BulkRNABert: Cancer prognosis from bulk RNA-seq based language models

Maxence Gélard[1,2]
m.gelard@instadeep.com

Guillaume Richard[1]
g.richard@instadeep.com

Thomas Pierrot[1]
t.pierrot@instadeep.com

Paul-Henry Cournède[2]
paul-henry.cournede@centralesupelec.fr

[1]InstaDeep

[2] Université Paris-Saclay, CentraleSupélec,
Lab of Mathematics and Computer Science (MICS)

Abstract

RNA sequencing (RNA-seq) has become a key technology in precision medicine, especially for cancer prognosis. However, the high dimensionality of such data may restrict classic statistical methods, thus raising the need to learn dense representations from them. Transformers models have exhibited capacities in providing representations for long sequences and thus are well suited for transcriptomics data. In this paper, we develop a pre-trained transformer-based language model through self-supervised learning using bulk RNA-seq from both non-cancer and cancer tissues, following BERT's masking method. By probing learned embeddings from the model or using parameter-efficient fine-tuning, we then build downstream models for cancer type classification and survival time prediction. Leveraging the TCGA dataset, we demonstrate the performance of our method, *BulkRNABert*, on both tasks, with significant improvement compared to state-of-the-art methods in the pan-cancer setting for classification and survival analysis. We also show the transfer-learning capabilities of the model in the survival analysis setting on unseen cohorts. Code available at `https://github.com/instadeepai/multiomics-open-research`

1 Introduction

New high-throughput technologies have revolutionized multi-omics research by providing extensive genomic and transcriptomic data. These have proved to be essential in oncology to understand the molecular foundations of the disease and thus may be informative for cancer prognosis [28] [8], notably for cancer type classification or survival analysis. The Cancer Genome Atlas (TCGA, `https://portal.gdc.cancer.gov/`) is a publicly available dataset that gathers multi-omics data such as genome sequencing, RNA sequencing, DNA methylation, or whole slide images from thousands of tumor samples. This dataset, divided into 33 cohorts, also provides clinical information such as survival time, making it a widely used dataset for the benchmark of survival analysis and cancer type classification methods. Survival analysis, or time-to-event prediction, aims at predicting the time from diagnosis to patient death from the disease using censored data. The classic approach for this task has long been the semi-parametric Cox proportional hazards (CPH) model [6]. More recently, deep-learning methods such as *Cox-nnet* [4] or *DeepSurv* [19] estimate the log-risk function of the CPH model using the Cox partial likelihood as the loss function. Compared to CPH, these models have the advantage of capturing non-linear patterns underlying the complexity of RNA-seq data.

Machine Learning in Computational Biology 2024.

However, the previous methods are confronted by the *curse of dimensionality* that comes with the high number of genes considered when using RNA-seq data and the small number of available samples. Though some *a priori* feature selection from domain knowledge (such as highly variable genes selection in the case of RNA-seq data) can be carried out, one tends to first learn representations of such high-dimensional data before applying them to any downstream task. Such representations are often obtained through statistical methods such as Principal Component Analysis [18] or Non-negative Matrix Factorization (NMF) [23] to take advantage of the positivity of gene expressions.

Nevertheless, deep learning-based methods have progressively emerged as state-of-the-art methods for representation learning and have already proven their capacity when applied to omics data. In [14], a robust evaluation of deep learning-based methods for survival analysis and gene essentiality prediction is presented, with methods mainly based on auto-encoders or graph neural networks. [2] presents a model called *CustOmics* based on Variational Auto-Encoders [21] that combines multiple sources of omics data (RNA-seq, Copy Variation Number, and Methylation) for cancer type prediction and survival analysis. They namely show in an ablation study that RNA-seq is the most informative modality for predicting the phenotype of interest, thus motivating our choice of building a model exclusively based on RNA-seq. [29] also consider a multi-modal model, including especially RNA-seq and whole slide images for survival analysis. Finally, [31] leverages multi-task learning using the tabular form of RNA-seq data and obtains good performance in survival analysis with a pre-selection of a small subset of genes. Motivated by their wide success in Natural Language Processing, Transformers [30] models have been used in a variety of domains (image [12], audio, ...) by further empowering self-supervision to build foundation models. Thanks to the large amount of available unannotated data, these models have already had an impact in the computational biology field, especially in the analysis of DNA sequences [9] [17]. More recently, transformer-based models have been used for transcriptomics data to learn representations of single-cell RNA-seq data, with models like *scBERT* [33] and *scGPT* [7] with application to cell-type annotation for example, thus suggesting applying the same techniques on bulk RNA-seq data.

In this paper, we present a transformer-based encoder-only language model, *BulkRNABert*, pre-trained on bulk RNA-seq data through self-supervision using masked language modeling from BERT's method [11]. We will consider language models trained on cancer data from TCGA, but also models trained on non-cancer datasets, GTEx [3] and ENCODE [10]), both to increase the number of bulk RNA-seq samples available, but also to show that one can get good performance in cancer downstream tasks from non-cancer pre-trained models. The learned RNA-seq embeddings are then used to train separate models for the two downstream tasks that we will consider: cancer-type classification and survival analysis. For survival prediction, we will consider both the per-cohort and the pan-cancer setting, the latter allowing us to consider more samples, thus stabilizing models' training and also showing transfer learning to unseen cohorts. Moreover, for the classification problem, we will show that fine-tuning the language model through a parameter-efficient fine-tuning method (PEFT) [32] further increases our performance. Our method constitutes the first language model pre-trained on bulk RNA-seq data and tailored for cancer type classification and survival analysis, achieving state-of-the-art performance in the pan-cancer setting.

