Supervised Hierarchical Autoencoders for Multi-Omics Integration in Cancer Survival Models

David Wissel[1, 2], **Daniel Rowson**[2], and **Valentina Boeva**[2, 3, 4, *]

[1]Department of Mathematics, Seminar for Statistics, ETH Zurich, 8092 Zurich, Switzerland
[2]Department of Computer Science, Institute for Machine Learning, ETH Zurich, 8092 Zurich, Switzerland
[3]Swiss Institute for Bioinformatics (SIB), Zurich, Switzerland
[4]Institut Cochin, Inserm U1016, CNRS UMR 8104, Paris Descartes University UMR-S1016, 75014 Paris, France
[*]To whom correspondence should be addressed: valentina.boeva@inf.ethz.ch

Abstract

With the increasing amount of high-throughput sequencing data becoming available, the proper integration of differently sized and heterogeneous molecular and clinical groups of variables has become crucial in cancer survival models. Due to the difficulty of multi-omics integration, the Cox Proportional-Hazards (Cox PH) model using clinical data has remained one of the best-performing methods [Herrmann et al., 2021]. This motivates the need for new models which can successfully perform multi-omics integration in survival models and outperform the Cox PH model. Furthermore, there is a strong need to make multi-omics models more sparse and interpretable to encourage their usage in clinical settings. We developed a novel neural architecture, termed Supervised Hierarchical Autoencoder (*SHAE*), based on supervised autoencoders and Sparse-Group-Lasso regularization. Our new method performed competitively with the best performing statistical models used for multi-omics survival analysis. Moreover, it outperformed the Cox PH model using clinical data across all 17 cancers from The Cancer Genome Atlas (TCGA) considered in our work. We further showed that surrogate linear models for *SHAE* trained on a subset of multi-omics groups achieved competitive performance at consistently high sparsity levels, enabling usage within clinics. Alternatively, surrogate models can act as a feature selection step, permitting improved performance in arbitrary downstream survival models. Code for the reproduction of our results is available on Github.

1 Introduction

Accurate prediction of survival times is essential for clinicians and researchers to decide treatment and identify which variables drive survival. The Cox Proportional-Hazards (Cox PH) model [Cox, 1972, Breslow, 1975] is still the *de facto* standard model for survival analysis today, despite proposals for various other methods such as random survival forests [Ishwaran et al., 2008], boosting [Hothorn et al., 2010] and neural networks [Ching et al., 2018].

Survival analysis of cancer patients can be particularly challenging due to the heterogeneous nature of the disease, even for patients suffering from the same type of cancer [Polyak et al., 2011, Fisher et al., 2013, Melo et al., 2013]. With the advent of high throughput sequencing technologies, researchers hoped to leverage the information inherent in biological data such as gene expression, DNA methylation, and others (jointly referred to as multi-omics) to help explain and mitigate this heterogeneity.

However, even using the wealth of newly available biological data in large scale projects such as The Cancer Genome Atlas Program (TCGA) [Tomczak et al., 2015], significant improvements in performance in cancer survival analysis as measured by performance metrics such as Harrell's concordance [Harrell et al., 1982] or the brier score [Brier et al., 1950] have been elusive.[1] Herrmann et al. [2021] showed that the Cox PH model using clinical data outperformed most other methods, even when these were designed to integrate multi-omics data.

1.1 Multi-omics survival models

There have been various proposals for statistical and neural network-based models that perform multi-omics integration in the context of (cancer) survival analysis. Hornung and Wright [2019] proposed five variations of the random forest algorithm [Breiman, 2001], all of which change the split point selection by taking into account that the input variables belong to different groups (e.g., multi-omics data). *BlockForest*, their best performing method, statistically significantly outperformed random survival forest in their work [Hornung and Wright, 2019] and was shown to outperform the clinical Cox PH model on TCGA by Herrmann et al. [2021]. Boulesteix et al. [2017] proposed a modified Lasso regularized Cox PH model that scales the Lasso penalty λ with a group-specific penalty factor that can be chosen through *a priori* knowledge or using cross-validation. The authors showed that their new model termed *ipfLasso* performed better than the standard Lasso regularized Cox PH model in simulations and on TCGA. Klau et al. [2018] introduced a sequential Lasso regularized Cox PH approach based on offsetting, *priorityLasso*, which considers input groups one at a time in order and uses the previous model prediction as an unregularized offset for the next model. *PriorityLasso* outperformed Lasso regularized Cox PH and offers clinicians the advantage of deciding which variables to preferentially include in the model [Klau et al., 2018].

Cheerla and Gevaert [2019] proposed a novel neural network architecture that integrates gene expression, miRNA, clinical data, and whole slide images to predict cancer survival on 20 TCGA cancers. Their model benefitted (that is, exhibited an increased concordance index) from pan-cancer training relative to training solely on each cancer type for most considered cancers. Kim et al. [2020] proposed an architecture based on a variational autoencoder (VAE) [Kingma and Welling, 2013] on which they applied transfer-learning. They trained their VAE on 20 TCGA cancers and transferred the weights of the first two layers to their survival model, which they then fine-tuned on each of the same ten TCGA cancers on which *cox-nnet* was benchmarked [Ching et al., 2018]. Their model outperformed both *cox-nnet* and regularized Cox PH on seven out of ten cancers when trained on gene expression data only. Huang et al. [2019] modeled TCGA breast cancer survival by integrating miRNA, mRNA, copy number variation, and mutation data. They proposed an architecture that takes the mRNA-seq eigengene matrix and the miRNA-seq eigengene matrix; both passed through a hidden layer individually (*i.e.,* miRNA does not interact with mRNA). Afterward, a final layer predicts the relative risk from the output of the hidden layer of miRNA, the hidden layer of mRNA, copy number variation, mutation, and selected clinical variables. Their new model outperformed random survival forest, regularized Cox PH, and deepsurv [Katzman et al., 2018].

