*Article*

# Performance comparison of deep learning autoencoders for cancer subtype detection using multi-omics data

**Edian F. Franco [1,2,3,‡], Pratip Rana [4,‡], Aline Cruz [5], Víctor V. Calderón [3], Vasco Azevedo [6], Preetam Ghosh [4,‡], and Rommel T. J. Ramos [3,‡] ***

[1] Institute of Biological Sciences, Federal University of Para, Belem-PA 66075-110, Brazil
[2] Laboratory of Virology and Environmental Genomics, Instituto de Innovacion en Biotecnologia e Industria (IIBI), San Geronimo, Santo Domingo, DN, Dominican Republic.
[3] Instituto Tecnológico de Santo Domingo (INTEC), Santo Domingo, Dominican Republic
[4] Department of Computer Science, Virginia Commonwealth University, Richmond, VA 23284, USA
[5] Programa de Pós-Graduação em Enfermagem, Federal University of Para, Belem-PA 66075-110, Brazil
[6] Institute of Biological Science, Federal University of Minas Gerais, Belo Horizonte-MG 31270-901, Brazil
* Correspondence: rommelramos@ufpa.br / rommelthiago@gmail.com

**Abstract:** A heterogeneous disease like cancer is activated through multiple pathways and different perturbations. Depending upon the activated pathway(s), patients' survival vary significantly and show different efficacy to various drugs. Therefore, cancer subtype detection using genomics level data is a significant research problem. Subtype detection is often a complex problem, and in most cases, needs multi-omics data fusion to achieve accurate subtyping. Different data fusion and subtyping approaches have been proposed, such as kernel-based fusion, matrix factorization, and deep learning autoencoders. In this paper, we compared the performance of different deep learning autoencoders for cancer subtype detection. We performed cancer subtype detection on four different cancer types from The Cancer Genome Atlas (TCGA) datasets using four autoencoder implementations. We also predicted the optimal number of subtypes in a cancer type using the silhouette score. We observed that the detected subtypes exhibit significant differences in survival profiles. Furthermore, we also compared the effect of feature selection and similarity measures for subtype detection. To evaluate the results obtained, we selected the Glioblastoma multiforme (GBM) dataset and identified the differentially expressed genes in each of the subtypes identified by the autoencoders; the obtained results coincide well with other genomic studies and can be corroborated with the involved pathways and biological functions. Thus, it shows that the results from the autoencoders, obtained through the interaction of different datatypes of cancer, can be used for the prediction and characterization of patient subgroups and survival profiles.

**Keywords:** Cancer subtype detection; Multi-omics data; Data integration; Autoencoder; Survival analysis)



**Citation:** Franco, E.; Rana, P.; Cruz Ramos, A.; Vasco, A.; Preetam, G.; Ramos, R. Autoencoders for cancer data fusion]. *Preprints* **2021**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:



## 1. Introduction

Due to technological advancement and decreasing costs, high-throughput sequencing technology such as RNA-seq, SNP-chip, UPLC-MS, and GC-MS techniques generate extensive and diverse amounts of omics data that allow biologists to understand the different processes and interactions within biological organisms with unprecedented detail. These omics technologies provide the ability to interpret and explain the genome through DNA sequencing, genome expression based transcriptome studies, protein identification from the proteome, and others. However, such individual data can only provide a limited information of the molecular complexity occurring inside the organisms due to the multi-level regulation inside biological units [1]. For example, we observe the combined effects of transcripts and methylome in the tumor cell due to genomics defect [2]. Considering gene expression data alone ignores the effect of point mutation, which alters the efficacy of gene products [3]. Furthermore, the dimension and the diversity of such data make it extremely challenging to perform proper data handling and in-depth analysis. Hence, there is an urgent requirement for mathematical models that can efficiently fuse these diverse

 doi:10.20944/preprints202102.0365.v1



molecular data from different measurements and provide us with a comprehensive and robust insight of biological phenotypes.