2 Methods

2.1 Language model pre-training

Language Models (LM) aim at approximating probability distributions over sequences of tokens. In the context of RNA-seq data, each sequence is composed of real values corresponding to gene expression (in *Transcript Per Million*, or *TPM*, units). The first step is thus to transform each gene expression sequence into a sequence of tokens; this is achieved by binning the TPM values on a linear scale. Thus, the token id associated with a given RNA-seq value is its corresponding bin id, so after tokenization one RNA-seq sample is a vector $\mathbf{X} \in [\![0\ ;\ B-1]\!]^N$, with N the number of considered genes and B the number of expression bins. Our architecture follows an encoder-only transformer trained using Masked Language Modelling, following a classic *BERT* [11] methodology. Compared to classic transformer architecture, as gene expression is permutation invariant, positional encoding is here replaced by a gene embedding added to the gene expression embedding. Gene embedding matrix, of size $N \times n_{gene_embedding}$, given $n_{gene_embedding}$ the dimension of gene embedding, is initialized with *Gene2Vec* method [35] as suggested in [33]. To avoid being restricted to the embedding dimension chosen by *Gene2Vec*, the obtained gene embedding is passed through an

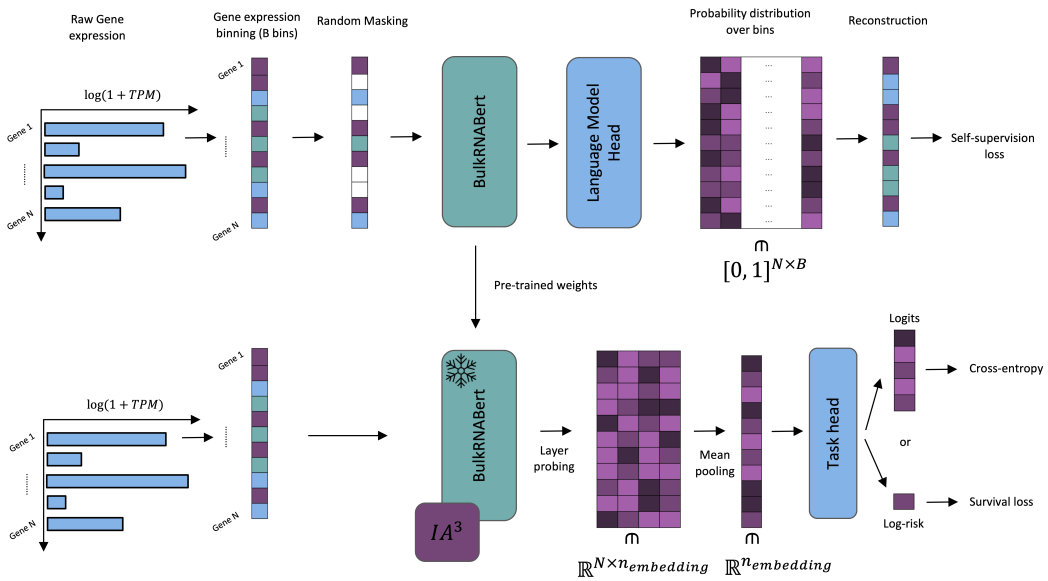

Figure 1: *BulkRNABert* pipeline: pre-training and task-specific fine-tuning.

additional linear layer to reach effective embedding size ($n_{embedding}$). The final embedding is then processed through a stack of self-attention layers before being transformed by a language model head to get a probability distribution over the RNA-seq bins for each token. We follow typical parameters for Masked Language Modeling: for a given sequence, 15% of tokens are corrupted and used to train the model. Out of those 15%, 80% is replaced by a special mask token [MASK], 10% of the 15% are replaced by random tokens, and the remaining 10% tokens are kept unchanged, but still counted in the masked language modeling loss. The following loss is then optimized:

$$\mathcal{L}_{MLM} = -\frac{1}{|\mathcal{M}|}\sum_{i\in\mathcal{M}} log(p_{i,y_i})$$

with $\mathcal{M} \subset [\![0\ ;\ N-1]\!]$ is the set of masked token indices, $y_i \in [\![0\ ;\ B-1]\!]$ is the target gene expression bins and $p \in \mathbb{R}^{N\times B}$ is the probability distribution output by the language model.

2.2 Cancer type classification

We evaluate our pre-training strategy on several downstream tasks. The first task that we consider is a classification task over cancer type. Indeed, cancer classification is vital for accurate diagnosis and therapy, yet classifications based on morphological or histological characteristics are limited and gene-expression-based methods prove to provide accurate results [26] [13]. The $TCGA$ dataset is divided into 33 cohorts, allowing classification using RNA-seq data with cohorts as labels. The output of the last self-attention layer is extracted and averaged across the sequence dimension and is used as the embedding for a given RNA-seq sample. Therefore, from a raw RNA-seq sample (a vector $\mathbb{R}^N$), one obtains a representation in $\mathbb{R}^{n_{embedding}}$. This embedding is then fed to a Multi-Layer Perceptron (MLP), with dropout [27] and layer normalization [1] to get the cancer type prediction.

2.3 Survival analysis

The second task is survival time prediction, *i.e.* we are interested in building a model that estimates the expected time until the occurrence of an event, in our case, the time until the death of a cancer patient from the time of diagnosis. Thus, we aim to develop a model to predict survival time T_i^* for a given individual i. One major issue in survival problems is the concept of right-censoring: the period of observation time C_i^* might be less than T_i^* (end of the study, lost contact with the patient). One thus defined $T_i = min(T_i^*, C_i^*)$ as the target survival time, as well as $\delta_i = \mathbb{1}_{T_i^* \leq C_i^*}$ (so $\delta_i = 1$ if

event occurred (death), otherwise $\delta_i = 0$), thus constituting a dataset $\mathcal{D} = \{T_i, x_i, \delta_i\}_{i=1}^N$, with x_i covariates (RNA-seq mean embeddings in our study). The most popular survival method to solve such a problem is the semi-parametric model called Cox proportional hazard model [6] in which one aims at modeling the hazard function λ, defined as the instantaneous event rate at time t given survival until time t ($\lambda(t) = \lim_{\delta t \to 0} \frac{\mathbb{P}(t<T\leq t+\delta t|T>t)}{\delta t}$) as $\lambda(t|x) = \lambda_0(t)e^{\hat{h}_\beta(x)}$, with β a vector of parameters (so in our case $x, \beta \in \mathbb{R}^{n_{embedding}}$), $\lambda_0(t)$ the baseline hazard, and in the Cox model, $\hat{h}_\beta(x) = \beta^T x$. To estimate the vector of parameters, maximum likelihood estimation is used, which is done in the Cox model by maximizing the following partial likelihood:

$$L(\beta) = \prod_{i|\delta_i=1} \frac{e^{\hat{h}_\beta(x_i)}}{\sum_{j\in\mathcal{R}_i} e^{\hat{h}_\beta(x_j)}} \quad with \quad \mathcal{R}_i = \{j|T_j \geq T_i\}$$

As done in other works, like DeepSurv [19] or Cox-nnet [4], we can replace $\hat{h}_\beta(x)$ by a neural network (same architecture as for the previous classification head) output and optimize negative partial log-likelihood:

$$\mathcal{L}_{survival} = -\sum_{i|\delta_i=1} \left(\hat{h}_\beta(x_i) - log \sum_{j\in\mathcal{R}_i} e^{\hat{h}_\beta(x_j)}\right)$$

The stability of this training method is dependent on the number of positive examples, and alternatives like L_2-like losses [16] could also be explored. This study is left for future work.