1.2 Sparse group Lasso regularized models

An alternative architecture-agnostic method for incorporating knowledge about input feature groups is (Sparse-)Group-Lasso regularization. Sparse-group Lasso [Friedman et al., 2010a, Simon et al., 2013] (*SGL*) regularization is a convex combination of the Lasso [Tibshirani, 1996] and the group-Lasso. Initially introduced by Friedman et al. [2010a], *SGL* regularization promotes sparsity both within groups (through the Lasso) and between groups (through the Group-Lasso).

Lemsara et al. [2020] proposed a multi-modal autoencoder, deemed PathME, which leverages *SGL* between different multi-omics feature groups to perform, in combination with sparse non-negative matrix factorization, clustering of TCGA patients. Xie et al. [2019] used Group-Lasso regularization to propose an architecture similar to Ching et al. [2018] but designed for multi-omics integration. The authors showed that group-Lasso regularization prevented overfitting (relative to Lasso and using no regularization) and statistically significantly outperformed the same architecture with Lasso regularization on three out of 14 considered TCGA cancers.

[1] From here on out, we will use concordance index and Harrell's concordance interchangeably.

1.3 Autoencoders

Autoencoders have also been widely used for multi-omics integration and cancer survival analysis more broadly. Tong et al. [2020] explored multi-modal autoencoders for the integration of multi-omics data in breast cancer survival. They proposed two architectures, each of which has a dedicated autoencoder per modality. The first, termed *CrossAE*, tries to reconstruct both its own input and all other input groups. Afterward, *CrossAE* mean-pools the latent representations and uses the result to predict the relative risk. In their second architecture, *ConcatAE*, each autoencoder only reconstructs its input. Afterward, the model concatenates all latent representations and uses them to predict the relative risk. When comparing different combined multi-omics groups and their different architectures, *ConcatAE* performed the best on TCGA breast cancer when integrating methylation and miRNA and using PCA for dimensionality reduction.

Instead of using autoencoders solely for dimension reduction, their reconstruction loss can also act as a regularizer to be trained jointly with a supervised loss. Le et al. [2018] showed that these models have good theoretical properties and that training such an autoencoder (deemed a supervised autoencoder) performed as well or better than neural networks that optimized only a supervised loss on the respective supervised task. Thus, the reconstruction loss can be treated as a form of regularization, the level of which is a hyper-parameter. Le et al. [2018] found that when $L_{\text{total}} = L_{\text{reconstruction}} + \alpha L_{\text{supervised}}$, levels of α around 0.01 were optimal on their datasets. Tan et al. [2020] presented an application of supervised autoencoders on TCGA data. The authors tried to predict binarized clinical endpoints (*e.g.,* overall survival, disease-free survival) from TCGA data using methylation, miRNA, mRNA, and reversed-phase protein arrays (RPPA). They trained one supervised autoencoder for each omics group and fused their latent spaces using mean-pooling, from which they then predicted the endpoints. Using this architecture, they outperformed other machine learning models such as random forests and SVMs.

Our work explored a neural network-based approach to multi-omics integration, which leverages hierarchical supervised autoencoders [Le et al., 2018] combined with sparse group Lasso regularization [Simon et al., 2013] to achieve competitive performance on TCGA. In particular, we showed that neural models could perform as well or better than *BlockForest* and its variants, even without transfer learning. Furthermore, we studied how training surrogate models on a subset of all multi-omics groups could help transfer multi-omics models to clinical settings by achieving close to state-of-the-art results at very high sparsity. In addition, we partially reframed multi-omics integration as feature selection, showing that fitting arbitrary survival models on a suitable feature set partially closes the performance gap to integrative models.

2 Methods

2.1 Architecture

We developed a novel method, termed *SHAE* (**S**upervised **H**ierarchical **A**uto**e**ncoder) for multi-omics integration in cancer survival models, which builds on hierarchical multi-modal autoencoders and Sparse-Group-Lasso regularization (Figure 1). Our models have two levels of autoencoders. On the first level, each input group $m \in M$ has a separate autoencoder that compresses the features of each group ($X^{(m_i)}$) into a latent space $\tilde{m}_i$. Since we modeled survival using supervised autoencoders, we also introduced linear layers from each latent space to a predicted relative risk $\hat{\phi}_i(\tilde{m}_i)$. On the second level, another autoencoder takes the concatenation of all latent spaces $C = [\tilde{m}_1, ..., \tilde{m}_{|M|}]$ from all first-level autoencoders and compresses them into another latent space $\tilde{C}$ used for the final prediction of the relative risk for each patient $\hat{\varphi}(\tilde{C})$. The hierarchical aspect of our model was partially inspired by Simidjievski et al. [2019], who provided an overview of different possible architectures of VAEs for multi-omics integration.

We also explored a residual variant of *SHAE* (Figure 1). We fed the first modality (m_1, clinical data in our case) into our model and skipped it to the final layer in this variant. Using this skipping, we hoped to achieve two goals:

1. The model could still take clinical data into account within the second-level autoencoder, which might help other features, both with the reconstruction and for learning a better latent space for survival prediction.

2. This may aid performance since clinical data (in most datasets) contains valuable survival information.

This residual idea was heavily inspired by the favoring approach of *BlockForest* [Hornung and Wright, 2019] and *resnets* [He et al., 2016]. We used the mean squared error (MSE) as our reconstruction loss for all autoencoders and the Breslow approximation of the negative partial log-likelihood as the loss for all predicted relative risk values. Let A denote the set of all linear layers in our networks and let B be the linear regularized using *SGL* (blue edges in Figure 1). We regularized all linear layers $a \in A \backslash B$ with L2 regularization and layer B with *SGL*. Let δ_i be the event indicator for patient i where $\delta_i = 1$ if patient i experienced the event during the study and $\delta_i = 0$ otherwise. Further, supposing that X_i is the time of the event, we let $T_i = \min(X_i, S_i)$ where S_i is the time at which patient i was censored. ξ and γ are hyper-parameters controlling the strength of the reconstruction regularization [2]. λ_1, λ_2 are hyper-parameters controlling the strength of the L2 and SGL regularization respectively and α is a trade-off hyper-parameter between the Lasso ($\alpha = 1$) and the Group Lasso ($\alpha = 0$). $X^{(m_1)}$ denotes the first input modality and $\hat{X}^{(m_1)}$ denotes a reconstruction of the first input modality. Then, the full loss (1) for *SHAE* becomes:

