. ISBN 0-387-98784-3.

Terry M Therneau. *A Package for Survival Analysis in R*, 2022. URL `https://CRAN.R-project.org/package=survival`. R package version 3.3-1.

Bradley Efron. The efficiency of cox's likelihood function for censored data. *Journal of the American statistical Association*, 72(359):557–565, 1977.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

Marian Tietz, Thomas J. Fan, Daniel Nouri, Benjamin Bossan, and skorch Developers. *skorch: A scikit-learn compatible neural network library that wraps PyTorch*, July 2017. URL `https://skorch.readthedocs.io/en/stable/`.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

Sebastian Pölsterl. scikit-survival: A library for time-to-event analysis built on top of scikit-learn. *J. Mach. Learn. Res.*, 21(212):1–6, 2020.

Jared L Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC medical research methodology*, 18(1):1–12, 2018.

Kaiwen Tan, Weixian Huang, Jinlong Hu, and Shoubin Dong. A multi-omics supervised autoencoder for pan-cancer clinical outcome endpoints prediction. *BMC Medical Informatics and Decision Making*, 20(3):1–9, 2020.

Remco R Bouckaert and Eibe Frank. Evaluating the replicability of significance tests for comparing learning algorithms. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 3–12. Springer, 2004.

Insuk Sohn, Jinseog Kim, Sin-Ho Jung, and Changyi Park. Gradient lasso for cox proportional hazards model. *Bioinformatics*, 25(14):1775–1781, 2009.

A note on standardisation. We note that our focus on standardisation is meant to apply mainly to the design of our benchmark and less to the results we provide below. Concretely, since model training always requires certain choices (*e.g.,* which hyperparameters the search space should include and what ranges), we believe this part of a benchmark cannot be perfectly standardised. We thus provide what we believe are reasonable default experiments and look forward to other researchers contributing their models (with potentially more diverse modeling approaches) to SurvBoard, while benefiting from our standardised preprocessing and benchmark design.

Breadth of considered models. We note that our experiments below serve two primary goals: first, presenting the first large-scale comparison of neural models and (group-aware) statistical models on a multi-omics survival analysis problem, and second, showcasing the benefits of SurvBoard relative to prior work, potentially revealing new findings now and in the future (*e.g.,* clinical-only models may outperform on some but not all projects, or that pan-cancer methods only provide limited benefit, relative to statistical models in standard settings). We would like to emphasize that our focus is on providing the best possible preprocessing and benchmark design, such that the community may provide their predictions to SurvBoard in the future and thus keep track of state of the art performance.

A Datasets

The Cancer Genome Atlas (TCGA)

Background. TCGA [Tomczak et al., 2015] is arguably one of the most impactful large-scale cancer projects in recent years. In addition to clinical data, TCGA provides various omics modalities from 33 cancer types and over 10000 patients, as well as whole slide images. TCGA has also had a large impact on the survival analysis community, with various papers focusing on survival prediction from both multi-omics [Hornung and Wright, 2019, Cheerla and Gevaert, 2019, Vale-Silva and Rohr, 2021] and gene expression [Ching et al., 2018, Kim et al., 2020] data.

Chosen modalities. In our work, we used TCGA for survival analysis with multi-omics data, in combination with clinical data. Since our goal was to benchmark the ability of different models to integrate high-dimensional multi-omics data, we chose the highest number of modalities for each dataset in TCGA, while still respecting our other selection criteria (see Section 3.2). Table S1 gives a full overview of all datasets in TCGA. We did not include whole slide image data in our benchmark, since it is non-trivial to integrate into statistical models, hindering comparisons between neural and statistical methods. Otherwise, we included all available modalities for a particular dataset in TCGA, including gene expression, protein expression, microRNA expression, mutation, DNA methylation, copy number variations, and clinical data.

Clinical data. To enable comparability between the standard, missing modality, and the pan-cancer settings (see Section 3.1 for details), we included the same clinical variables for all datasets on TCGA: Age at diagnosis, gender, race, tumor stage, clinical stage, and histological type. We note that we also included variables that were constant for a particular dataset (*e.g.,* we included gender for breast cancer) to more easily create a pan-cancer dataset. We note that within our single-cancer experiments, these constant features were removed. All categories indicating that a particular clinical variable was not recorded or not applicable (or similar) for a particular patient were set to NA.

Gene expression. Pan-cancer normalised gene expression data was obtained from *PANCANATLAS* [Weinstein et al., 2013]. Subsequently, we applied our standard preprocessing as detailed in Section 3.2. We further log transformed gene expression using $\text{expression} = \log_2(\text{expression} + 1)$.

Protein expression. Protein expression data was obtained from *PANCANATLAS* - aside from our standard preprocessing (see Section 3.2) which was applied to all modalities, protein expression was not modified further.

DNA methylation. We used *Merged 27K+450K Only* DNA methylation data from *PANCANATLAS*. No further preprocessing (aside from the standard steps n Section 3.2) was applied to this DNA methylation data.

Mutation. We used the *MAF* file available from *PANCANATLAS*, which we read and transformed into a *gene* $\times$ *patient* matrix using the *mutCountMatrix* function available in *maftools* to calculate the number of non-silent mutations per gene per patient [Mayakonda et al., 2018]. Genes that were non-mutated in a particular dataset were still included for uniformity in the pan-cancer setting. For

experiments on single-cancers, these constant genes were dropped. The data was then processsed as described in Section 3.2.

Copy number variations. For copy number variations, we used the pan-cancer *GISTIC2* [Mermel et al., 2011] thresholded file available on *Xenabrowser* [Goldman et al., 2020] which maps copy number variations to $\{-2, -1, 0, 1, 2\}$. Copy number variations were directly processed as described in Section 3.2.

MicroRNA. MicroRNA data were obtained from *PANCANATLAS*. In addition to our standard preprocessing, described in Section 3.2, we also applied a log-transformation to the microRNA expression using $\text{expression} = \log_2(\text{expression} + 1)$.

Miscellaneous. Throughout our preprocessing of TCGA, features that were constant for a specific dataset were kept. In our experiments, we excluded constant features at training time and recommend other researchers to do the same, unless they have a specific research question necessitating the usage of constant features.

If a particular patient on TCGA had multiple (primary) samples available for a specific modality (as may happen, *e.g.,* when multiple vials were used in sequencing), we used the lexicographically smaller vial (*i.e.,* if both vial A and vial B were available, we chose vial A) based on the sample barcode. If there were still multiple primary samples available for that patient, we followed the recommended sort replicate filter by the Broad Institute, choosing the sample with the highest lexicographical sort value (which chose the sample with a higher portion or plate number).[3]

International Cancer Genome Consortium (ICGC)

ICGC [Consortium et al., 2010] is a consortium focused on leading various large-scale cancer projects, such as TCGA. Of note, TCGA is a part of ICGC. Within our work, ICGC only refers to the non-TCGA datasets that were available from ICGC. Concretely, we included four datasets from ICGC, namely a lymphotic leukemia dataset from Spain (CLLE-ES), two pancreatic cancer datasets, one from California and from Australia (PACA-CA and PACA-AU) and a liver cancer dataset from Japan (LIRI-JP). All data for our ICGC datasets was obtained from the ICGC data portal. Table S1 provides an overview of all ICGC datasets we included in SurvBoard.

Chosen modalities. Based on our general requirements for all datasets (see Section 3.2), we only included clinical data, gene expression, and mutation for all of our datasets. We note that while the ICGC data portal includes copy number variation data for our datasets, the segment means (average log2 ratio of probes in that particular segment) were not available for any of our datasets, leading us to exclude copy numbers.

Clinical data. We included the following clinical variables for all of our datasets in ICGC (if available): Gender, age at diagnosis, tumor stage, cancer history among family members of the patient, as well as indicators for alcohol and smoking history.[4] All categories indicating that a particular clinical variable was not recorded or not applicable (or similar) for a particular patient were set to NA.

Gene expression. For PACA-AU, we used array-based gene expression data, since sequencing-based gene expression was available for only less than 100 patients. For all other datasets, we used sequencing-based gene expression. Sequencing-based gene expression was log-transformed using $\text{expression} = \log_2(\text{expression} + 1)$ while array-based gene expression was left untransformed as the data was not skewed. Aside from this, only our general preprocessing was performed on all gene expression data (see Section 3.2 for details).

Mutation. Mutation data was transformed to *MAF* format using the *icgcSimpleMutationToMAF* function from *maftools*. Afterward, mutation data was transformed into a gene by a patient matrix where for each gene of each patient, we counted the number of non-silent mutations using *mutCount-Matrix* from *maftools*. Genes without any mutations for a particular dataset were kept. Otherwise, no modality-specific preprocessing was performed for the mutation data of ICGC.

[3] Details available at: https://gdac.broadinstitute.org/runs/stddata__2014_01_15/samples_report/READ_Replicate_Samples.html.

[4] Although ICGC did not have a pan-cancer setting, we did not particularly find many differing clinical variables between our datasets and thus restricted ourselves to this set of common clinical variables.

Miscellaneous. Throughout our preprocessing of ICGC, features that were constant for a specific dataset were kept. In our experiments, we excluded constant features at training time and recommend other researchers to do the same, unless they have a specific research question necessitating the usage of constant features.

If a particular patient on ICGC had multiple (primary) samples or specimens available for a specific modality, we chose the specimen that was closest to the time of diagnosis and the sample that had been stored for the shortest amount of time. If two specimens were taken at identical times or two samples had been stored for the same amount of time or no information on sample/specimen time was available, we randomly chose one of the available samples/specimens.

Therapeutically Applicable Research To Generate Effective Treatments (TARGET)

The Therapeutically Applicable Research to Generate Effective Treatments (TARGET) Initiative GenomeOC [2021] is a study on childhood cancers that seeks to understand the molecular basis of pediatric malignancy through comprehensive molecular characterisation. The purpose of the study is to accelerate research in novel marker discovery and drug development. The TARGET project currently hosts omics information on 9 pediatric cancers. Following the selection criteria described in Section 3.2 we end up with 3 out of the 9 cancers in our benchmark study, namely, Acute Lymphoblastic Leukemia (ALL) Expansion Phase 2, Acute Myeloid Leukemia (AML), and Wilms Tumor (WT). All datasets are publicly available and can be accessed at `https://portal.gdc.cancer.gov/projects`. The datasets for this benchmark study, however, were accessed through the cBioPortal site Cerami et al. [2012], Gao et al. [2013] hosted by the Center for Molecular Oncology at Memorial Sloan Kettering Cancer Center.