2.4 IA^3 fine-tuning

For the classification setting, we furthermore employ the IA^3 technique [24], a parameter-efficient fine-tuning method that adds new learnable weights to the transformer layers while keeping frozen the pre-trained ones. To this end, three learned vectors $l_k \in \mathbf{R}^{d_k}$, $l_v \in \mathbf{R}^{d_v}$, and $l_{ff} \in \mathbf{R}^{d_{ff}}$ are adjoined into the attention mechanism:

$$softmax\left(\frac{Q(l_k \odot K)}{\sqrt{d_k}}\right)(l_v \odot V)$$

and in the position-wise feed-forward networks through $(l_{ff}\gamma(W_1 x))W_2$, where γ is the feed-forward network non-linearity, and $\odot$ represents the element-wise multiplication. Compared to a full fine-tuning of the language model and the classification head, this method only adds $L(d_k + d_v + d_{ff})$ parameters (with L the number of self-attention layers), which represents less than 0.07% of the total number of model parameters, thus speeding up the fine-tuning. The same classification head is still used. Complete architecture, from pre-training to task-specific fine-tuning, is presented in Figure 1.

3 Experimental Results

3.1 Language Model pre-training

Datasets Three bulk RNA-seq datasets were considered to pre-train *BulkRNABert*: 2 non-cancer datasets, GTEx [3] and ENCODE [10], representing 20,406 samples, and one cancer dataset, TCGA (all cohorts are considered here), representing 11,274 samples. Among all the genes available in these 3 datasets, we keep all the genes that are common in all 3 datasets, representing a total of $N = 19{,}042$ genes. We use the transcript per millions (TPM) for the RNA-seq samples unit, and apply $x \mapsto log_{10}(1 + x)$ transformation followed by a max-normalization. The same transformations are applied to RNA-seq samples used in the downstream tasks in the following sections. Using these datasets, we were able to train three different language models: *BulkRNABert(GTEx ENCODE)* - pre-trained on GTEx and ENCODE only; *BulkRNABert(TCGA)* - pre-trained on TCGA only; and *BulkRNABert(GTEx ENCODE TCGA)* - pre-trained on the concatenation of GTEx, ENCODE, and TCGA. For the masked language modeling step, each dataset is split into a train and test set, keeping 5% of the dataset for the test set. The masked language modeling loss is optimized using *AdamW* [25] optimizer with gradient accumulation to reach an effective batch size of 3×10^6 tokens per batch, split onto 4 TPUs *v4-8*.

Pre-training We trained 3 models using the same set of hyperparameters. Following the presented tokenization method, RNA-seq samples are binned using $B = 64$ bins. Each model is composed

of 4 transformer blocks with 8 attention heads each and uses an embedding dimension of 256 (so $n_{embedding} = 256$). Models are trained on a total of $12B$ tokens. Unlike methods like [33], we decided not to consider only non-zero expressions in masked language modeling but all of them. Models are evaluated using the reconstruction accuracy. We provided in supplementary materials 5.1 the learning curves (in accuracy) of the 3 models: we observe that the reconstruction accuracy is higher when considering the non-cancer datasets GTEx and ENCODE than for TCGA, which can be qualitatively explained by the higher variability of gene expressions in cancerous tissues. Overall, all 3 models exhibit reconstruction accuracies between 0.5 and 0.6 which is a significantly great performance as we are using $B = 64$ bins, *i.e.* 64 possibilities for the labels (and $1/B \approx 0.016$). In addition to these 3 base models, noticing a large number of genes with very small variance, we decided to pre-train a 4th language model using the TCGA dataset, but relying only on highly variable (HV) genes, leading us to use $12{,}403$ genes. This model is referred to as ***BulkRNABert(TCGA - HV genes)***. Unlike methods that proceed to a selection of a very small subset of genes tailored for particular cohorts, we perform a soft dimension reduction as follows: for each cohort, we identify the set of genes whose expression's standard deviation is less than the 3rd quantile of the global cohort gene expressions' standard deviation distribution. We then remove from our initial $19{,}042$ genes the set of genes corresponding to the intersection of the set of genes obtained for each cohort following the method we just described, thus still leaving us with a significant number of genes.

3.2 Cancer-type classification

We then first assess the quality of the learned representations through cancer type classification, using TCGA cohorts as class labels. We consider two settings for this classification problem:

- 5 cohorts classification using BRCA, BLCA, GBMLGG, LUAD, and UCEC cohorts (GBMLGG is the fusion of two brain cancer cohorts, GBM and LGG. These are the cohorts considered in [2] for survival analysis, so as a first step we consider this simpler classification problem using these cohorts), see Appendix 5.2 for label distribution.
- Pan-cancer classification using the 33 cohorts, see Appendix 5.2 for label distribution.

Experimental setup Classification models consist of a Multi-Layer Perceptron (MLP) with two hidden layers - respectively with 256 and 128 units, using *SELU* [22] as activation function, followed by a classification head according to the number of classes. We also tried to fit a Support Vector Machine (SVM) [5] classifier instead of an MLP. We use cross-entropy loss, optimized using Adam [20] optimizer. The 5 cohorts classification problem does not involve a significant dataset imbalance, we thus report the weighted F_1 score as an evaluation metric. For the pan-cancer classification, we report both macro F_1 and weighted F_1 scores to avoid any bias due to class imbalance. For each task, the dataset is split into 80% train and 20% test. This is done with 5 different seeds, and the reported score corresponds to the mean score across these 5 seeds. The splits are stratified to ensure equal representation of each class in the train and test splits. Multiple RNA-seq samples might be available for a given patient; thus, we ensure that samples from the same patient are not mixed in the train and test sets.