$$
\begin{aligned}
&L_{\text{total}}\left(\hat{\varphi}, \hat{\phi}_1, ..., \hat{\phi}_{|M|}, X, \hat{X}, C, \hat{C}\right) \\
&= L_{\text{cox}}(U, \delta, \hat{\varphi}) + \xi \cdot \text{MSE}(C, \hat{C}) \\
&+ \sum_{q=1}^{|M|} \left(L_{\text{cox}}(T, \delta, \hat{\phi}_q) + \gamma \cdot \text{MSE}\left(X^{(m_q)}, \hat{X}^{(m_q)}\right)\right) \\
&+ \lambda_1 \sum_{a \in A \backslash B} ||a||_F^2 + \lambda_2(1-\alpha) \sum_{m \in \{m_1, ..., m_{|M|}\}} \sqrt{|m|}||B^{(m)}||_F \\
&+ \lambda_2 \alpha ||B||_{1,1}
\end{aligned}
\tag{1}
$$

Where $||.||_F$ denotes the Frobenius norm, L_{cox} is the Breslow approximation of the negative partial log-likelihood, and $||.||_{1,1}$ denotes the entrywise matrix L1 norm. MSE denotes the mean-squared error, and $B^{(m)}$ is a submatrix of B with only those columns corresponding to inputs of modality m. $U_i = \min(T_i, X_i)$, is the observed time until a patient was either right-censored C_i or experienced the event T_i (death, in this setting).

We developed all neural nets using Pytorch [Paszke et al., 2019] and skorch [Tietz et al., 2017]. We defaulted to scaling the supervised loss by multiplying it by the batch size (*i.e.,* $\xi = \gamma =$ batch size) since we found this worked well initially and saved one hyper-parameter. For both *SHAE* and *SHAE residual*, we performed standardization of all features before fitting the model. Further, we used batch normalization [Ioffe and Szegedy, 2015] and Parametric Rectified Linear Units for non-linear activations [He et al., 2015]. We used the *Adam* optimizer [Kingma and Ba, 2014] for training our models and trained for 25 epochs, with a batch size equal to the size of the dataset (*i.e.,* no batching) and an initial learning rate of 0.01 (the highest learning rate before the training loss started diverging for most cancers). We tuned the regularization parameters $\lambda_1, \lambda_2 \in \{1e-2, 1e-3, 1e-4\}$ using cross-validation. Since our models employ batch normalization, research suggests that it may be possible to fix the learning rate given the right level of weight decay due to their strong interdependence in the presence of batch normalization, even in adaptive methods like *Adam* [Hoffer et al., 2018, Van Laarhoven, 2017]. Thus, we tuned only the regularization parameters and not the learning rate. All latent space sizes were set to 64, and all encoders (and, by symmetry, decoders) were set to have one hidden layer with 128 nodes. While all of these architectural hyper-parameters were trainable in our implementation, we did not tune these in order not to complicate the hyper-parameter search space.

We included both *SHAE* and *SHAE residual* in our benchmarks, where the clinical input variable group was skipped for *SHAE residual*.

[2] Instead of down-weighting the supervised loss, as Le et al. [2018] did, we upweighted the reconstruction loss.

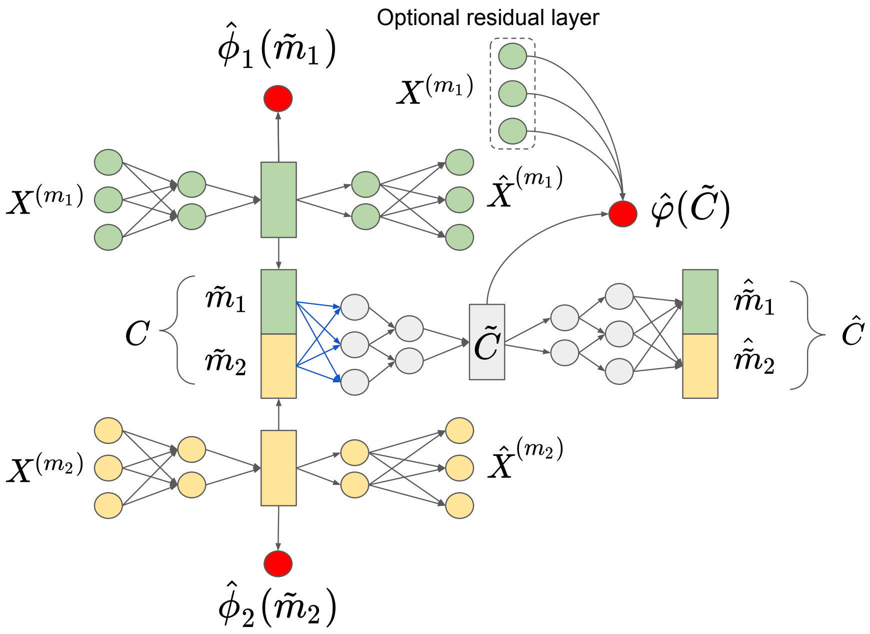

Figure 1: Architecture diagram of *SHAE*. Grey edges denote L2 regularization, blue edges SGL regularization. Let $X^{(m_i)}$ denote the part of X which corresponds to the i-th input variable group, $\hat{X}^{(m_i)}$ a reconstruction of the same. $\tilde{m}_i$ is the latent space of an autoencoder taking $X^{(m_i)}$ as its input and $C = [\tilde{m}_1, ..., \tilde{m}_{|M|}]$ is a concatenation of all first level latent spaces. $\hat{\phi}_1, ..., \hat{\phi}_m$ as well as $\hat{\varphi}$ are predicted log-partial hazards based on their respective inputs. $\tilde{C}$ is the latent space of the second-level autoencoder that takes C as its input.