Ritchie et al. [4] define multi-omics data integration as the method in which diverse types of omics data are combined as predictor variables to allow more accurate and extensive modeling of complex traits or phenotypes. The integrated multi-omics methods permit the identification of crucial genomic factors and biomarkers, generate models to explain and predict disease risk, and help to understand the genetics and genomics architecture of complex phenotypes. Such integrated data also provide a holistic view of the biological system compared with traditional data-based methods [4–7]. Several data fusion models have been proposed recently, which fall into the following three categories: a) early fusion, b) intermediate fusion, and c) late fusion. One example of a data fusion algorithm is similarity network fusion (SNF) [8]. Here, diverse types of data are first normalized into a network form through a nonlinear kernel function. Next, SNF efficiently fuses these networks through an iterative fusion algorithm. Recently, deep-learning framework autoencoder also exhibited significant potential as a data fusion algorithm. An autoencoder reconstructs its input by a nonlinear transformation of its original input features. Hence, in this process, the autoencoder generates new nonlinear features from its original input feature-set. Several cancer studies used autoencoders to analyze multi-omics data. Autoencoder based data integration has been successfully applied to predict drug response [9] and kidney graft survival analysis [10]. The autoencoder is an unsupervised deep learning (DL) algorithm for dimensionality reduction and heterogeneous data integration based on feed-forward neuronal networks [11]. Autoencoders automatically learn nonlinear features from the unlabeled dataset after setting the output value equal to the input value. An autoencoder is constructed through the hooking of simple neurons together where the output of one neuron is the input to the other neurons. The network forms a "butterfly" structure, where the number of inputs is equal to the number of outputs, and in the middle, the autoencoder has bottleneck hidden layers. This design drives the network to seek a compressed representation of the data while preserving the input data's most important features (Figure 1). The architecture of an autoencoder allows it to concatenate the features and information of different omics sources [12–15].

A critical application of such data fusion algorithms is cancer subtype detection using omics data. Multiple oncogenes are involved in a heterogeneous disease like cancer, and they are perturbed through several pathways. Cancer patients' severity and their survival also differ considerably depending upon this perturbation. For example, glioblastoma multiforme (GBM) has four established subtypes: Classical, Mesenchymal, Neural, and Proneural. Subtype detection is a complex problem and frequently require fusion of the various heterogeneous datasets. Recently autoencoders were also used for subtype detection problems for Liver cancer by fusing three heterogeneous data types. For example, [16] used autoencoders on methylation, RNA-seq, and miRNA-Seq data from liver cancer patients to develop a robust model to predict two distinct survival groups. Also, [17] used the denoising autoencoder to develop a model that can identify and extract an intricate pattern from omics data in breast cancer. Deep learning autoencoders were also used for subtype classification in colorectal cancer using multi-omics data [18], while [14] applied autoencoders to identify two subtypes in neuroblastoma.

In [19], the authors used a variational autoencoder to integrate multi-omic cancer data. The model was used to develop pan-cancer classification analysis and obtained an average precision of 97.49% after 10-fold cross-validation of 33 tumor types and normal samples. The authors in [20] explored the different architectures, designs, and construction of multi-omic data integration using Variational Autoencoders; they demonstrated that autoencoders are suitable methods for representing data and the production of stable and accurate diagnostics. To study genes that mediate human lung adenocarcinoma, a model was created based on the denoising autoencoder, which allowed the identification of more positive genes related to this type of cancer compared to other methods [21].

 doi:10.20944/preprints202102.0365.v1



Depending upon the deep learning layer construction and regularization, an autoencoder can be of different types. The popular types of autoencoders are vanilla autoencoder, denoising autoencoder, sparse autoencoder and variational autoencoder. Though autoencoders showed promises for data fusion and subtype detection in the recent past, the different types of autoencoders' performance on the different datasets are still unknown. In this work, we compared the performance of four different autoencoders to integrate and reduce multi-omics data. By data fusion, autoencoders created new features to represent the input datasets. The new features were used to implement a survival-based clustering algorithm to define groups of patients with a similar distribution of features and survival prognosis. We evaluated the efficiency of the different autoencoders (vanilla, denoising, sparse and variational) for the fusion and reduction of cancer data dimensions from different sources such as RNA-seq, methylation and miRNA-Seq, on various types of cancer.

## 2. Materials and Methods

### *2.1. Dataset and prepossessing*

We obtained the multi-omics cancer data from The Cancer Genome Atlas Program (TCGA) database. TCGA integrates more than 20,000 primary cancer samples over 33 cancer types. We applied autoencoder based subtyping on four cancer types comprising the cancer dataset of Glioblastoma multiforme (GBM) and Colon Adenocarcinoma (COAD) from TGCA. While the other two datasets were for Kidney renal clear cell carcinoma (KRCC) and Breast invasive carcinoma (BIC) that were reprocessed by [8]. We utilized three types of data: gene expression, DNA methylation, and miRNA expression.