5 cohorts classification We compare our method to *CustOmics*[2] and to two dimension reduction techniques (PCA and NMF) applied to raw RNA-seq and fed to an SVM. We present our results in Figure 2. We exhibit that only pre-training on non-cancer datasets (GTEx and ENCODE) already provides better results than CustOmics - and pre-training on cancer datasets even allows us to reach a nearly perfect score (weighted F_1 score of 0.993). One may notice that mixing cancerous and non-cancerous tissues in the pre-training does not bring any additional performance, which may root from the same number of parameters kept for the language model, hence a dilution of the data. Also, considering all genes and not only the highly variable ones provides the best results. More exhaustive experimental results, including training MLP without IA^3 method, are presented in Appendix 5.2.

Pan-cancer classification Results are presented in Table 1. Again, we demonstrate that a language model pre-trained on GTEx and ENCODE and fine-tuned using IA^3 goes head to head with *CustOmics*. Pre-training the language model on TCGA and fine-tuning it using IA^3 provides state-of-the-art results for cancer-type classification. Further improvements are shown (weighted F_1 of 0.97) by *CustOmics* when adding Copy Number Variation and Methylation. A confusion matrix for our best model is presented in Appendix 5.2, revealing that the remaining errors made by the model

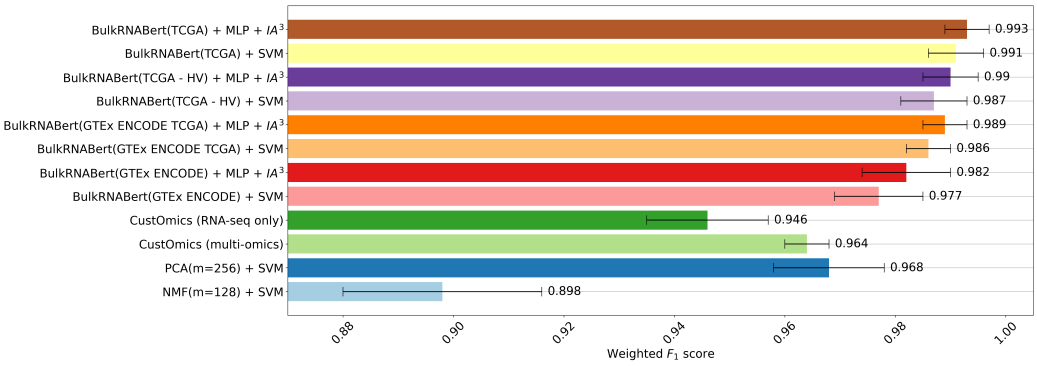

Figure 2: 5 cohorts classification results. "*BulkRNABert(Pre-Training Dataset)* + MLP/SVM" means that the MLP/SVM is trained, while freezing the language model weights. Adding "+ IA^3" indicates that the language models' weights are fine-tuned using the IA^3 method. m for PCA and NMF indicates the number of components.

are mostly due to very similar cancers (typically *READ* and *COAD*, respectively linked to rectum and colon).

Methods	test macro-F1	test weighted-F1
CustOmics (RNA-seq only) [2]	xx	0.930
NMF(m=128) + SVM	0.749 ± 0.006	0.80 ± 0.009
PCA(m=256) + SVM	0.849 ± 0.010	0.877 ± 0.006
BulkRNABert(GTEx ENCODE) + SVM	0.887 ± 0.005	0.918 ± 0.005
BulkRNABert(GTEx ENCODE) + MLP + IA^3	0.914 ± 0.007	0.935 ± 0.007
BulkRNABert(GTEx ENCODE TCGA) + SVM	0.894 ± 0.008	0.924 ± 0.005
BulkRNABert(GTEx ENCODE TCGA) + MLP + IA^3	0.914 ± 0.008	0.937 ± 0.003
BulkRNABert(TCGA - HV) + SVM	0.891 ± 0.010	0.924 ± 0.005
BulkRNABert(TCGA - HV) + MLP + IA^3	0.914 ± 0.004	0.936 ± 0.003
BulkRNABert(TCGA) + SVM	0.902 ± 0.007	0.933 ± 0.002
BulkRNABert(TCGA) + MLP + IA^3	**0.918 ± 0.006**	**0.942 ± 0.004**

Table 1: Pan-cancer classification. "xx" indicates that the score is not provided by the author.

3.3 Survival analysis

The performance analysis of *BulkRNABert* on survival prediction is divided into two parts: first, we consider pan-cancer models and then we dig into 5 per-cohort survival analyses.

3.3.1 Pan-cancer analysis

Experimental setup For the pan-cancer analysis, we trained survival models (same architecture as the classification head, with a slightly larger MLP with hidden layers sizes of [512, 256]), based on the different pre-trained language models. The pan-cancer dataset is composed of $11{,}035$ samples, split into train/test following the same experimental methodology as for the classification task.

Results We provide in Table 2 the pan-cancer survival results, comparing with *CustOmics* [2] and with Masking Auto-Encoder (MAE) from [14] inspired by a method designed to learn representations of tabular data [34]. Our models are evaluated using Harrell's C-index [15]. However, for the pan-cancer setting, we report two different C-indexes: C-index on the whole test set (all cohorts) referred to as "C-index", and a weighted sum of the C-indexes computed per cohort on the pan-cancer test set (with weights corresponding to the number of samples of each cohort in the test set), referred to as "Weighted C-index". The motivation behind providing these 2 metrics was to confirm that our model was not only learning to differentiate between cancer types without being able to correctly

predict survival time within a given cohort. We also provide in Appendix 5.3 the non-weighted average C-index ("Macro C-index"). Overall, our method exhibits state-of-the-art results using the language model pre-trained on TCGA, with further improvements when only looking at highly variable genes, showing that manually reducing the dimension with domain knowledge reduces the problem complexity. One notices that the survival model built on top of *BulkRNABert(GTEx ENCODE)* provides results very close to the baselines we are comparing with, thus highlighting the effectiveness of the pre-training. We keep in mind that *CustOmics* reaches a weighted test C-index of 0.68 when adding other omics sources, which paves the way for further improvement.