2.2 Reference models

For reference models, we used the best performing model from the benchmark study of Herrmann et al. [2021], *BlockForest* (*BF*), as well as *RandomBlock favoring* (*RBF*)[3]. Favoring refers to the model always considering a particular block of variables in the split-point selection (most often clinical variables). *RBF* and *BF* were implemented using the *blockForest* package. For *RBF*, we favored clinical variables.

We also included a random survival forest method (*RSF*) and a Lasso Regularized Cox PH (*Lasso*) as two benchmark methods that did not use the group structure information present in the multi-omics variables. Lastly, we included a Ridge regularized Cox PH using only the clinical variables (*Clin. Cox PH*). We opted for a Ridge regularized model since this allowed us to prevent convergence issues often seen with one-hot encoded categorical clinical variables. *RSF* was implemented using *ranger* [Wright and Ziegler, 2017] while *Lasso* and *Cox PH* were both implemented using *glmnet* [Friedman et al., 2010b, Simon et al., 2011]. *Lasso* and *Cox PH* were set to standardize their input matrices, while no further preprocessing was performed for *RSF*, *BF* and *RBF*.

2.3 Datasets

We benchmarked all models on the TCGA dataset. We followed the approach of Herrmann et al. [2021] in selecting cancers with at least 100 samples (after preprocessing in our case) and an average event ratio $\geq 5\%$ to ensure there were enough patients and enough events to calculate meaningful concordance values. Further, we used *GISTIC 2.0* for copy number variation (CNV) data [Mermel et al., 2011] and the number of non-silent *MC3* mutation calls per gene per patient [Ellrott et al., 2018], the former of which is taken from *Xenabrowser* [Goldman et al., 2020], with mutation coming

[3] *RandomBlock* is an alternative version of *BlockForest* - Hornung and Wright [2019] showed that it outperformed *BF* when clinical variables were favored with multi-omics data on TCGA.

Table 1: Summary information of all 17 considered TCGA datasets used in our study. TCGA cancer abbreviations used for space, please refer to https://gdc.cancer.gov/resources-tcga-users/tcga-code-tables/tcga-study-abbreviations for a full overview. Tabel format adapted from [Herrmann et al., 2021].

Cancer	n	p	Event ratio	clinical	mRNA	CNV	methylation	miRNA	mutation	RPPA
BLCA	325	84380	0.44	9	20225	24776	22124	740	16317	189
BRCA	765	80668	0.13	9	20227	24776	19371	737	15358	190
COAD	284	82221	0.22	16	17507	24776	21424	740	17569	189
ESCA	118	75752	0.36	17	19076	24776	21941	737	9012	193
HNSC	201	79286	0.60	16	20169	24776	21647	735	11752	191
KIRC	309	74652	0.28	14	20230	24776	19456	735	9252	189
KIRP	199	76294	0.15	5	20178	24776	21921	738	8486	190
LGG	395	78254	0.22	15	20209	24776	21564	740	10760	190
LUAD	338	82999	0.40	11	20165	24776	21059	739	16060	189
PAAD	100	76654	0.58	26	19932	24776	21586	732	9412	190
SARC	190	76068	0.36	45	20206	24776	21724	739	8385	193
SKCM	238	85254	0.52	3	20179	24776	21635	741	17731	189
STAD	304	80860	0.42	7	16765	24776	21506	743	16870	193
UCEC	392	84130	0.16	24	17507	24776	21692	743	19199	189
OV	161	72763	0.58	17	19064	24776	19639	731	8347	189
LIHC	157	76247	0.51	3	20078	24776	21739	742	8719	190
LUSC	280	82125	0.41	20	20232	24776	20659	739	15510	189

from *PANCANATLAS* [Chang et al., 2013].[4] In addition, we considered miRNA, mRNA, DNA methylation, RPPA, and clinical data, all of which are also taken directly from the PANCANATLAS. We log-transformed both mRNA and miRNA expression. Otherwise, no further preprocessing of the datasets was performed.

For comparability, we used the same clinical variables as Herrmann et al. [2021], with the caveat that we dropped clinical variables missing for more than five patients. Categorical clinical variables were one-hot encoded. We excluded molecular variables if they were missing for more than one patient to preserve as many patients as possible. Table 1 shows an overview of all TCGA datasets which were used in our study.

2.4 Performance metric

Similar to other works [Cheerla and Gevaert, 2019, Kim et al., 2020, Tong et al., 2020], we used Harrell's concordance (2) [Harrell et al., 1982] to measure the performance of our models, where $\hat{\phi}_i$ is an estimated score for patient i (where higher estimated score implies higher estimated risk), $U_i = \min(T_i, C_i)$ is the time until a patient was either censored (C_i) or experienced the event (T_i). We have $\delta = 0$ for censored patients and $\delta = 1$ otherwise. Equivalently, Harrell's concordance is the ratio of concordant pairs and all comparable pairs.

$$\text{Concordance}(U, \delta, \hat{\phi}) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \mathbb{1}(T_i > T_j)\mathbb{1}(\hat{\phi}_i < \hat{\phi}_j)\delta_j}{\sum_{i=1}^{n} \sum_{j=1}^{n} \mathbb{1}(T_i > T_j)\delta_j} \tag{2}$$

2.5 Validation

We tuned all models using nested cross-validation with five inner folds or out-of-bag error (for random forest-based methods) to choose the best parameters to refit on the outer fold. We used the *glmnet* internal *cv.glmnet* function to optimize the regularization parameter for *Cox PH* and *Lasso*. *BF* and *RBF* were tuned using the *blockForest::blockfor* function. For *RSF*, we tuned the *mtry* parameter using the *tuneRanger::tuneMtryFast* function. For *SHAE* and *SHAE residual*, we tuned only the regularization parameters $\lambda_1, \lambda_2 \in \{1e-2, 1e-3, 1e-4\}$, while setting the other parameters to the defaults detailed in Section 2.1.