Glioblastoma multiforme (GBM) is one of the most aggressive brain tumors; the diagnosed patients have a survival estimate of 13 months on average, even after chemotherapy and radiotherapy treatments. We analyzed 276 patients from this cancer type, of which 164 were male and 112 were female. Dataset comprising mRNA expression (17,814 features), miRNA expression (470 features), and DNA methylation (13,000 features) from this group was used to compare the autoencoders.

The invasive breast carcinoma (BIC) is one of the most common types of breast cancer; about 80% of breast cancers are invasive [22]. The BIC data from [8] was also used that contains 106 patients, with 335 features in the miRNA dataset, 23,094 features for DNA methylation, and 17,814 for mRNA gene expression.

Colon Adenocarcinoma (COAD) is a type of cancer that usually arises from the epithelium lining inside the large intestine. This type of cancer is prevalent in the population that is more than 50 years old and in countries with a low fiber diet as Europe, the US, and Australia. COAD represented approximately 10% of diagnosed cancers [23,24]. We used the data from 92 patients with 17,814 features for mRNA expression, methylation dataset with 23,087 features, and miRNA expression with 311 features from COAD.

Kidney renal clear cell carcinoma (KRCC) is the most common type of kidney cancer. This cancer affects the lining cell and small tubules that filter waste from the blood and make urine. This type of cancer is more usual in men over 55 years of age [25,26]. The KRCC datasets contain the information of 122 patients, with 17,898 features in the mRNA dataset, 24,959 in the methylation dataset, and 329 features in the miRNA dataset.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 February 2021 doi:10.20944/preprints202102.0365.v1



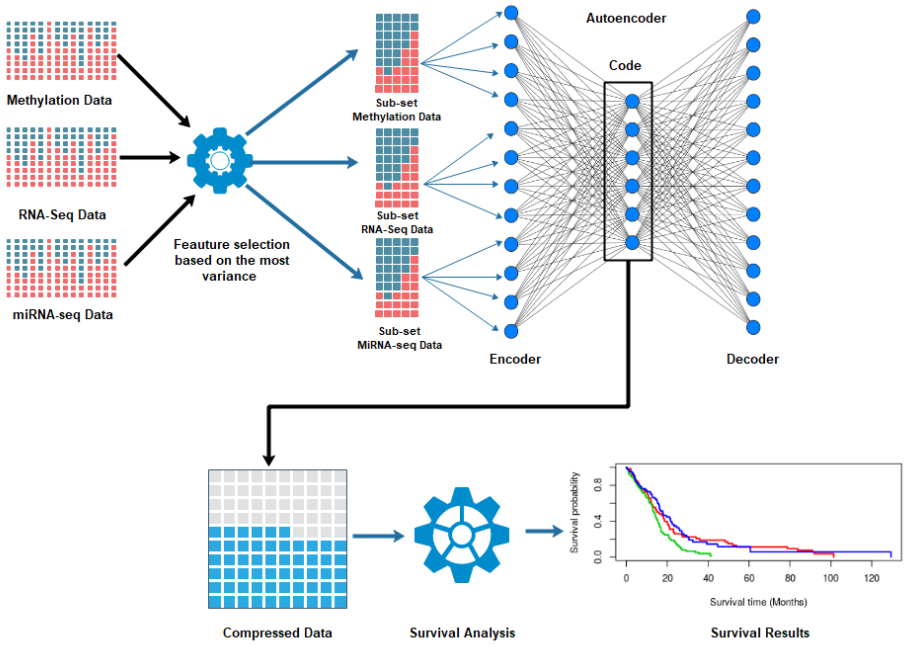


**Figure 1.** The workflow of subtype detection using autoencoders. The first step is feature selection from the multi-omics data of the same patients from the TCGA database. Next, autoencoders fuse the selected features by encoding and decoding. Next, we run two clustering algorithms on the patient similarity networks constructed from the bottleneck layer to identify the subtypes of cancer. Lastly, we run a survival analysis of the identified clusters to validate the result.

TCGA dataset comprising gene expression, DNA methylation, and miRNA expression was downloaded from TCGA using the TCGAbiolink package [27]. Next, we chose the common patients in these datasets for our analysis. To perform survival analysis, we also downloaded the patients' clinical data.