Chosen modalities. Based on our general requirements for all datasets (see Section 3.2), we included clinical data, gene expression, copy number variations, and microRNA expression for TARGET-ALL and TARGET-WT cancers, and only clinical data, copy number variations, and DNA methylation data for TARGET-AML.

Clinical data. Clinical variables were selected such that there was no potential leakage of survival or genetic information. For example, information about treatment or karyotype mutations was not included. Similarly, numerical variables with missing values were also not included. Based on these criteria, we included gender, age at diagnosis, and race for all the datasets. Additionally, we included dataset-specific clinical variables that satisfied our criteria for inclusion: for TARGET-ALL – CNS_status, testicular_involvement, congenital_abnormality, for TARGET-AML – CNS_disease, chloroma, FAB, other_MLL, and for TARGET-WT – clinical_stage. All categories indicating that a particular clinical variable was not recorded or not applicable (or similar) for a particular patient were grouped into a single "NA" category.

Gene expression. For all datasets in TARGET, we used sequencing-based RPKM (Reads Per Kilobase of transcript per Million mapped reads) normalised gene expression values of 26136 genes. The data was preprocessed as detailed in Section 3.2 and finally log-transformed using expression $= \log_2(\text{expression} + 1)$ after median imputation.

Copy number variations. Putative copy-number calls were determined using *GISTIC2* [Mermel et al., 2011] for all datasets. The mapping is as follows: -2 = homozygous deletion; -1 = hemizygous deletion; 0 = neutral / no change; 1 = gain; 2 = high level amplification. Copy number variations were directly preprocessed as detailed in Section 3.2 without any further modality-specific preprocessing.

microRNA expression. The microRNA expression values accessed through cBioPortal for TARGET-ALL and TARGET-WT datasets were preprocessed as detailed in Section 3.2 and finally log-transformed using expression $= \log_2(\text{expression} + 1)$ after median imputation.

DNA methylation. For the TARGET-AML dataset, methylation beta-values from the HM27 platform were used. The DNA methylation values were directly preprocessed as detailed in Section 3.2 without any further modality-specific preprocessing.

B Performance metrics

For a particular patient i, let D_i denote their true event time, let C_i denote their right-censoring time and denote their observed time $T_i = \min(D_i, C_i)$. We denote the censoring indicator as

$\delta_i = \mathbb{1}\{T_i = D_i\}$ for the same patient i. Lastly, let $\hat{S}(t \mid X_i)$ be the predicted survival time for patient i at time t.

Below, we briefly detail the three performance metrics used throughout SurvBoard. We include one metric for discriminative performance, namely Antolini's concordance [Antolini et al., 2005] and two metrics measuring calibration, the Integrated Brier Score [Graf et al., 1999] and D-Calibration [Haider et al., 2020].[5] For a broader discussion on how discriminative and calibration measures differ, as well as the different *flavors* of calibration metrics, we refer the reader to Chapter 2 of Haider et al. [2020].

Antolini concordance. In survival analysis, a pair of samples is said to be concordant if the risk score of the event predicted by the model is lower for the sample who experienced the event at a later time, provided that the pair of samples is comparable. Various concordances may be defined which generally correspond to a (possibly weighted) ratio of all concordance pairs over all comparable pairs of subjects. Concordances thus fall into the $[0, 1]$ interval, where 1 corresponds to a perfect ordering, and 0.5 corresponds to a random ordering, or, equivalently, an ordering that does not differentiate between patients (such as a Kaplan-Meier).

While many recent works in multi-omics survival analysis (*e.g.,* (*e.g.,*) [Herrmann et al., 2021, Hornung and Wright, 2019, Cheerla and Gevaert, 2019]) rely on Harrell's Harrell et al. [1982] or Uno's concordance Uno et al. [2011], we evaluated Antolini's concordance [Antolini et al., 2005], a time-dependent concordance measure (Eq. (1)). Although Antolini's concordance exactly corresponds to Harrell's concordance for proportional-hazards models, it is not the case for other models such as RSF (see also Sonabend et al. [2022] for a discussion on why a comparison between different concordances should be avoided). Thus, we evaluated Antolini's concordance, primarily to enable SurvBoard to generalize to arbitrary models - models using non-proportional hazards formulations are generally unable to produce per-patient risk measures, which makes evaluating them using Harrell's or Uno's concordance impossible.

$$C_{\text{Antolini}} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \delta_i \mathbb{1}\{\hat{S}(T_i \mid X_i) < \hat{S}(T_i \mid X_j)\}}{\sum_{i=1}^{n} \sum_{j=1}^{n} \delta_i \mathbb{1}\{T_i < T_j\}} \tag{1}$$

Here, n denotes the total number of patients within a particular dataset.

We implemented Antolini's concordance using the default method provided by the *pycox* paper [Kvamme et al., 2019].[6]

Integrated Brier Score The Brier Score (BS), originally conceived for classification [Brier et al., 1950], was extended to right-censored survival data by Graf et al. [1999] (Eq. (2)). Since then, the integrated version, the Integrated Brier score (IBS) (Eq. (3)) has become a widely used for measuring the quality of survival predictions, both in genomic applications [Herrmann et al., 2021, Klau et al., 2018] and for low-dimensional clinical survival models [Zhong et al., 2021, Kvamme et al., 2019]. A lower IBS indicates better model performance.

$$BS(t) = \frac{1}{n} \sum_{i=1} \left[\frac{\hat{S}(t \mid X_i)^2 \mathbb{1}\{T_i \leq t \cap \delta_i = 1\}}{\hat{C}(T_i)} + \frac{(1 - \hat{S}(t \mid X_i))^2 \mathbb{1}\{T_i > t\}}{\hat{C}(t)} \right] \tag{2}$$

Here, $\hat{G}(t \mid X_i)$ is an estimate of the censoring survival function, *i.e.,* (*i.e.,*) $\hat{G}(t \mid X_i) = P(C_i > t \mid X_i)$ [Kvamme et al., 2019].

$$IBS = \frac{1}{b-a} \int_a^b BS(x) dx \tag{3}$$

We used the *pycox* implementation of IBS. In particular, we set a equal to the minimum of all test times in each split and b to the maximum of all test times in each split and considered a time grid of

[5] Technically, the IBS also (partially) measures discrimination.

[6] The default *pycox* implementation is termed the *adjusted* Antolini concordance and differs very slightly from the original paper [Kvamme et al., 2019, Antolini et al., 2005].

100 points between the minimum and maximum time to approximate the true integral. Further, we set $\hat{G}(t \mid X_i) = \hat{G}(t)$ to be the Kaplan-Meier estimator on the test set.

D-CAL D-Calibration (D-CAL) is a rather new metric focused on evaluating the distributional calibration of *individual survival curves*. We refer the reader to Haider et al. [2020] for the full algorithm, since it is beyond the scope of our work to reproduce here.

We computed D-Calibration using the *survival_evaluation* package provided by the paper authors [Haider et al., 2020]. Throughout our results, we report test statistics instead of p-values for better comparability, which implies that lower values signify better performance. We used D-CAL with its default parameter of 10 bins. In case this created missing values for a specific split of a particular model, we replaced the missing value with the mean D-CAL across all other splits for the same model.

C SurvBoard walk-through

Here, we provide a short example walk-through on our web service, SurvBoard, by submitting an example file to it. More detailed documentation is available under the documentation tab of the web service (Fig. C1).

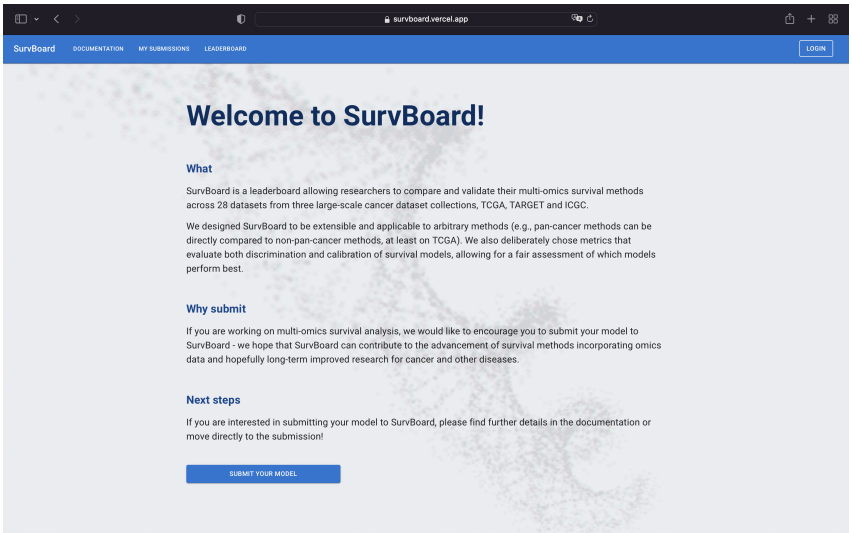

Figure C1: The welcome page of our web service provides further documentation on how to submit and other details.

C.1 Leaderboard

Our leaderboard was designed with a simple spreadsheet structure in mind - users may easily subset all models to only include a particular setting (see Section 3.1 for an overview of different settings considered in SurvBoard) or datasets from a particular project Fig. C2. To avoid overwhelming users, SurvBoard only displays one metric at a time, showing the mean of a metric across test splits and its standard deviation in parentheses, both rounded to three digits, per dataset. We also included an aggregated column that shows the pooled mean and standard deviation across datasets for a particular project. The public leaderboard is freely accessible, without a need to sign-up.

C.2 Sign-up

If users want to submit their models to SurvBoard, they are required to sign-up with a name, email, and institution/organisation (Fig. C3). We require these details in order to communicate with users and ensure authenticity such that repeated submissions to game SurvBoard or the usage of automated systems is not possible.