[4] Mutation calls were calculated from the *PANCANATLAS MAF* file using *Maftools* [Mayakonda et al., 2018].

For all cancers in our dataset, we performed outer five-fold cross-validation, twice repeated, giving us a total of ten outer splits per cancer. For statistical significance testing, we tested for an overall difference between models by adopting the same approach as Hornung and Wright [2019]. We calculated the mean concordance per model per cancer (17 in total) and treated these mean values as independent between datasets. We then ran a one-sided paired t-test with a null hypothesis of non-inferiority of each benchmark model relative to *SHAE* and *SHAE residual*. The alternative hypothesis was that our model performed better than the respective benchmark method. We thus performed a total of ten tests (*SHAE* and *SHAE residual* compared to each of the five reference models), for which we corrected using Bonferroni-Holm [Holm, 1979]. We report both the raw p-values and p-values after correction.

2.6 Surrogate models

For enhanced clinical applicability, we chose to fit global surrogate Lasso models on the predictions of *SHAE* and *SHAE residual*. We fitted the models using *python-glmnet* [Friedman et al., 2010b] on clinical and gene expression data only and predicted the outputs of the full multi-omics *SHAE* and *SHAE residual* models. We did not penalize the clinical variables, as they are generally known to contain much prognostic information. We then compared the performance of these surrogate models to the reference models fitted on only clinical and gene expression data to investigate whether the surrogate model learned multi-omics specific information (which might show up in terms of increased test concordance) even though it did not have access to them in training directly.

Our approach in predicting the relative risk estimated by *SHAE (residual)* is somewhat reminiscent of pseudo-value methods in survival analysis, which replace right-censored survival data with *jackknife* pseudo-observations, thus enabling a change in model class to perform regression instead of survival analysis [Zhao and Feng, 2020, Klein and Andersen, 2005]. However, note that our approach does not have any statistical guarantees (as far as we know) that some of the other pseudo-value techniques do. We also emphasize that care must be taken when interpreting the pseudo hazard ratios obtainable by analyzing the coefficients of the surrogate models. In effect, these cannot be directly interpreted as hazard ratios but rather as a prediction of the hazard ratio which would have been predicted by *SHAE (residual)*.

Alternatively, we can also frame the surrogate models as a feature selection technique, discovering the main features learned to be essential by our multi-omics methods. We can then refit a survival model such as an *RSF*, or a Ridge regularized Cox PH model on the feature set selected by the surrogate models. While this can also be powerful, we note that it does not allow for a transfer of multi-omics information (as the downstream survival model must be fit on the original survival data, thus not enabling a transfer from the *SHAE (residual)* predictions).

3 Results

3.1 Performance on TCGA

Overall, both *SHAE* and *SHAE residual clinical* (*residual SHAE*) statistically significantly outperformed both *RSF* and *Lasso* after multiple testing correction (Table 2). The comparison to the other baseline models was more subtle, as none of the other p-values was statistically significant after correction. Still, both of our proposed models outperformed *BF* and *Cox PH* in terms of overall concordance (Figure 2A) and in terms of rank across datasets (Figure 2B). There were no strong differences between the concordance of our models and that of *RBF* as all of them perform roughly equally (Figure 2A). *RBF* had the lowest (that is, best) rank across datasets however (Figure 2B).

For computation times, both *SHAE* and *residual SHAE* ran much faster than *RBF* and *BF* (Figure S1), with both *SHAE* models being around a factor of four times faster than *RBF* across datasets.[5] Thus, *SHAE* achieved comparable performance to *RBF* in considerably faster computation times. Of course, *RSF*, *Lasso* and *Cox PH* all ran much faster than any of the multi-omics models, although this came at the price of decreased concordance.

[5] Timing benchmarks were performed on the Euler cluster of ETH Zurich using eight CPU cores with 4096 MHz, using *tictoc* [Izrailev, 2021] in *R* and *timeit.default_timer* in *Python*. Note that we did not use a GPU for the benchmarks.

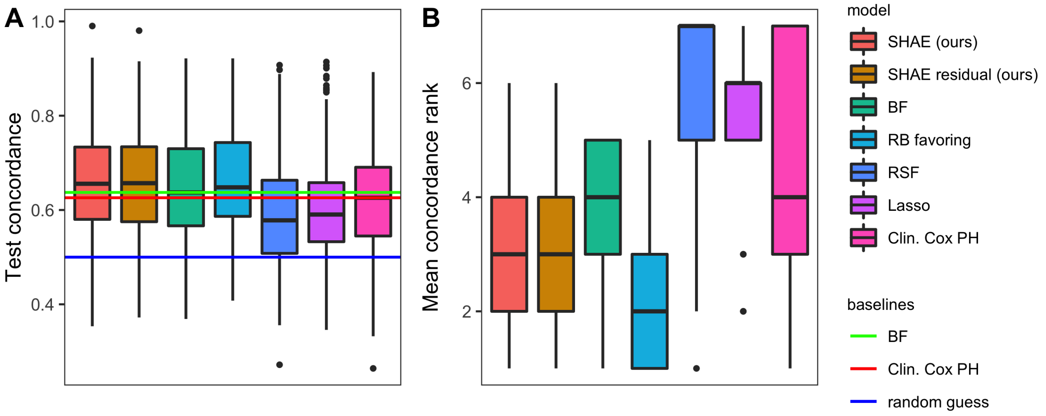

Figure 2: Performance of all models on TCGA. A: Overall test concordance across across the 17 TCGA cancers. B. Mean test concordance rank per cancer across the 17 TCGA cancers (lower is better).