Table 2: Pan-cancer survival analysis

Methods	C-index	Weighted C-index
CustOmics (RNA-seq only) [2]	xx	0.630 ± 0.020
MAE [14]	0.756 ± 0.010	xx
BulkRNABert(GTEx ENCODE)	0.753 ± 0.009	0.621 ± 0.015
BulkRNABert(GTEx ENCODE TCGA)	0.762 ± 0.009	0.641 ± 0.018
BulkRNABert (TCGA)	**0.765 ± 0.011**	0.642 ± 0.014
BulkRNABert(TCGA - HV genes)	0.764 ± 0.010	**0.646 ± 0.012**

3.3.2 Per-cohort analysis analysis.

Experimental setup The same analysis is performed by considering survival models trained on specific cohorts: BLCA, BRCA, GBMLGG, LUAD, UCEC. Due to the smaller number of samples available per-cohort compared to the pan-cancer setting, training a model for a particular cohort is a harder problem (see Appendix 5.3 for the exact number of samples per cohort in the survival setting).

Results We compare in Table 3 our method to the same baselines (*CustOmic* and MAE), to which we add a comparison with a simple MLP (same architecture as the one used on top of the language model) trained on raw RNA-seq data, which proves to be the best on BRCA. However, our method (using pre-training on TCGA) reaches state-of-the-art results for the BLCA and LUAD cohorts, and brings a significant improvement on GBMLGG and UCEC cohorts, with slight improvement from *BulkRNABert(TCGA)* when considering the highly variable genes on BRCA and BLCA. One notices that we even reach better results than *CustOmics* trained on multiple omics sources on 3 cohorts.

Table 3: Per-cohort survival analysis

Methods	BLCA	BRCA
Baseline	0.598 ± 0.034	**0.676 ± 0.057**
CustOmics (multi-omics) [2]	**0.637 ± 0.050**	0.633 ± 0.018
CustOmics (RNA-seq only) [2]	0.626 ± 0.071	0.611 ± 0.022
MAE [14]	0.615 ± 0.039	0.632 ± 0.044
BulkRNABert(TCGA)	0.627 ± 0.069	0.604 ± 0.032
BulkRNABert(TCGA - HV genes)	0.636 ± 0.065	0.621 ± 0.032

Methods	GBMLGG	LUAD	UCEC
Baseline	0.831 ± 0.016	0.550 ± 0.053	0.609 ± 0.065
CustOmics (multi-omics) [2]	0.642 ± 0.028	0.625 ± 0.037	0.667 ± 0.022
CustOmics (RNA-seq only) [2]	0.609 ± 0.057	0.638 ± 0.040	0.660 ± 0.032
MAE [14]	0.775 ± 0.038	0.621 ± 0.070	0.633 ± 0.071
BulkRNABert(TCGA)	**0.844 ± 0.023**	**0.648 ± 0.057**	0.703 ± 0.040
BulkRNABert(TCGA - HV genes)	0.842 ± 0.028	0.638 ± 0.071	**0.707 ± 0.028**

3.3.3 Transfer learning from pan-cancer to per-cohort.

Experimental setup We conducted a final experiment in which we wanted to further attest the performance of pan-cancer models on particular cohorts (in addition to the previously reported "Weighted Test C-index" metric). To this end, we compare 3 models for a given cohort as shown in

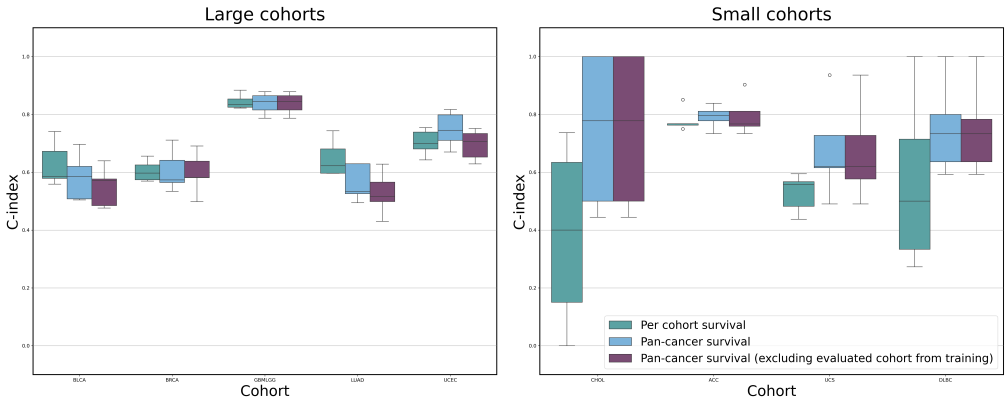

Figure 3: Survival analysis: comparison of per-cohort and pan-cancer models evaluated on selected cohort. (left: evaluation on BLCA, BRCA, GBMLGG, LUAD and UCEC. right: evaluation on 4 smallest TCGA cohort, CHOL, ACC, UCS and DLBC). First box plot: per-cohort model. Second box plot: pan-cancer model (*BulkRNABert(TCGA)*. Third box plot: pan-cancer (*BulkRNABert(TCGA)*) model with evaluated cohort excluded from training set.

Figure 3: a per-cohort survival model corresponding to a model trained on that particular cohort from a language model pre-trained on TCGA, a pan-cancer model (*BulkRNABert(TCGA)*) from which we only report the C-index on its test set restricted to the cohort of interest, and finally the same pan-cancer model but whose training set has been pruned of any samples of the cohort of interest. This experiment has been carried out on the 5 previous cohorts but also on the 4 smallest cohorts of the TCGA datasets: CHOL, ACC, UCS, and DLBC.

Results For the 5 first cohorts, one notices that pan-cancer models, though sometimes providing slightly worse scores, reach C-indexes that are very close to the per-cohort setting, even when the cohort of interest had been removed from the training set: this shows transfer learning capabilities of our method by being able to infer survival time on unseen cohorts. Finally, for the UCEC cohort, an improvement is observable from the per-cohort to the pan-cancer model, which may be rooted in the fact that UCEC is a highly censored dataset (only 16% of non-censored data). The same conclusion applies to the 4 smallest cohorts, for which one makes a significant improvement when considering pan-cancer models, showing that survival signals from other cohorts may benefit others.