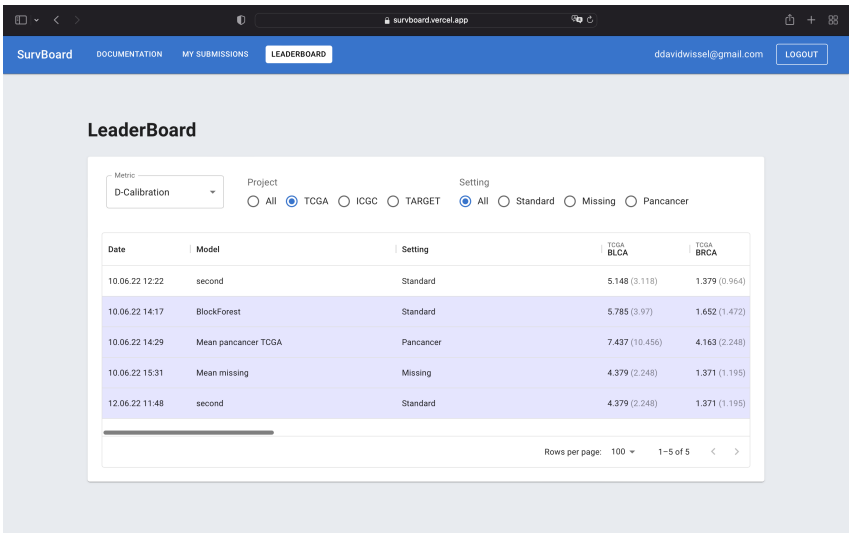

Figure C2: The SurvBoard leaderboard can be easily subset to different projects, settings and display one of three metrics (see Appendix B for an overview).

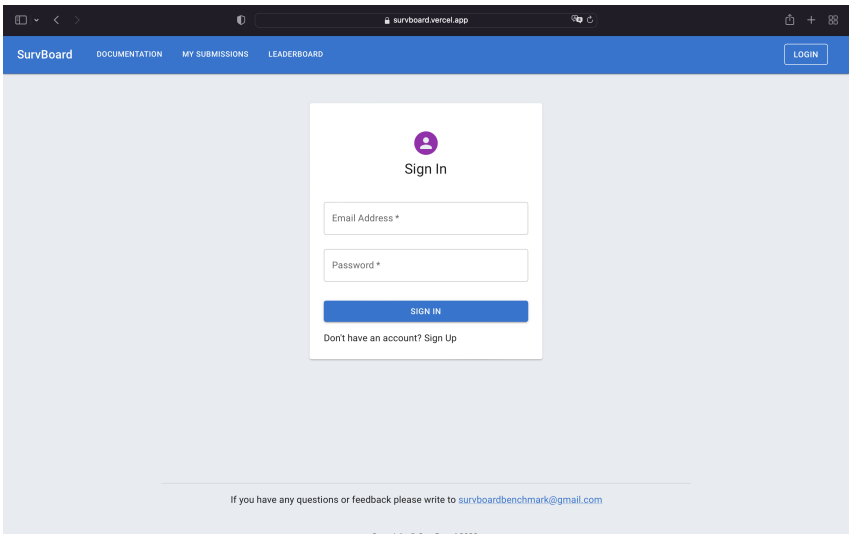

Figure C3: If users want to submit predictions, we require a registration, in order to prevent malicious usage.

C.3 Downloading data and splits

We provide users with the option of downloading all the datasets we used in our benchmark, including the splits for creating the test set. However, signing up is not necessary for simply downloading the data which can instead be accessed directly from here.

C.4 Submitting predictions

Once users have trained their respective models and made predictions in our required format (please refer to our web service documentation for details), model predictions may be uploaded through a simple three-step workflow on SurvBoard (Fig. C5 - Fig. C7).

After submission, local results (that is, results only available to the user who submitted them) will generally be available within minutes. We note that we require code submission (see Fig. C6) in the form of an online code repository for submissions that will be shown publicly. In case users opt

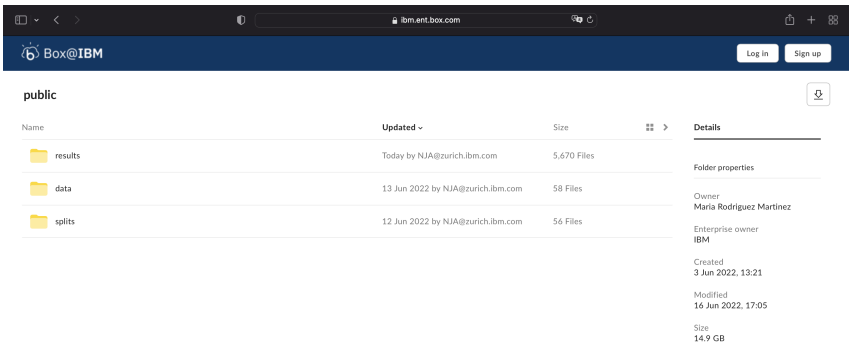

Figure C4: We publicly share all of our preprocessed datasets, including train and test splits, as well as the results including survival functions and performance metrics obtained during our benchmark. The data can also be retrieved directly from the respective source project and preprocessed using our code available on our Github repository (please see Appendix A for an overview of where each respective source file was obtained for a particular project).

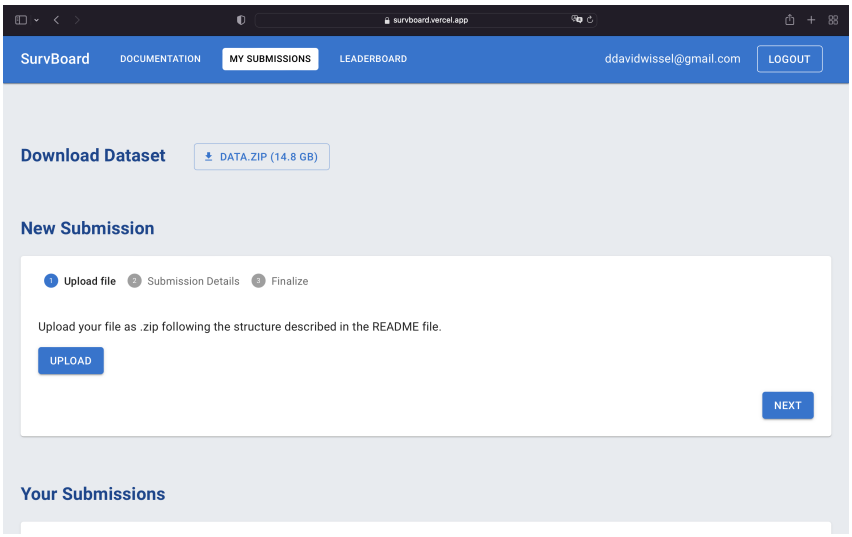

Figure C5: Initially, SurvBoard requires the user to upload a ZIP file containing model predictions for one or more datasets.

for public submission, by submitting a link to a code repository implementing their chosen model, results will be available on the public leaderboard after review for potential overfitting or of previous submission attempts to SurvBoard, which will generally take less than a week.

D Statistical models

Reproducibility. Results for all statistical reuslts may be reproduced using requisite scripts within our Github repository. For completeness, we provide information on which hardware each of the statistical models was run in the sections below. All of our models were interfaced using *mlr3* [Lang et al., 2019] and *mlr3proba* [Sonabend et al., 2021].

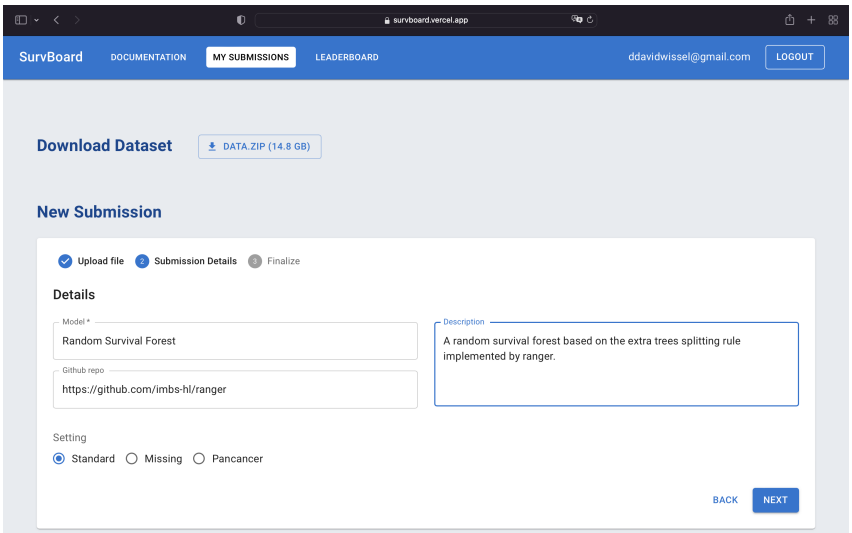

Figure C6: In the second step of submission, the user may give their model a name, and provide a description of what their model is based on. Further, if users want to opt for a public release of their model performance, we also require the submission of a code repository link.

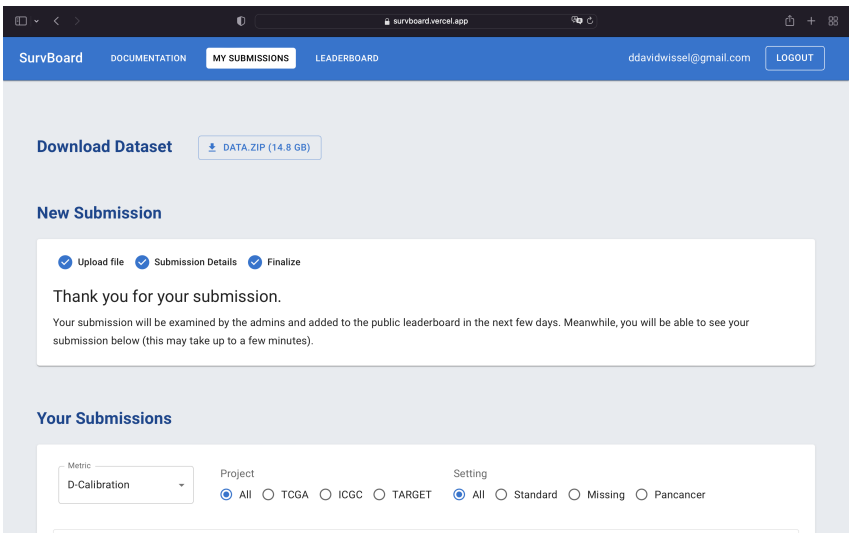

Figure C7: Lastly, users are informed that it may take a few days until their model performance is visible publicly - although they can immediately see their performance on SurvBoard locally.

Licenses. All considered statistical models are available on CRAN (with the exception of CoxBoost, which is available through Github). To the best of our knowledge, all statistical models used open source licenses at the time of our experiments. We refer to each of the statistical models for more details on specific licenses used.