Table 2: P-values of testing non-inferiority of each reference method (rounded to four digits).

model	comparison model	p-value (before correction)	p-value (after correction)
SHAE (ours)	BF	0.0978	0.3912
SHAE (ours)	Cox PH	0.0265	0.1456
SHAE (ours)	Lasso	0.0007	0.0048
SHAE (ours)	RB favoring	0.6403	1.0000
SHAE (ours)	RSF	0.0002	0.0023
SHAE residual (ours)	BF	0.1089	0.3912
SHAE residual (ours)	Cox PH	0.0243	0.1456
SHAE residual (ours)	Lasso	0.0006	0.0048
SHAE residual (ours)	RB favoring	0.6725	1.0000
SHAE residual (ours)	RSF	0.0003	0.0024

3.2 Surrogate model performance

The performance of our Lasso surrogate models of *SHAE (residual)* exceeded that of any other model trained on clinical and gene expression (Figure 3A) except *BF*. All multi-omics models and surrogate models beat *Lasso*, *RSF* and *Cox PH* when trained on clinical and gene expression only (Figure 3A). We note that the good performance of the surrogate models was not merely due to the favoring of clinical variables, as a Lasso model fitted on clinical and gene expression, which did not penalize clinical variables underperformed both of our surrogates (Figure S2).

While the training R^2 of our surrogate models was generally quite good (Figure S3), the test R^2 was much more mixed (Figure S4). This suggests that while the models effectively reproduced the predictions of *SHAE* on the training set, they struggled to do so on the test set for certain cancers. Most cancers (for example, *ESCA* and *OV*) where the models achieved a median test R^2 of below 0.5 had a sample size of below 200 suggests that this issue might be at least partially due to the low sample size. We saw overall high test R^2 values for cancers such as bladder or breast, both of which contained over 300 patients (Figure S4). Interestingly, low test R^2 of the surrogates did not necessarily correspond to lower performance relative to the full multi-omics *SHAE (residual)* models (Figure S5).

Despite their competitive performance, our surrogate models were highly sparse, with a median of 35 and 37 variables selected across all cancers for the *SHAE* surrogate and the *SHAE residual* surrogate, respectively (Figure 3B). The sparsity structure of the surrogate models by cancer and splits revealed that the models were relatively stable in the number of features selected across splits (Figure 4 and Figure S6). In addition, the surrogate models were able not to use any gene expression features when it presumably could have hurt performance (*e.g.,* on *ESCA*; Figure 4). The Lasso model trained

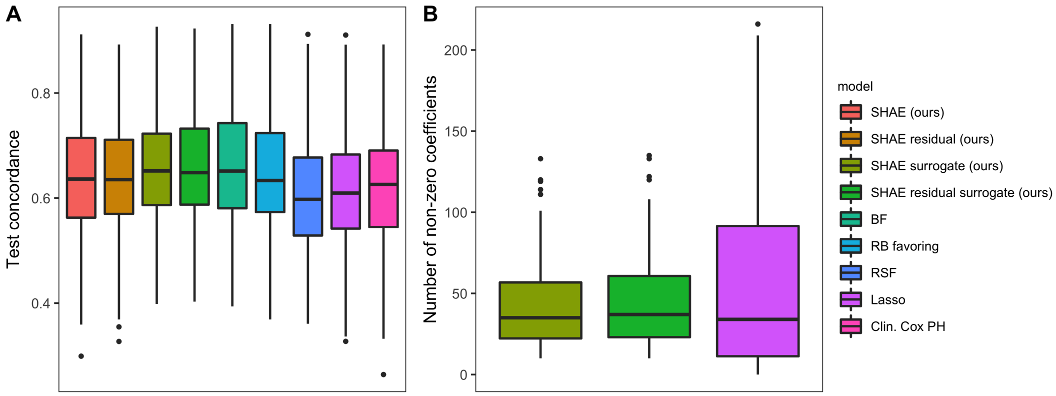

Figure 3: Performance of the Lasso surrogate models for multi-omics *SHAE (residual)*. A: Test concordance of all models trained on clinical and gene expression only. Cox PH trained on clinical only, surrogate models trained on the predictions of full multi-omics *SHAE (residual)*. B: Number of non-zero coefficients of the surrogate models and Lasso.

on clinical and gene expression data tended to have major differences in the number of non-zero coefficients across splits (Figure S7).

When leveraging surrogate models purely for feature selection, survival models trained on the same feature set as one of the surrogate models achieved improved performance relative to fitting them on all clinical and gene expression features (Figure S8). However, the models refit with the feature set recovered by the *SHAE residual* surrogate could not achieve the same performance as the surrogates (Figure S9).

4 Discussion

First and foremost, we showed that *SHAE* could be leveraged to achieve competitive performance on the TCGA dataset as measured by the concordance index. Our benchmark is especially interesting since there has not been a conclusive comparison of neural models for multi-omics integration to state-of-the-art statistical methods such as *BF*. Most other work either focused on statistical methods only [Herrmann et al., 2021] or used simple statistical methods such as the Lasso [Huang et al., 2019].

While our work was not a neutral benchmark study and thus likely contains some bias, we believe it validates *SHAE* as a solid alternative to *BF* and its variants. There were certain cancers such as Esophageal carcinoma (ESCA) where *SHAE* outperformed *RBF* while on other cancers (*e.g.,* Brain Lower Grade Glioma (LGG)), *RBF* dominated *SHAE* (Figure S10). Ensemble methods or a mixture of experts approach could leverage this diversity of performance across cancers to produce even better results. Neural networks such as *SHAE* were also shown to provide additional benefits relative to *RBF*, such as faster computation times (Figure S2). In addition, *SHAE* performed approximately as well as *RBF* overall, even without having to be told that clinical information was important. This suggests that *SHAE* might be a good alternative, especially when researchers are unsure whether clinical data contains considerable prognostic information.