4 Conclusion

In this work we present *BulkRNABert*, the first language model pre-trained on bulk RNA-seq data and fine-tuned on cancer type classification and survival analysis. We came up with new state-of-the-art results on both tasks, with significant improvements in the pan-cancer settings, showing the strength of our pre-training method. We additionally showcase the transfer learning capabilities of our pan-cancer survival models to unseen cohorts. Further improvements may involve refining the results in the per-cohort setting, especially trying to understand the grounding of the great performance of baselines for the BRCA cohort. This may call for leveraging some interpretation methods, in particular quantifying the importance of each gene in survival predictions. Also, as shown in [2], the addition of other omics sources exhibits an increase in performance on both downstream tasks. We are thus interested in leveraging genomics and imaging data to consider multi-modal models. Finally, though only pre-training with non-cancerous RNA-seq data (GTEx and ENCODE) already provides models that go head to head with state-of-the-art methods, we did not observe further improvement by mixing these data with cancerous data compared to models directly pre-trained on TCGA. Scaling up the language models could help cope with it.

Acknowledgments

Research supported by Cloud TPUs from Google's TPU Research Cloud (TRC).

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Hakim Benkirane, Yoann Pradat, Stefan Michiels, and Paul-Henry Cournède. Customics: A versatile deep-learning based strategy for multi-omics integration. *PLOS Computational Biology*, 19(3):e1010921, 2023.

[3] Latarsha J Carithers and Helen M Moore. The genotype-tissue expression (gtex) project. *Biopreservation and biobanking*, 13(5):307, 2015.

[4] 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.

[5] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

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

[7] Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature Methods*, pages 1–11, 2024.

[8] Xiaofeng Dai and Li Shen. Advances and trends in omics technology development. *Frontiers in Medicine*, 9:911861, 2022.

[9] Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P de Almeida, Hassan Sirelkhatim, et al. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *bioRxiv*, pages 2023–01, 2023.

[10] Natalie de Souza. The encode project. *Nature methods*, 9(11):1046–1046, 2012.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] Pilar García-Díaz, Isabel Sánchez-Berriel, Juan A Martínez-Rojas, and Ana M Diez-Pascual. Unsupervised feature selection algorithm for multiclass cancer classification of gene expression rna-seq data. *Genomics*, 112(2):1916–1925, 2020.

[14] Baptiste Gross, Antonin Dauvin, Vincent Cabeli, Virgilio Kmetzsch, Jean El Khoury, Gaetan Dissez, Khalil Ouardini, Simon Grouard, Alec Davi, Regis Loeb, et al. Robust evaluation of deep learning-based representation methods for survival and gene essentiality prediction on bulk rna-seq data. *bioRxiv*, pages 2024–01, 2024.

[15] 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.

[16] Jong-Hyeon Jeong and Yichen Jia. Deepcent: Prediction of censored event time via deep learning. *arXiv preprint arXiv:2202.05155*, 2022.

[17] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021.

[18] Ian T Jolliffe. *Principal component analysis for special types of data*. Springer, 2002.

[19] 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*, 18:1–12, 2018.

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

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

[22] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. *Advances in neural information processing systems*, 30, 2017.

[23] Daniel Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2000.

[24] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022.

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[26] Mohanad Mohammed, Henry Mwambi, Innocent B Mboya, Murtada K Elbashir, and Bernard Omolo. A stacking ensemble deep learning approach to cancer type classification based on tcga data. *Scientific reports*, 11(1):15626, 2021.

[27] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[28] Rory Stark, Marta Grzelak, and James Hadfield. Rna sequencing: the teenage years. *Nature Reviews Genetics*, 20(11):631–656, 2019.

[29] Luís A Vale-Silva and Karl Rohr. Long-term cancer survival prediction using multimodal deep learning. *Scientific Reports*, 11(1):13505, 2021.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[31] You Wu, Omid Bazgir, Yongju Lee, Tommaso Biancalani, James Lu, and Ehsan Hajiramezanali. Multitask-guided self-supervised tabular learning for patient-specific survival prediction. In *Machine Learning in Computational Biology*, pages 10–22. PMLR, 2024.

[32] Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*, 2023.

[33] Fan Yang, Wenchuan Wang, Fang Wang, Yuan Fang, Duyu Tang, Junzhou Huang, Hui Lu, and Jianhua Yao. scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. *Nature Machine Intelligence*, 4(10):852–866, 2022.

[34] Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela Van der Schaar. Vime: Extending the success of self-and semi-supervised learning to tabular domain. *Advances in Neural Information Processing Systems*, 33:11033–11043, 2020.

[35] Quan Zou, Pengwei Xing, Leyi Wei, and Bin Liu. Gene2vec: gene subsequence embedding for prediction of mammalian n6-methyladenosine sites from mrna. *Rna*, 25(2):205–218, 2019.

5 Supplementary Materials

5.1 Language model pre-training

In Figure 4, we provide the learning curves, in terms of reconstruction accuracy, of the masked language modeling task for the 3 *BulkRNABert* models corresponding to the 3 pre-training datasets: *BulkRNABert(GTEx ENCODE)*, *BulkRNABert(GTEx ENCODE TCGA)*, and *BulkRNABert(TCGA)*.