D.1 Group-aware statistical models

Priority-Lasso. We included Priority-Lasso, a recently proposed method for survival prediction from multi-modal data, in our benchmark primarily due to its good performance in the benchmark of Herrmann et al. [2021]. We use the authors' official R package for implementation [Klau et al., 2018]. Specifically, we used the approach described by Herrmann et al. [2021] to choose the block order, first fitting an ℓ_2 penalized Cox model on variables that only belong to each of the considered modalities. Afterwards, we chose the first block to be the modality with the highest absolute mean

coefficients, the second block to be the modality with the second highest value, and so on. The modality-specific regularization hyperparameters were chosen based on the maximization of partial (log)-likelihood on the validation folds (*i.e.,* the test folds of our *inner* cross-validation, which is used for tuning) (*lambda.type="lambda.min"* and *type.measure="deviance"*). We used five-fold cross validation, stratified by the event indicator for choosing the hyperparameters for each block. Contrary to Herrmann et al. [2021], we also use five-fold cross validation to estimate the offsets - (*cvoffset=TRUE* and *cvoffsetnfolds=5*). Lastly, we dummy encoded categorical variables and standardised all input variables using *prioritylasso* (*i.e., standardize = TRUE*). All other settings of *prioritylasso* were left at their defaults. The search space for the regularization hyperparameter λ for both Priority-Lasso and the initial step to choose the priority order used the sequence approach in *glmnet* [Friedman et al., 2010, Simon et al., 2011], which chooses 100 values for λ between a completely sparse fit (for λ_{max}) and a minimum $\lambda_{\text{min}} = 0.01\lambda_{\text{max}}$. All experiments for Priority-Lasso were run on a cluster using AMD EPYC-Rome CPUs.

BlockForest. Since it was among the best performing models in the study of Herrmann et al. [2021], we also included BlockForest in our benchmark, a random forest variant adapted to group structured (*e.g.,* multi-omics) data. We use the authors' official R package for implementation [Hornung and Wright, 2019]. We used the function provided for hyperparameter tuning in the authors' package called *blockfor*, with the caveat that instead of considering 300 hyperparameter weight sets during tuning, we only considered a maximum of 100 due to computational constraints. The *BlockForest* package tuned hyperparameters using out-of-bag estimates of Harrell's concordance and chose the hyperpameter weight set which maximized the out-of-bag estimate of Harrell's concordance [Harrell et al., 1982]. All other parameters were left at their defaults. Categorical variables were passed to BlockForest directly and numerical covariates were not standardised. Due to execution times, experiments for TCGA datasets of BlockForest were run on a cluster using AMD EPYC-Rome CPUs, while experiments for ICGC and TARGET datasets were run on a 2020 M1 MacBook Pro.

D.2 Group-naive statistical models

Lasso. We included a standard ℓ_1 penalized Cox regression model as a statistical baseline that does not take the group structure of the multi-omics data into account. We implemented the Lasso using *glmnet* [Friedman et al., 2010, Simon et al., 2011]. We tuned the regularisation hyperparameter λ using (nested) five-fold cross validation using the *cv.glmnet* function. The optimal λ was selected based on which λ provided the highest partial-(log)-likelihood on the validation folds (*i.e.,* the test folds of our *inner* cross-validation, which is used for tuning) (*i.e., type.measure="deviance"* and *lambda.type="min"* in *cv.glmnet*). The search space for the regularization hyperparameter λ for the Lasso used the sequence approach in *glmnet* [Friedman et al., 2010, Simon et al., 2011], which chooses 100 values for λ between a completely sparse fit (for λ_{max}) and a minimum $\lambda_{\text{min}} = 0.01\lambda_{\text{max}}$. For cross validation, we stratified the folds on the event indicator. For the Lasso, we dummy encoded categorical variables and standardised all input variables using *glmnet* (*i.e., standardize = TRUE*). All experiments for the Lasso were run on a cluster using AMD EPYC-Rome CPUs.

RSF. Further, we included a Random Survival Forest (RSF) implementation. Specifically, our RSF was a quasi-ablation study of BlockForest. Since the official BlockForest package uses default RSF hyperparameters (except for group-specific BlockForest hyperparameters), we mirrored these choices in our inclusion of RSF. Specifically, we used the *ranger* [Wright and Ziegler, 2017] package, with 2000 trees in the forest (*num.trees = 2000*), the *extratrees* split rule (*splitrule = "extratrees"*), and $\sqrt{p}$ variables at each split point (*mtry = NULL* since this is the default of *ranger*), where p is the dimensionality of a particular input dataset. All other options in the *ranger* package were left at their defaults (mirroring BlockForest). Thus, RSF employed the exact same settings as BlockForest, except for BlockForest specific hyperparameters, which take the group structure into account. Categorical variables were passed to *ranger* directly and numerical covariates were not standardised.

Coxboost. Lastly, we included a likelihood boosting model that optimizes the negative partial (log)-likelihood (*i.e.,* the Cox loss), mirroring Herrmann et al. [2021]. We used the *CoxBoost* package for implementation [Binder and Schumacher, 2008], relying on it for optimising the number of boosting steps using five-fold cross validation stratified by the event indicator.[7] Specifically, we used the *cv.CoxBoost* function with default parameters to estimate the optimal number of boosting

[7] At the time of our experiments, *CoxBoost* was not available on either CRAN or Bioconductor - thus, we used the current version of the projects' official Github repository.

steps, between 0 and 100.[8] The optimal value was chosen based on which maximized the partial (log)-likelihood across the validation folds (*i.e.,* the test folds of our *inner* cross-validation, which is used for tuning). For CoxBoost, we dummy encoded categorical variables and standardised all input variables using *CoxBoost* (*i.e., standardize = TRUE*). All experiments for CoxBoost were run on a cluster using AMD EPYC-Rome CPUs.

D.3 Baselines

Clinical-only Cox Proportional Hazards. As a baseline, we included an unregularized Cox proportional hazard model that was trained only on clinical data. We used the implementation of the *survival* R package [Terry M. Therneau and Patricia M. Grambsch, 2000, Therneau, 2022] with the *Efron* approximation for handling ties [Efron, 1977]. All experiments for the Cox model were run on a 2020 M1 MacBook Pro.

D.4 Miscellaneous

We note that for all statistical methods based on the Cox model, except CoxBoost, we used the *Breslow* estimator of the *survival* R package to calculate survival function estimates. Survival functions for CoxBoost were calculated using package-internal prediction functions of the *CoxBoost* R package. In addition, constant features were removed at training time for all statistical models (including the clinical-only Cox proportional hazards baseline). Copy number was treated as numerical (as opposed to categorical) for all statistical models.

E Neural models

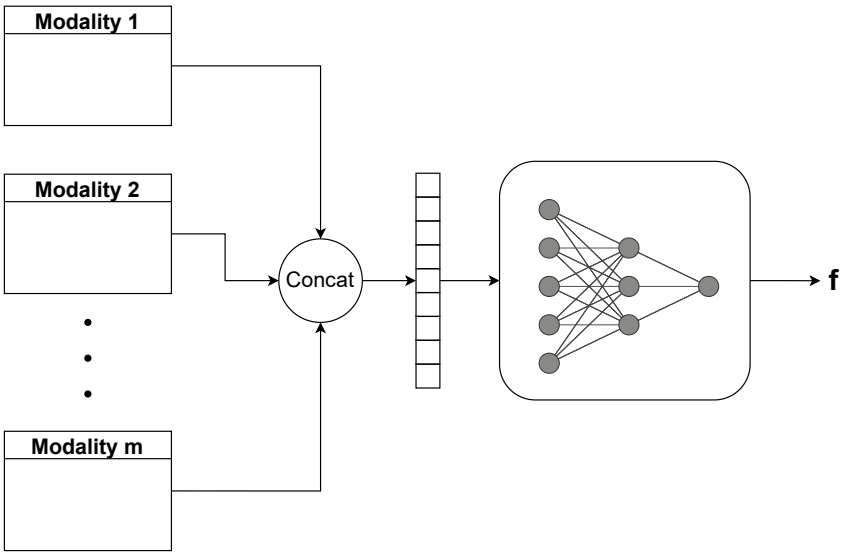

Figure E1: Architecture of the fully connected model. Our fully connected model was based on early fusion and learns a joint log hazard f based on the full input data, without incorporating the group structure in any way.

Reproducibility. We followed best practices in *Pytorch* for reproducibility. Results may be reproduced using requisite scripts within our Github repository. However, we note that hardware differences or multi-threading may cause results to be similar but not exact. For completeness, we provide information on which hardware each of the neural models was run in the sections below.

[8] We note that we briefly tried also estimating the optimal penalty hyperparameter using the *optimCoxBoostPenalty*, but found it to experience numerical and other issues consistently, and thus relied on the default penalty hyperparameter, as previous has done to the best of our knowledge [Herrmann et al., 2021].

Licenses. All neural models were implemented in *Pytorch* [Paszke et al., 2019] and *skorch* [Tietz et al., 2017] both of which are open source and freely available on Github, via conda, or via pip. We refer to the respective code repositories of *Pytorch* and *skorch* for details on their respective licenses.

Miscellaneous. Since there is little agreement on minor architectural decisions such as how many hidden layers to fit for each modality (or overall) in the multi-omics survival community, we abstracted these away and focused instead on the key modeling choices and thus research foci.

All models were fit using batch normalisation [Ioffe and Szegedy, 2015] and ReLUs [Nair and Hinton, 2010] as activation functions. For optimisation, we used the negative partial log-likelihood with the Breslow approximation for tied time points (Eq. (4)). In addition, all models were trained using the Adam [Kingma and Ba, 2015] optimiser with an initial learning rate of 0.001. We used 10% of the training set as a validation set for early stopping (using patience of 10 and restoring the model after the epoch with the lowest validation loss) and learning rate scheduling (reducing the learning rate by a factor of 10 whenever validation loss did not improve for three epochs). Due to the small size of our datasets, we only tuned dropout probability for all considered neural networks, using five-fold cross-validation stratified by the event indicator and choosing the dropout probability which yielded the lowest negative partial log-likelihood across validation folds. Table E1 gives an overview of all hyperparameters for our neural models.

$$L(f) = \sum_{i:\delta_i=1} \left(f(X_i) - \log \left(\sum_{j:T_j \geq T_i} \exp(f(X_j)) \right) \right) \tag{4}$$

Here, we denote the features of each patient i by X_i, the (possibly unknown) true event time as D_i, the right-censoring time as C_i, the observed time as $T_i = \min(D_i, C_i)$ and the censoring indicator as $\delta_i = \mathbb{1}\{T_i = D_i\}$ for a particular patient i. Further, f denotes a neural network with learnable parameters used for predicting the log hazard.