More broadly, our results were consistent with those of Herrmann et al. [2021]. We saw *BF* outperform *Clin. Cox PH* by a small margin. At the same time, both *RSF* and *Lasso* failed to beat the clinical-only model. One difference in our work is that we only kept clinical variables if no more than five patients were missing them. Since our objective was to compare *SHAE* to other multi-omics methods such as *BF*, this limitation seemed acceptable since all multi-omics models should be affected roughly uniformly by having access to fewer clinical variables.

We found that the difference in performance between *SHAE* and *SHAE residual* was much smaller than that between *BF* and *RBF*. Across all cancers, the two *SHAE* models had practically the same performance in terms of mean and median concordance (Figure 2A). Future work is thus needed to

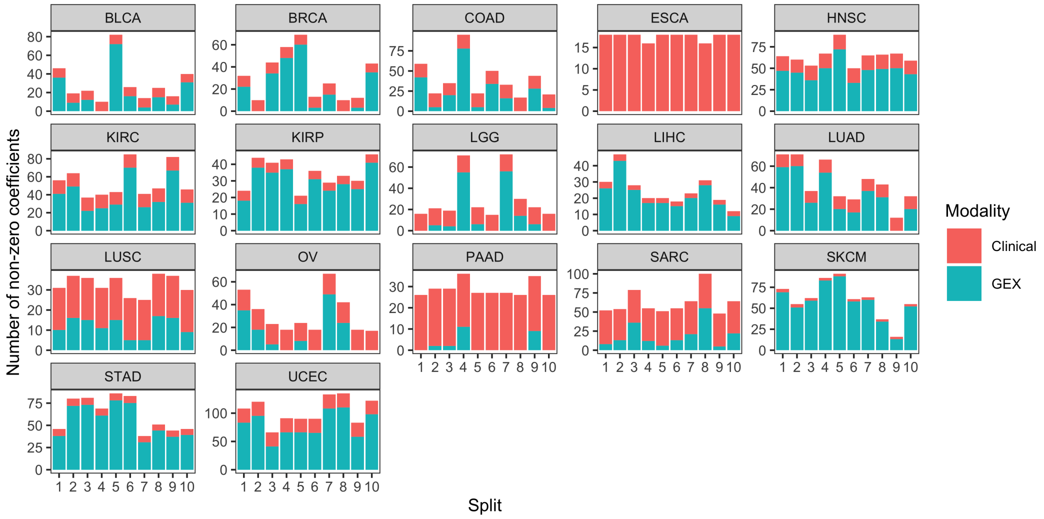

Figure 4: Sparsity structure of the Lasso surrogate model for multi-omics *SHAE residual* trained only on clinical and gene expression data.

establish the effect of and ideal way (if any) of forcing autoencoders to include a particular variable group, potentially yielding a similar performance boost to that seen in *BF* based models.

We also showed that Lasso surrogate models for *SHAE (residual)* could maintain good surrogate performance (in terms of train R^2) even when trained on only a subset of available modalities (specifically clinical and gene expression) (Figure S3). Although the test R^2 was more nuanced and very cancer-specific (Figure S4), we nevertheless validated that training surrogate models of full multi-omics models can outperform most multi-omics models trained from scratch on the same subset of input groups (Figure 3A). We found that survival models trained on the same feature set as the surrogates could not entirely match their performance (Figure S9). This might imply that the surrogates benefitted from some of the multi-omics information present in the *SHAE (residual)* predictions.

The only model that our best surrogate model could not outperform was *BF*. *BF* performed better when trained on clinical and gene expression only as opposed to all multi-omics data (for clinical and gene expression only, *BF* had 0.014 higher mean and median concordance), which matches the study of Hornung and Wright [2019]. Possible reasons for this include high redundancy of information in additional added blocks beyond clinical and gene expression [Hornung and Wright, 2019]. Nevertheless, all other multi-omics models (*RBF*, *SHAE* and *SHAE residual*) performed better with multi-omics data.

The surrogate models produced highly sparse models overall (Figure 3B). Especially considering clinical variables were left unpenalized (and thus always selected), the surrogate models did not use many additional variables relative to Clin. Cox PH, yet were able to outperform it consistently (Figure 3A and Figure S9).

Furthermore, since all that needs to be communicated for the application of regression models such as our surrogates are the coefficients and intercept (if any), they are much more robust and easier to deploy relative to complicated black-box models [Klau et al., 2018, Boulesteix et al., 2017]. Of course, sufficient care must be taken when deploying surrogate models in clinics, both due to mismatches in sequencing protocol (although this is the case for all methods) and potential difficulties in analyzing surrogate coefficients. Alternatively, clinicians or researchers can refit regularized Cox PH models at the same sparsity level using the feature set recovered by the surrogates. Although this does not quite recover the same performance, it nevertheless outperformed *Clin. Cox PH* at high sparsity levels (Figure S9).

There are many promising avenues of future work connected with our approach in this study which we did not pursue further. First and foremost, it would have been interesting to disentangle better the effects of *SGL*, the supervised autoencoder, and the hierarchical autoencoder architecture on model performance. This may yield additional insights as to which architectures should be preferentially explored in future work. Another avenue concerns the favoring of the clinical variables: We examined the residual approach, in which we both fed clinical data into our autoencoder and skipped it to the end, where it was fed directly into the linear layer connected to the final prediction. One straightforward extension would explore differentially regularizing the latent space from the autoencoder and the skipped clinical data, *e.g.,* using different penalty factors, similar to *ipfLasso* [Boulesteix et al., 2017].

Lastly, our work has some limitations. First and foremost, we chose a fixed initial learning rate and number of epochs for our models that may have caused overfitting, even though we did not choose these hyper-parameters based on test performance. Therefore, more experiments should be conducted to validate the robustness of *SHAE* to its different hyper-parameters as well as modern techniques for learning rate scheduling such as cosine annealing [Loshchilov and Hutter, 2016]. Another limitation is that we only benchmarked on TCGA; even though this is a common strategy among researchers, future work should run a comparable benchmark, including statistical methods and neural networks on another large-scale multi-omics survival dataset. In addition, our exploration of surrogate models should be taken with a grain of salt. Despite their good performance, our approach does not have any statistical guarantees that we are aware of.