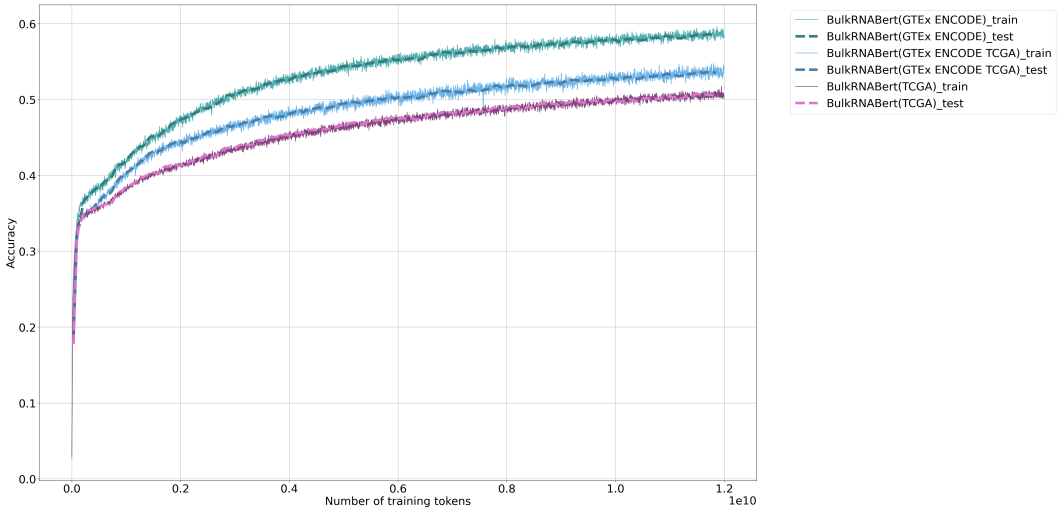

Figure 4: BulkRNABert pre-training with Masked Language Modeling on 3 different datasets. One observes higher reconstruction accuracy on non-cancerous tissues (GTEx ENCODE) than cancerous tissues (TCGA).

5.2 Cancer type classification

For the cancer type classification task, we report the label distribution for both the 5 cohorts setting (Figure 5) and the pan-cancer setting (Figure 6). The latter shows a high class imbalance, hence the need to report both weighted and macro F_1 scores. We furthermore report in Table 4 the performance of *BulkRNABert* on the 5 cohorts classification problem, completed with the results where we only train a MLP without using the IA^3 method. We thus observe a slight improvement in mean weighted F_1 score when using the parameter-efficient method. Finally, a confusion matrix for the pan-cancer setting is provided in Figure 7 for the *BulkRNABert(TCGA) + MLP + IA3* model, showing great classification performance with errors mainly between very similar cancer types (*e.g* between COAD - colon cancer - and READ - rectum cancer).

5 cohorts classification

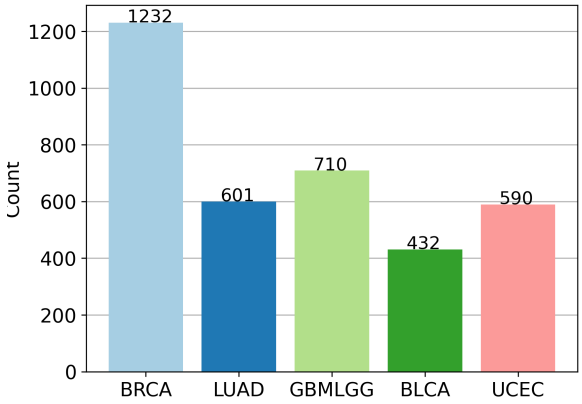

Figure 5: 5 cohorts classification label distribution.

Methods	test weighted-F1
NMF(m=128) + SVM	0.898 ± 0.018
PCA(m=256) + SVM	0.968 ± 0.010
CustOmics (RNA-seq only) [2]	0.946 ± 0.011
CustOmics (multi-omics) [2]	0.964 ± 0.004
BulkRNABert(GTEx ENCODE) + SVM	0.977 ± 0.008
BulkRNABert(GTEx ENCODE) + MLP	0.965 ± 0.010
BulkRNABert(GTEx ENCODE) + MLP + IA^3	0.982 ± 0.008
BulkRNABert(GTEx ENCODE TCGA) + SVM	0.986 ± 0.004
BulkRNABert(GTEx ENCODE TCGA) + MLP	0.982 ± 0.006
BulkRNABert(GTEx ENCODE TCGA) + MLP + IA^3	0.989 ± 0.004
BulkRNABert(TCGA - HV) + SVM	0.987 ± 0.006
BulkRNABert(TCGA - HV) + MLP	0.989 ± 0.006
BulkRNABert(TCGA - HV) + MLP + IA^3	0.990 ± 0.005
BulkRNABert(TCGA) + SVM	0.991 ± 0.005
BulkRNABert(TCGA) + MLP	0.991 ± 0.004
BulkRNABert(TCGA) + MLP + IA^3	**0.993 ± 0.004**

Table 4: 5 cohorts classification complete results. "*BulkRNABert(Pre-Training Dataset)* + MLP/SVM" means that the MLP/SVM is trained, while freezing the language model weights. Adding "+ IA^3" indicates that the language models' weights are fine-tuned using the IA^3 method. m for PCA and NMF indicates the number of components.

Pan-cancer classification

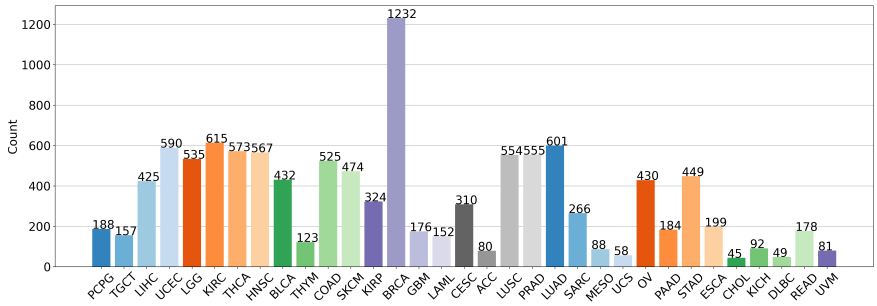

Figure 6: Pan-pancancer classification label distribution.