For all neural models, we removed constant features at training time, one-hot encoded categorical variables and standardised all covariates using *StandardScaler* from *scikit-learn* [Pedregosa et al., 2011].[9] The survival functions for all neural models were estimated using the Breslow estimator implementation from *scikit-survival* [Pölsterl, 2020].

E.1 Group-naive neural network

First, we included a group-naive neural network, roughly equivalent to the architectures of *DeepSurv* [Katzman et al., 2018] or *cox-nnet* [Ching et al., 2018] (Fig. E1), that is a fully connected neural network. We refer to this network as **fully connected** throughout the text. Our fully connected network used two hidden layers with 128 and 64 hidden nodes respectively. All experiments for the fully connected model were run on a cluster using AMD EPYC-Rome CPUs. The fully connected network used the negative partial-log likelihood based on the Cox model (Eq. (4)) with no further components, that is, $L_{\text{Fully connected}} = L(f)$ (Eq. (4)), where f is a neural network predictor of the log hazard.

E.2 Group-aware neural network

Mean pooling In addition, we included a group-aware neural network that relies on a similar mean-pooling architecture as the model of [Cheerla and Gevaert, 2019], although we did not implement the similarity loss used by the same authors.[10] We refer to our mean-pooling architecture as **Mean pooling** throughout the text. To encourage our mean pooling neural model to learn survival relevant modality-specific hidden representations, we also predicted log hazard ratios from the hidden layer of each modality (Fig. E2 and Eq. (5)), similar to other previous works [Lee and van der Schaar, 2021,

[9] Categories that were present in a particular test split but not the training split were ignored (*i.e.,* all columns for that particular one-hot encoded category were set to zero for unknown categories, using *handle_unknown=ignore* in *OneHotEncoder*).

[10] We note that we briefly experimented with an alternative pooling technique, but found other techniques to achieve similar results (data not shown). In addition, since Vale-Silva and Rohr [2021] also experimented with various pooling and other integration techniques of modality-specific representations and found no major differences, we focused on the overall choice of incorporating group structure and left the exploration of different integration techniques to future work.

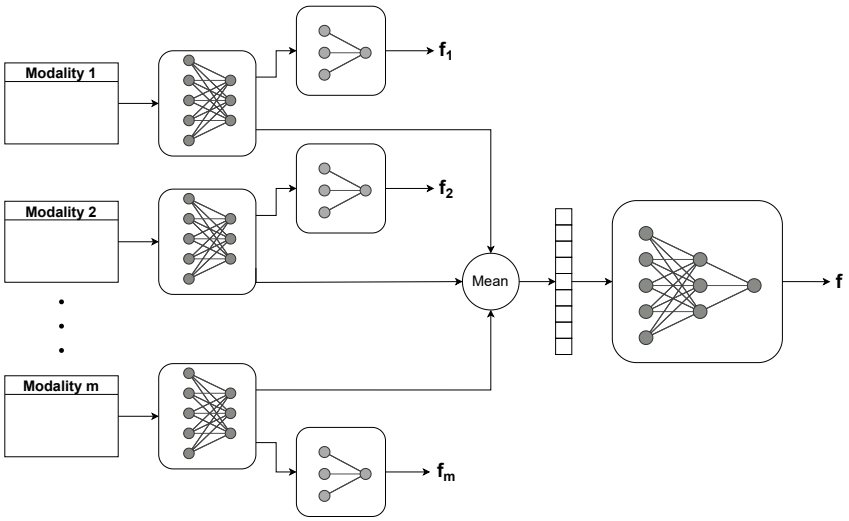

Figure E2: Architecture of the mean pooling neural model. Our mean pooling model was inspired by works such as Cheerla and Gevaert [2019] and Cheerla and Gevaert [2019]. The model learns modality-specific representations which are integrated using mean pooling after which the joint log hazard f is predicted. In addition, we added modality-specific log hazard predictions, to encourage the model to learn survival-relevant information within each modality []

Tan et al., 2020]. Throughout the main text, we set $\alpha = 1$, although we also present an ablation in which we considered $\alpha = 0$ (see Appendix G).

Here, m denotes the total number of modalities, f_1, ..., f_m denote modality-specific predictors of the log hazard while f denotes the joint multi-omics predictor of the log hazard.

$$\begin{aligned} L(f, f_1, ..., f_m)_{\text{Mean pooling}} = & \sum_{i:\delta_i=1} \left(f(X_i) - \log \left(\sum_{j:T_j \geq T_i} \exp(f(X_j)) \right) \right) \\ & + \sum_{m=1}^{M} \sum_{i:\delta_i=1} \left(f_m(X_i) - \log \left(\sum_{j:T_j \geq T_i} \exp(f_m(X_j)) \right) \right) \end{aligned} \tag{5}$$

Concretely, for our mean-pooling architecture, we fit one network per modality, composed of one hidden layer with 128 nodes and activation, followed by a second hidden layer of 64 nodes with no activation. Afterward, the 64 dimensional embeddings from each modality were averaged and input into another hidden layer of 32 nodes (with activation) before predicting the log-partial hazard (Fig. E2). All experiments, except two, for the mean pooling neural model were run on a cluster with AMD EPYC-Rome CPUs. We encountered memory issues related to parallelisation on our cluster when running the pancreatic dataset from California (PACA-CA) for the missing setting and the leukemia dataset from Spain (CLLE-ES) in the standard setting. Thus, we ran these two datasets on a 2020 M1 MacBook Pro.

E.3 Settings for neural networks

For simplicity, and since previous methods [Cheerla and Gevaert, 2019, Vale-Silva and Rohr, 2021] have only implemented non-standard settings (see Section 3.1 for an overview of all settings) for group-aware neural networks, we did the same for our mean pooling architecture.

Missing modality setting. In the setting that included samples with missing modalities, we used samples with missing modalities as additional training data. In addition, we implemented multi-modal dropout [Cheerla and Gevaert, 2019], with a fixed probability of 25%.

Table E1: Overview of hyperparameters of all neural models in our benchmark.

Hyperparameter	Tuned or fixed	Range
Initial learning rate	Fixed	0.001
Batch size	Fixed	128
Early stopping patience	Fixed	10
Learning rate scheduler patience	Fixed	3
Learning rate scheduler reduction factor	Fixed	10
α (Mean pooling specific)	Fixed	1.0
Dropout probability	Tuned	{0, 0.1, 0.2, 0.3, 0.4, 0.5}

Pan-cancer data setting. For the setting incorporating data from multiple cancer types on TCGA, we pooled data across cancers before training a joint model. In addition, we included the cancer type of each sample as an additional clinical variable, similar to Vale-Silva and Rohr [2021]. We trained our pan-cancer neural model using multi-modal dropout with a fixed probability of 25% since our datasets included samples that did not have all modalities present in some of the cancer types.

F Recommendations

- **Preprocessing**: Do your best to simulate a clinical environment. If including molecular data, try to use (only) tissue samples that are as close to primary tissue as possible. In general, aim to only use variables available at or close to the time of diagnosis.
- **Dataset selection**: Unless you are working on a specific biological problem, aim to include as many diverse datasets as possible, ideally from different projects, such as TARGET, ICGC or others.
- **Test split selection**: Use cross-validation, ideally stratified on the event indicator, especially for small datasets. Be careful not to choose K too large in K-fold cross-validation, since this will reduce the number of events in your test splits. This will in turn increase variance in many survival metrics calculations, which depend mostly on non-censored samples. Repeat your cross-validation at least two or three times, ideally five or ten, depending on the size of your dataset.
- **Ensuring comparability**: Comparability to past and future work needs to be ensured, otherwise your (potential) new SOTA will not be used by people who cannot directly compare to your model. This means that if you are using samples with missing modalities or training on multiple cancer types, you should choose your train and test splits such that they are also usable by models not using these settings (concretely, use missing modality samples as additional training data as opposed to including them in the test sets).
- **Reproducible hyperparameter tuning**: In order to make it easy for other researchers to use your model, provide a reproducible hyperparameter tuning strategy. We found several works in which the authors optimised hyperparameters by hand - making it difficult to deploy the model in question to other datasets since the original hyperparameter tuning is impossible to emulate. Thus, we recommend authors perform automatic hyperparameter tuning, ideally in a dedicated function, ensuring that their model is adequately tuned in all cases.
- **Performance metrics**: To evaluate your survival model from various angles (especially since some models tend to discriminate well but not be well calibrated), include at least one discriminative metric (some kind of concordance) as well as one calibration metric (usually the IBS). You may also want to include newer measures of distributional calibration (*e.g.,* D-CAL [Haider et al., 2020]) since its results may at times differ significantly from the IBS.
- **Reporting results**: When reporting results, aim to provide as granular information to the reader as possible. Specifically, be very careful when aggregating performance metrics across datasets, as this may mask important variance between datasets. Ideally, report performance *per* dataset, combined with a statistical test such as the *corrected repeated k-fold cv test* when performing repeated cross-validation [Bouckaert and Frank, 2004]. If performance measures are being aggregated across datasets, provide an appropriate requisite statistical test such as the Wilcoxon signed rank test (when comparing one model to another model) or the Friedman test when comparing all models to each other [Demšar, 2006].

G Supplementary experiments

G.1 Analysing the impact of α for the mean pooling architecture

Table G1: Pooled mean performance metrics stratified by project for the mean-pooling architecture Fig. E2 with $\alpha = 1$ and $\alpha = 0$ (Eq. (4)). Standard deviation shown in parentheses. On TCGA, values not significantly worse than best are bold-faced per performance metric based on Wilcoxon signed-rank tests (p $\geq$ 0.05). $\uparrow$ signifies a metric for which higher is better, $\downarrow$ signifies a metric for which lower is better.