5 Conclusion

Our work demonstrated that supervised hierarchical autoencoders with sparse-group Lasso regularization are effective for multi-omics integration in cancer survival models. They performed on-par with the best statistical method in a recent large-scale benchmarking study [Herrmann et al., 2021], *BlockForest*, and one of its variants, *RandomBlock favoring*. We benchmarked all methods on 17 datasets of TCGA, following the dataset selection strategy of Herrmann et al. [2021].

The main contribution of our work was the architecture of our model, which to our knowledge, was the first to apply supervised hierarchical autoencoders to multi-omics integration in cancer survival. We further showed that by training global surrogate models using only clinical and gene expression, we could outperform all models except *BF* trained on clinical and gene expression only by transferring some of the multi-omics information learned by *SHAE*. Since our surrogate models also proved to be very sparse, they can enable high performance in clinical settings while only using few variables from one additional modality (namely gene expression).

6 Acknowledgements

The results shown here are in whole or part based upon data generated by the TCGA Research Network: https://www.cancer.gov/tcga. All benchmarks were run on the *Euler* cluster of ETH Zurich.

References

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.

David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

Norman E Breslow. Analysis of survival data under the proportional hazards model. *International Statistical Review/Revue Internationale de Statistique*, pages 45–57, 1975.

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.

Torsten Hothorn, Peter Bühlmann, Thomas Kneib, Matthias Schmid, and Benjamin Hofner. Model-based boosting 2.0. *Journal of Machine Learning Research*, 11:2109–2113, 2010.

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.

Kornelia Polyak et al. Heterogeneity in breast cancer. *The Journal of clinical investigation*, 121(10): 3786–3788, 2011.

Rosie Fisher, Lazos Pusztai, and C Swanton. Cancer heterogeneity: implications for targeted therapeutics. *British journal of cancer*, 108(3):479–485, 2013.

Felipe De Sousa E Melo, Louis Vermeulen, Evelyn Fessler, and Jan Paul Medema. Cancer heterogeneity—a multifaceted view. *EMBO reports*, 14(8):686–695, 2013.

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

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.

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

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.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Anne-Laure Boulesteix, Riccardo De Bin, Xiaoyu Jiang, and Mathias Fuchs. Ipf-lasso: integrative-penalized regression with penalty factors for prediction based on multi-omics data. *Computational and mathematical methods in medicine*, 2017, 2017.

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.

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

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.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Zhi Huang, Xiaohui Zhan, Shunian Xiang, Travis S Johnson, Bryan Helm, Christina Y Yu, Jie Zhang, Paul Salama, Maher Rizkalla, Zhi Han, et al. Salmon: survival analysis learning with multi-omics neural networks on breast cancer. *Frontiers in genetics*, 10:166, 2019.

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.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*, 2010a.

Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of computational and graphical statistics*, 22(2):231–245, 2013.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Amina Lemsara, Salima Ouadfel, and Holger Fröhlich. Pathme: pathway based multi-modal sparse autoencoders for clustering of patient-level multi-omics data. *BMC bioinformatics*, 21(1):1–20, 2020.

Gangcai Xie, Chengliang Dong, Yinfei Kong, Jiang F Zhong, Mingyao Li, and Kai Wang. Group lasso regularized deep learning for cancer prognosis from multi-omics and clinical features. *Genes*, 10(3):240, 2019.

Li Tong, Jonathan Mitchel, Kevin Chatlin, and May D Wang. Deep learning based feature-level integration of multi-omics data for breast cancer patients survival analysis. *BMC medical informatics and decision making*, 20(1):1–12, 2020.

Lei Le, Andrew Patterson, and Martha White. Supervised autoencoders: Improving generalization performance with unsupervised regularizers. *Advances in neural information processing systems*, 31:107–117, 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.

Nikola Simidjievski, Cristian Bodnar, Ifrah Tariq, Paul Scherer, Helena Andres Terre, Zohreh Shams, Mateja Jamnik, and Pietro Liò. Variational autoencoders for cancer data integration: design principles and computational practice. *Frontiers in genetics*, 10:1205, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037, 2019.

Marian Tietz, Thomas J. Fan, Daniel Nouri, Benjamin Bossan, and skorch Developers. *skorch: A scikit-learn compatible neural network library that wraps PyTorch*, jul 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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Elad Hoffer, Ron Banner, Itay Golan, and Daniel Soudry. Norm matters: efficient and accurate normalization schemes in deep networks. *arXiv preprint arXiv:1803.01814*, 2018.

Twan Van Laarhoven. L2 regularization versus batch and weight normalization. *arXiv preprint arXiv:1706.05350*, 2017.

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):1–17, 2017. doi: 10.18637/jss.v077.i01.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010b. 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, 2011.

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.

Kyle Ellrott, Matthew H Bailey, Gordon Saksena, Kyle R Covington, Cyriac Kandoth, Chip Stewart, Julian Hess, Singer Ma, Kami E Chiotti, Michael McLellan, et al. Scalable open science approach for mutation calling of tumor exomes using multiple genomic pipelines. *Cell systems*, 6(3): 271–281, 2018.

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.

K Chang, CJ Creighton, C Davis, L Donehower, et al. The cancer genome atlas pan-cancer analysis project. *Nat Genet*, 45(10):1113–1120, 2013.

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.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

Lili Zhao and Dai Feng. Deep neural networks for survival analysis using pseudo values. *IEEE journal of biomedical and health informatics*, 24(11):3308–3314, 2020.

John P Klein and Per Kragh Andersen. Regression modeling of competing risks data based on pseudovalues of the cumulative incidence function. *Biometrics*, 61(1):223–229, 2005.

Sergei Izrailev. *tictoc: Functions for Timing R Scripts, as Well as Implementations of Stack and List Structures*, 2021. URL `https://CRAN.R-project.org/package=tictoc`. R package version 1.0.1.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.