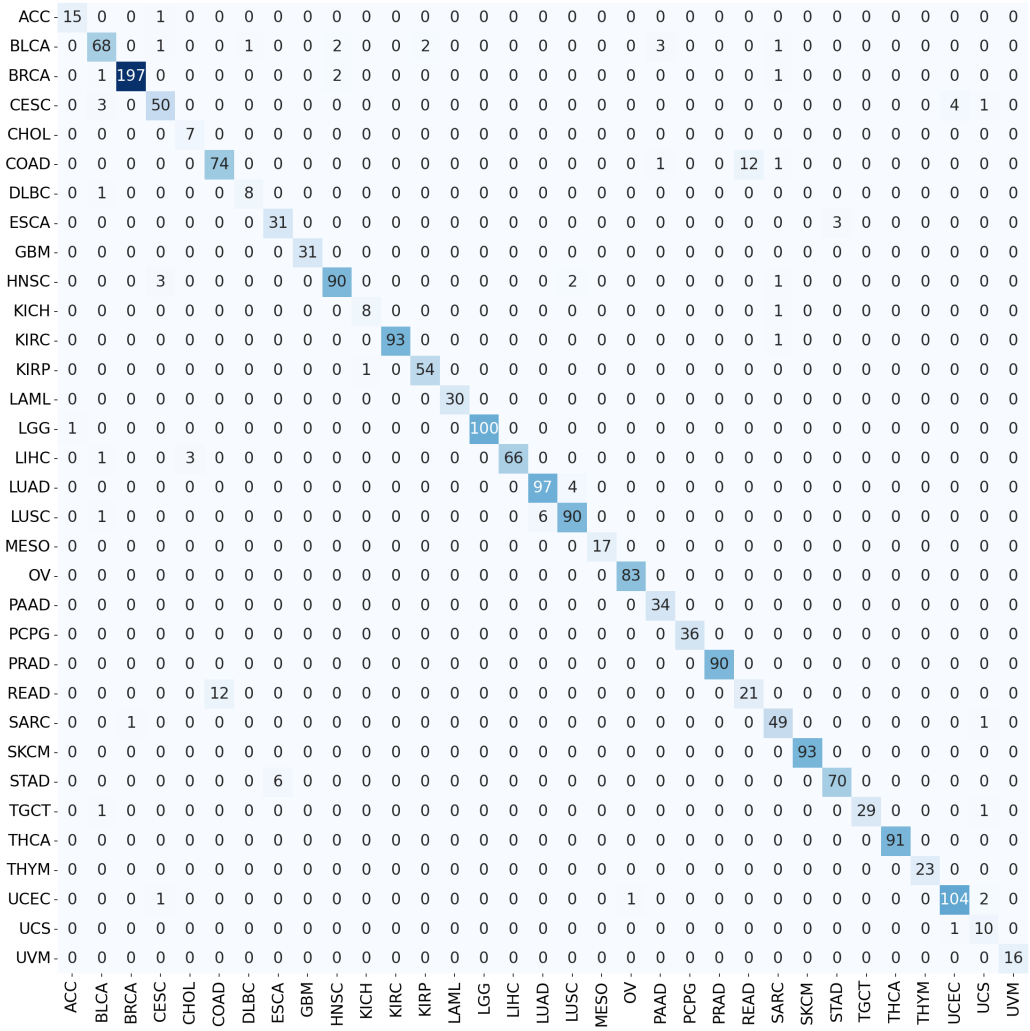

Figure 7: Confusion matrix for the pan-cancer classification problem for the model *BulkRNA-Bert(TCGA) + MLP + IA^3*

.

5.3 Survival Analysis

We additionally provide in Figure 8 the number of available samples in the TCGA datasets for survival analysis for the specific cohorts we are interested in, both for the 5 large cohorts that guided our study (BRCA, BLCA, GBMLGG, LUAD, and UCEC), and for the 4 smallest TCGA cohorts (CHOL, ACC, UCS, and DLBC). Finally, we present again in Table 5 the performance of the pan-cancer survival models, adding the "Macro C-index" corresponding to the mean of the C-indexes computed by cohort on the test set of the pan-cancer dataset.

Table 5: Pan-cancer survival analysis

Methods	C-index	Weighted C-index	Macro C-index
CustOmics (RNA-seq only) [2]	xx	0.630 ± 0.020	xx
MAE [14]	0.756 ± 0.010	xx	xx
BulkRNABert(GTEx ENCODE)	0.753 ± 0.009	0.621 ± 0.015	0.633 ± 0.026
BulkRNABert(GTEx ENCODE TCGA)	0.762 ± 0.009	0.641 ± 0.018	0.649 ± 0.034
BulkRNABert (TCGA)	**0.765 ± 0.011**	0.642 ± 0.014	0.656 ± 0.025
BulkRNABert(TCGA - HV genes)	0.764 ± 0.010	**0.646 ± 0.012**	**0.657 ± 0.025**

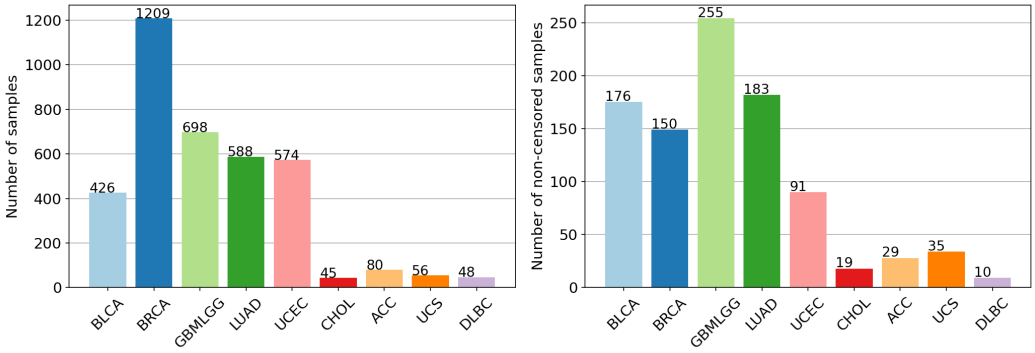

Figure 8: Per-cohort survival analysis: number of samples per cohort (left), and number of non-censored samples per cohort (right).

5.4 TCGA cohort abbreviations

Abbreviation	Name
BLCA	Bladder Urothelial Carcinoma
BRCA	Breast invasive carcinoma
GBM	Glioblastoma multiforme
LGG	Brain Lower Grade Glioma
LUAD	Lung adenocarcinoma
UCEC	Uterine Corpus Endometrial Carcinoma
CHOL	Cholangiocarcinoma
ACC	Adrenocortical carcinoma
UCS	Uterine Carcinosarcoma
DLBC	Lymphoid Neoplasm Diffuse Large B-cell Lymphoma

Table 6: TCGA cohort abbreviations