Model	Setting	Project	Antolini concordance ($\uparrow$)	IBS ($\downarrow$)	D-CAL ($\downarrow$)
Mean pooling ($\alpha = 1$)	Standard	TCGA	**0.602 (0.129)**	**0.192 (0.042)**	**4.655 (4.062)**
Mean pooling ($\alpha = 0$)	Standard	TCGA	**0.601 (0.128)**	0.197 (0.046)	5.718 (6.28)
Mean pooling ($\alpha = 1$)	Standard	ICGC	0.623 (0.102)	0.199 (0.072)	10.792 (13.657)
Mean pooling ($\alpha = 0$)	Standard	ICGC	0.634 (0.105)	0.202 (0.071)	10.757 (12.938)
Mean pooling ($\alpha = 1$)	Standard	TARGET	0.614 (0.099)	0.207 (0.026)	5.217 (5.385)
Mean pooling ($\alpha = 0$)	Standard	TARGET	0.643 (0.094)	0.205 (0.03)	5.519 (5.386)

When ablating α for the mean pooling intermediate fusion neural architecture, we found that while removing the modality-specific survival losses tended to increase discriminative performance on ICGC and TARGET (Table G1), discriminative performance on TCGA was virtually unchanged, although calibration performance in terms of the IBS and D-CAL was significantly worse for the model with $\alpha = 0$, that is, the model **without** modality-specific survival losses on TCGA. Future work is needed to determine the optimal loss function for multi-omics survival models - our results indicate, however, that adding additional modality-specific survival losses, as some works have opted to do [Lee and van der Schaar, 2021, Tan et al., 2020], tended not to help consistently.

H Maintenance plan

SurvBoard is currently deployed in the Cloud on a monthly subscription basis. The service uses common deployment patterns such as containers and SQL datastores and so fits the definition of a hybrid cloud application. The service can be migrated to other vendors if and when needed. The webservice deployment fees are currently covered by internal lab funds. Security patches, package updates and similar tasks will be handled by the paper authors regardless of the deployment environment.

The user provides us with an organisational email, which is public knowledge, to be able to create a profile and map their submission correctly. The results from the user's model are used to compute a score and place them on the leaderboard after which the result files are deleted immediately. We only store the user email and the final leaderboard scores for display. Code is only collected via Github repository links, *i.e.,*, all code submitted by users is anyway public on the respective Github repository under the respective license.

The datasets are available on Box `https://ibm.box.com/v/survboard-meta` as well as Kaggle `https://www.kaggle.com/datasets/jnikita/survboard` for additional discoverability. The two co-first authors will share joint responsibility of the web service maintenance for the next three years; after which responsibility will be handled jointly by the appropriate members of one of the two labs (in case one or both of the joint first authors are no longer in academia).

I Supplementary tables

Table S1: Overview of all datasets considered in our benchmark. GEX denotes gene expression, CNV denotes copy number variation, and RPPA denotes protein expression. Modality columns denote the dimensionality of that modality in a specific dataset. n denotes number of samples and e denotes number of events. Complete and incomplete denote the complete modality and incomplete modality sets respectively.

Project	Cancer	Clinical	GEX	Mutation	Methylation	CNV	RPPA	miRNA	Modalities	Dimensionality	n (complete)	e (complete)	n (incomplete)	e (incomplete)
TCGA	BLCA	6	20531	19687	22584	24776	189	743	7	88516	330	149	80	31
TCGA	BRCA	6	20531	19687	22591	24776	190	743	7	88524	771	103	312	48
TCGA	COAD	6	17507	19687	22583	24776	191	743	7	85493	289	64	153	34
TCGA	ESCA	6	19076	19687	22543	24776	193	743	7	87024	123	44	62	33
TCGA	HNSC	6	20531	19687	22589	24776	191	743	7	88523	202	122	325	101
TCGA	KIRC	6	20531	19687	22564	24776	191	743	7	88498	312	88	223	89
TCGA	KIRP	6	20531	19687	22580	24776	194	743	7	88517	203	29	85	15
TCGA	LGG	6	20531	19687	22588	24776	190	743	7	88521	412	95	99	30
TCGA	LUAD	6	20531	19687	22587	24776	189	743	7	88519	340	136	169	47
TCGA	PAAD	6	20531	19687	22538	24776	190	743	7	88471	109	61	75	39
TCGA	SARC	6	20531	19687	22533	24776	193	743	7	88469	194	70	67	29
TCGA	SKCM	6	20531	19687	22579	-	-	-	4	62803	102	29	352	185
TCGA	STAD	6	19076	19687	22577	24776	193	743	7	87058	304	126	113	42
TCGA	UCEC	6	17507	19687	22598	24776	191	743	7	85508	395	63	151	28
TCGA	OV	6	19076	19687	22521	24776	191	743	7	87000	166	98	416	250
TCGA	LIHC	6	20531	19687	22579	24776	190	743	7	88512	157	80	214	52
TCGA	LUSC	6	20531	19687	22591	24776	189	743	7	88523	288	117	209	98
TCGA	LAML	6	16765	-	22594	24776	-	743	5	64884	139	86	34	22
TCGA	CESC	6	20030	13181	22565	24776	193	737	7	81488	141	24	153	47
TCGA	GBM	6	20531	19687	-	24776	-	-	4	65000	144	114	451	377
TCGA	READ	6	17507	19687	22582	24776	-	743	6	85301	125	17	39	9
ICGC	CLLE	6	100090	3116	-	-	-	-	3	103212	294	46	230	34
ICGC	PACA-AU	4	47265	5204	-	-	-	-	3	52473	236	147	214	140
ICGC	PACA-CA	6	65802	6041	-	-	-	-	3	71849	162	128	63	36
ICGC	LIRI	4	22848	6828	-	-	-	-	3	29680	229	42	31	4
TARGET	ALL	6	23630	-	-	20641	-	1407	4	45684	108	66	1427	188
TARGET	AML	7	-	-	11719	19524	-	-	3	31250	178	55	708	263
TARGET	WT	4	22263	-	-	20641	-	1430	4	44338	119	48	533	66

Table S2: Tissue sites and full names of all cancer abbreviations used throughout our benchmark.

Project	Abbreviation	Tissue	Full name
TCGA	BLCA	Bladder	Bladder Urothelial Carcinoma
TCGA	BRCA	Breast	Breast invasive carcinoma
TCGA	COAD	Colon	Colon adenocarcinoma
TCGA	ESCA	Esophagus	Esophageal carcinoma
TCGA	HNSC	Head and Neck	Head and Neck squamous cell carcinoma
TCGA	KIRC	Kidney	Kidney renal clear cell carcinoma
TCGA	KIRP	Kidney	Kidney renal papillary cell carcinoma
TCGA	LGG	Brain	Brain Lower Grade Glioma
TCGA	LUAD	Lung	Lung adenocarcinoma
TCGA	PAAD	Pancreas	Pancreatic adenocarcinoma
TCGA	SARC	Bones and soft tissue	Sarcoma
TCGA	SKCM	Skin	Skin Cutaneous Melanoma
TCGA	STAD	Stomach	Stomach adenocarcinoma
TCGA	UCEC	Uterus	Uterine Corpus Endometrial Carcinoma
TCGA	OV	Ovaries	Ovarian serous cystadenocarcinoma
TCGA	LIHC	Liver	Liver hepatocellular carcinoma
TCGA	LUSC	Lung	Lung squamous cell carcinoma
TCGA	LAML	Blood forming tissue	Acute Myeloid Leukemia
TCGA	CESC	Cervix	Cervical squamous cell carcinoma and endocervical adenocarcinoma
TCGA	GBM	Brain and spine	Glioblastoma multiforme
TCGA	READ	Rectum	Rectum adenocarcinoma
ICGC	CLLE	Blood forming tissue	Chronic Lymphocytic Leukemia
ICGC	PACA-AU	Pancreas	Pancreatic Cancer
ICGC	PACA-CA	Pancreas	Pancreatic Cancer
ICGC	LIRI	Liver	Liver Cancer
TARGET	ALL	White blood cells	Acute lymphoblastic leukemia
TARGET	AML	Bone marrow	Acute myeloid leukemia
TARGET	WT	Kidney	Wilms tumor

Table S3: Number of failed model iterations per model in our main benchmark results, stratified by dataset. We found that none of our considered models ever failed, except the Lasso, which failed frequently, presumably due to attempted matrix inversions of highly correlated data (see *e.g.,* [Sohn et al., 2009]) - future work should thus consider gradient-based Lasso models to avoid model failures due to multi-collinearity.

Dataset	Lasso	RSF	BlockForest	Fully connected	Mean pooling (Standard)	Mean pooling (Missing)	Mean pooling (Pan-cancer)
ALL	-	0	0	0	0	0	-
AML	5	0	0	0	0	0	-
BLCA	1	0	0	0	0	0	0
BRCA	-	0	0	0	0	0	0
CESC	12	0	0	0	0	0	0
CLLE-ES	-	0	0	0	0	0	-
COAD	2	0	0	0	0	0	0
ESCA	21	0	0	0	0	0	0
GBM	6	0	0	0	0	0	0
HNSC	8	0	0	0	0	0	0
KIRC	-	0	0	0	0	0	0
KIRP	-	0	0	0	0	0	0
LAML	2	0	0	0	0	0	0
LGG	-	0	0	0	0	0	0
LIHC	-	0	0	0	0	0	0
LIRI-JP	3	0	0	0	0	0	-
LUAD	1	0	0	0	0	0	0
LUSC	20	0	0	0	0	0	0
OV	16	0	0	0	0	0	0
PAAD	20	0	0	0	0	0	0
PACA-AU	1	0	0	0	0	0	-
PACA-CA	10	0	0	0	0	0	-
READ	19	0	0	0	0	0	0
SARC	4	0	0	0	0	0	0
SKCM	21	0	0	0	0	0	0
STAD	11	0	0	0	0	0	0
UCEC	1	0	0	0	0	0	0
WT	5	0	0	0	0	0	-

Table S4: Antolini's concordance per dataset on TCGA, stratified by setting. Standard deviation shown in parantheses. Higher values imply better performance.

Model	Dataset	Missing modality samples	Pan-cancer	Standard
Mean	BLCA	0.636 (0.06)	0.648 (0.036)	0.628 (0.059)
Mean	BRCA	0.648 (0.064)	0.659 (0.055)	0.641 (0.081)
Mean	CESC	0.632 (0.113)	0.599 (0.176)	0.54 (0.149)
Mean	COAD	0.639 (0.072)	0.634 (0.088)	0.597 (0.093)
Mean	ESCA	0.573 (0.092)	0.529 (0.141)	0.488 (0.084)
Mean	GBM	0.567 (0.066)	0.528 (0.077)	0.543 (0.078)
Mean	HNSC	0.581 (0.067)	0.588 (0.061)	0.595 (0.058)
Mean	KIRC	0.745 (0.047)	0.713 (0.06)	0.719 (0.044)
Mean	KIRP	0.813 (0.101)	0.829 (0.127)	0.78 (0.105)
Mean	LAML	0.637 (0.054)	0.553 (0.079)	0.589 (0.085)
Mean	LGG	0.822 (0.038)	0.807 (0.065)	0.834 (0.036)
Mean	LIHC	0.582 (0.071)	0.643 (0.064)	0.584 (0.082)
Mean	LUAD	0.614 (0.063)	0.643 (0.044)	0.614 (0.055)
Mean	LUSC	0.536 (0.062)	0.588 (0.074)	0.511 (0.074)
Mean	OV	0.577 (0.06)	0.578 (0.065)	0.57 (0.062)
Mean	PAAD	0.549 (0.118)	0.574 (0.101)	0.513 (0.102)
Mean	READ	0.461 (0.2)	0.621 (0.191)	0.517 (0.187)
Mean	SARC	0.676 (0.077)	0.624 (0.098)	0.647 (0.074)
Mean	SKCM	0.539 (0.16)	0.56 (0.182)	0.469 (0.144)
Mean	STAD	0.556 (0.042)	0.581 (0.074)	0.542 (0.046)
Mean	UCEC	0.705 (0.077)	0.759 (0.083)	0.712 (0.07)

Table S5: Integrated Brier score per dataset on TCGA, stratified by setting. Standard deviation shown in parantheses. Lower values imply better performance.

Model	Dataset	Missing modality samples	Pan-cancer	Standard
Mean	BLCA	0.208 (0.025)	0.214 (0.031)	0.217 (0.029)
Mean	BRCA	0.192 (0.044)	0.181 (0.025)	0.199 (0.048)
Mean	CESC	0.208 (0.048)	0.208 (0.051)	0.194 (0.016)
Mean	COAD	0.213 (0.043)	0.209 (0.038)	0.226 (0.046)
Mean	ESCA	0.219 (0.035)	0.217 (0.04)	0.217 (0.027)
Mean	GBM	0.153 (0.02)	0.178 (0.062)	0.151 (0.019)
Mean	HNSC	0.206 (0.016)	0.216 (0.027)	0.217 (0.029)
Mean	KIRC	0.159 (0.022)	0.177 (0.023)	0.17 (0.017)
Mean	KIRP	0.195 (0.122)	0.139 (0.052)	0.138 (0.032)
Mean	LAML	0.188 (0.025)	0.218 (0.027)	0.2 (0.024)
Mean	LGG	0.188 (0.061)	0.167 (0.029)	0.167 (0.049)
Mean	LIHC	0.21 (0.027)	0.207 (0.035)	0.204 (0.025)
Mean	LUAD	0.2 (0.032)	0.181 (0.027)	0.201 (0.028)
Mean	LUSC	0.221 (0.032)	0.218 (0.031)	0.216 (0.019)
Mean	OV	0.161 (0.018)	0.168 (0.016)	0.197 (0.04)
Mean	PAAD	0.217 (0.057)	0.21 (0.049)	0.208 (0.021)
Mean	READ	0.157 (0.042)	0.148 (0.04)	0.154 (0.042)
Mean	SARC	0.221 (0.053)	0.234 (0.051)	0.2 (0.036)
Mean	SKCM	0.189 (0.039)	0.174 (0.032)	0.199 (0.055)
Mean	STAD	0.216 (0.023)	0.216 (0.034)	0.219 (0.023)
Mean	UCEC	0.144 (0.019)	0.134 (0.033)	0.142 (0.024)

Table S6: D-Calibration test statistic per dataset on TCGA, stratified by setting. Standard deviation shown in parantheses. Lower values imply better performance.

Model	Dataset	Missing modality samples	Pan-cancer	Standard
Mean	BLCA	4.379 (2.294)	7.437 (10.671)	4.905 (3.09)
Mean	BRCA	1.371 (1.22)	4.163 (2.294)	2.306 (2.331)
Mean	CESC	1.815 (1.214)	2.027 (1.366)	1.647 (0.924)
Mean	COAD	2.135 (1.16)	2.331 (1.562)	3.231 (3.125)
Mean	ESCA	4.235 (2.509)	3.679 (2.32)	5.119 (2.821)
Mean	GBM	7.996 (3.982)	15.185 (25.353)	9.445 (5.301)
Mean	HNSC	6.719 (3.835)	12.302 (12.009)	7.779 (5.594)
Mean	KIRC	4.138 (1.967)	4.162 (2.576)	3.833 (2.027)
Mean	KIRP	2.256 (1.588)	1.904 (1.316)	1.232 (0.805)
Mean	LAML	7.629 (3.56)	11.993 (11.047)	7.37 (3.597)
Mean	LGG	4.197 (7.573)	6.612 (3.219)	3.913 (3.475)
Mean	LIHC	6.465 (3.148)	7.848 (6.246)	6.235 (3.8)
Mean	LUAD	5.14 (4.02)	4.957 (2.564)	6.094 (3.898)
Mean	LUSC	5.907 (3.386)	5.663 (4.528)	3.649 (1.451)
Mean	OV	5.012 (1.664)	9.299 (4.583)	10.767 (5.345)
Mean	PAAD	8.714 (5.36)	6.624 (8.892)	5.593 (2.247)
Mean	READ	1.074 (0.832)	1.93 (1.398)	1.125 (0.912)
Mean	SARC	4.763 (4.202)	4.608 (4.553)	4.057 (2.113)
Mean	SKCM	2.486 (1.373)	3.094 (2.001)	3.043 (2.041)
Mean	STAD	4.831 (3.508)	7.029 (16.062)	5.354 (3.667)
Mean	UCEC	1.136 (0.744)	1.468 (1.302)	1.057 (0.879)

J Supplementary figures

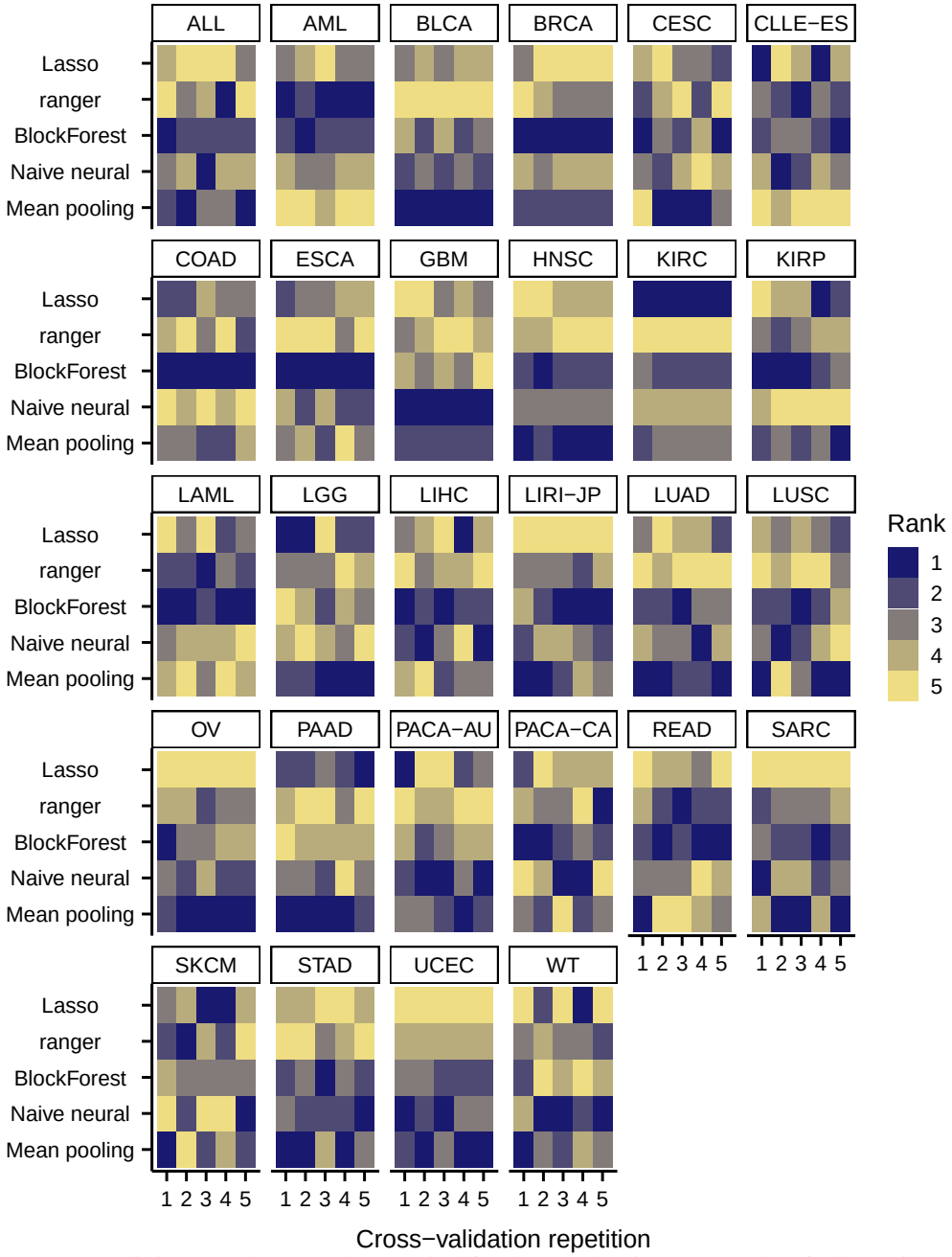

Figure S1: Antolini concordance rank (one signifies best rank, six worst rank) of all considered models across cross-validation repetitions on all cancer types.

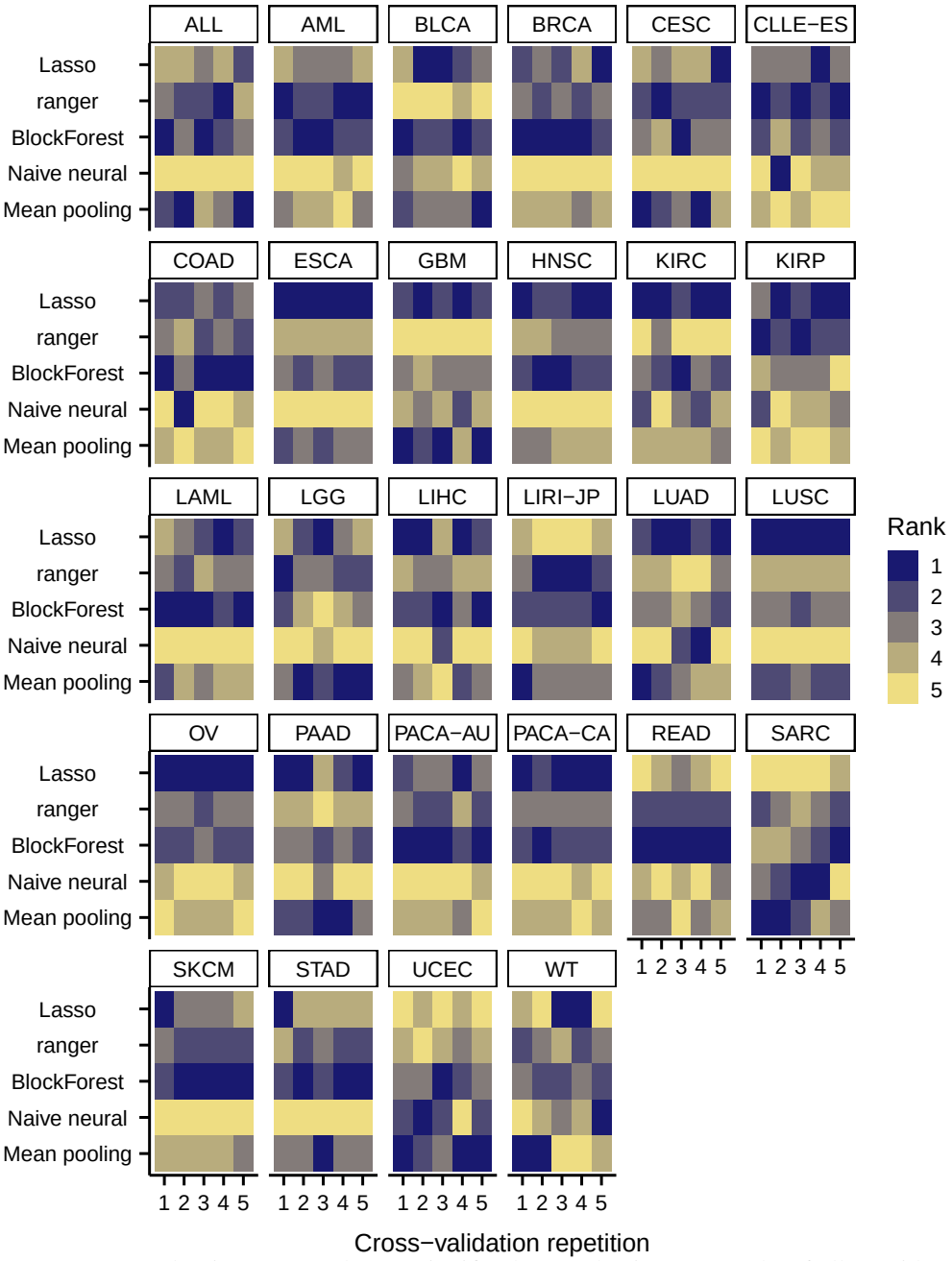

Figure S2: Integrated Brier score rank (one signifies best rank, six worst rank) of all considered models across cross-validation repetitions on all cancer types.

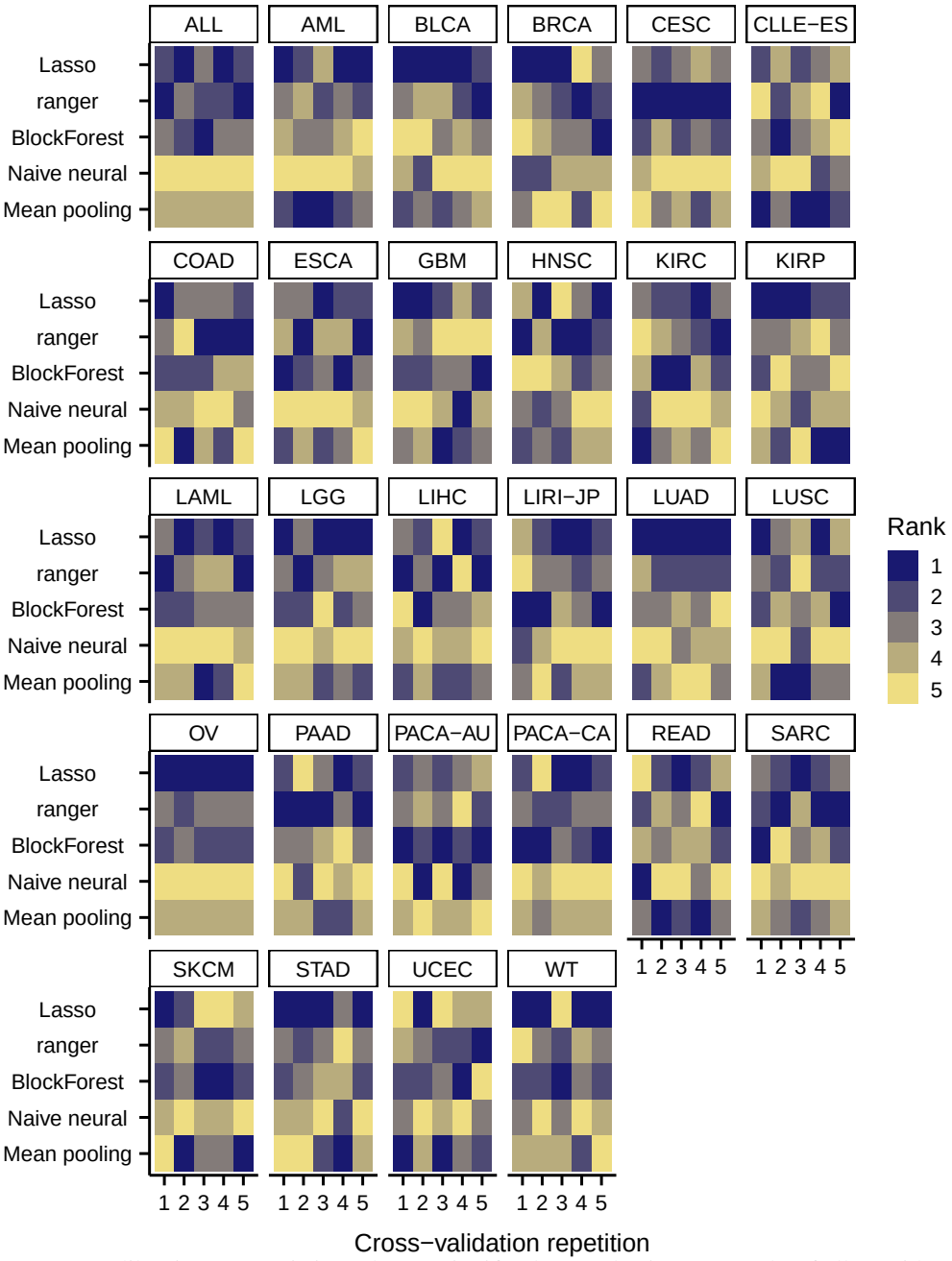

Figure S3: D-Calibration test statistic rank (one signifies best rank, six worst rank) of all considered models across cross-validation repetitions on all cancer types.

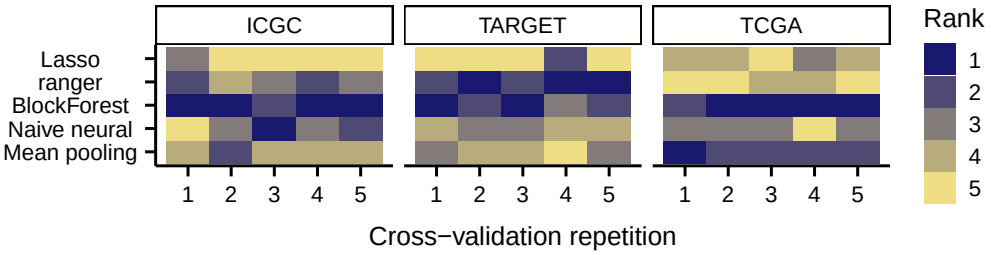

Figure S4: Antolini concordance rank (one signifies best rank, six worst rank) of all considered models across cross-validation repetitions pooled by project.

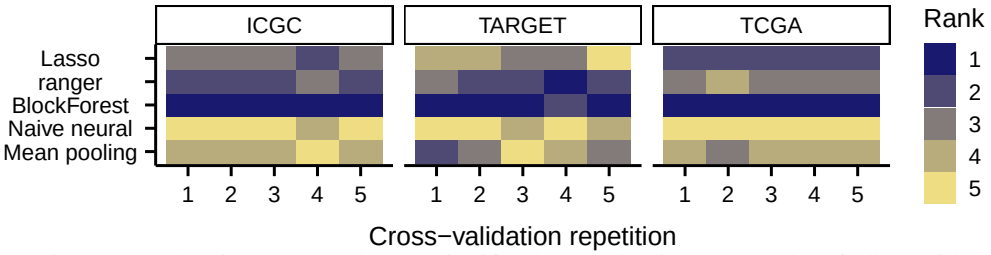

Figure S5: Integrated Brier score rank (one signifies best rank, six worst rank) of all considered models across cross-validation repetitions pooled by project.

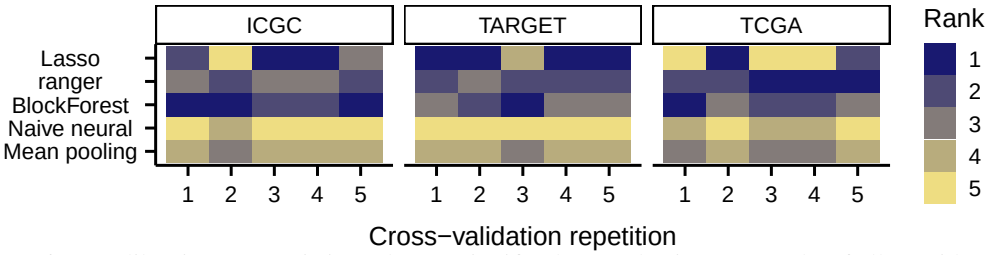

Figure S6: D-Calibration test statistic rank (one signifies best rank, six worst rank) of all considered models across cross-validation repetitions pooled by project.