Next, we scaled each data using the following equation.

$$X_n = \frac{X_i - x_{min}}{x_{max} - x_{min}} \tag{1}$$

where $X_i$ is the data instance while $x_{max}$ and $x_{min}$ are the minimum and maximum absolute value of feature $X$, and $X_n$ is the feature after normalization. Next, we choose $100/400/500$ number of important features from each dataset based on maximum variance (VAR) using the function FSbyVar from the CancerSubtypes package in R [28] (Figure 1). These selected features were fed as input to the autoencoders.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 February 2021 doi:10.20944/preprints202102.0365.v1



### 2.2. Autoencoder construction

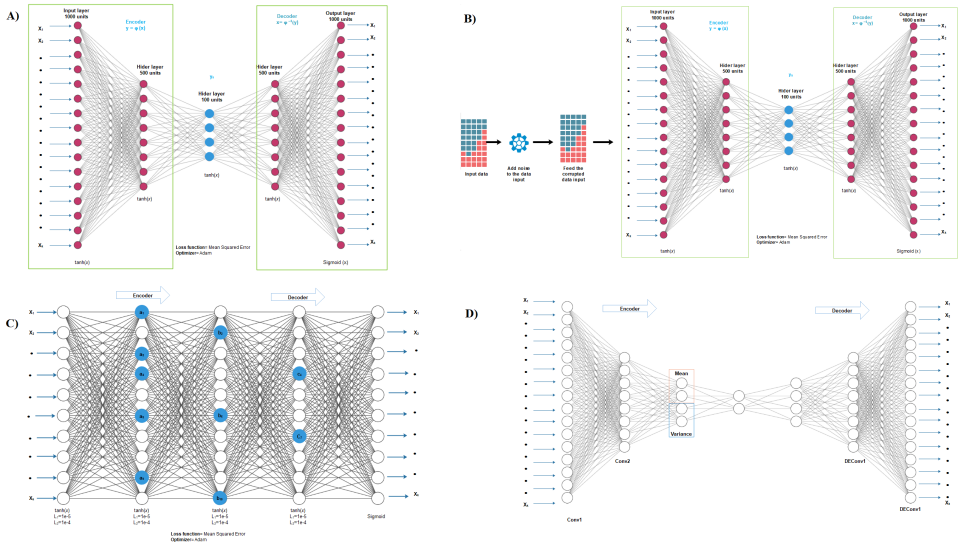

**Figure 2.** Autoencoder configuration: A) Vanilla autoencoder; B) Denoising autoencoder; C) Sparse autoencoder and D) Variational autoencoder.

By construction, an autoencoder can be of different types (Figure 2). One simple form of the autoencoder is vanilla autoencoder, traditionally which is constructed with a single layer of encoder and decoder. The learning minimizes the following loss function

$$L(x, g(f(x))) \tag{2}$$

where $L$ is the loss function of input $x$ and output $g(f(x))$. Due to the nonlinearity of the encoder and decoder's activation function, a vanilla encoder is allowed to learn nonlinear features from the data, which is not feasible from linear feature deduction methods like Principal Component Analysis (PCA) [29]. A vanilla autoencoder with multiple hidden layers is called a deep vanilla autoencoder.

Though vanilla autoencoder is simple, there is a high possibility of over-fitting. Denoising autoencoder, sparse autoencoder, and variational autoencoder are the regularized version of the vanilla autoencoder. Denoising Autoencoder reconstructs the original input from a corrupt copy of the input. Hence, a denoising autoencoder minimizes the following loss function.

$$L(x, g(f(\widetilde{x}))) \tag{3}$$

where $L$ is the loss function of input $x$ and output $g(f(\widetilde{x}))$. A corrupt copy of input is formed by introducing some noise with the original input. Denoising is achieved through stochastic mapping by setting some inputs to zero. The added noise helps the autoencoder to learn features without learning the original features directly from the data.

Sparse autoencoder adds a sparsity penalty $\Omega(h)$ on the bottleneck layer. The training of sparse autoencoder minimizes the following loss function.

$$L(x, g(f(x))) + \Omega(h) \tag{4}$$

We can consider the sparse autoencoder as a regularized version of vanilla autoencoder with a sparsity penalty of $\Omega(h)$. This representation is especially beneficial to learn the important features of data even when we set a large number of hidden units in the autoencoder.

Variational autoencoder uses a strong assumption about latent variables by generally using a latent Gaussian distribution [30] [31]. It imposes a constraint in the encoder

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 February 2021 doi:10.20944/preprints202102.0365.v1



network, which forces the bottleneck layer to follow a Gaussian distribution. Here the network minimizes the following loss function

$$L(x, g(f(x))) + L(l) \tag{5}$$

Where $L(l)$ is the latent loss, which is the Kullback-Leibler divergence of the bottleneck layer to a unit Gaussian distribution, which quantifies the difference between them. This assumption generates the latent variable with a generalization of the network.

### *2.3. Autoencoder Implementation:*

We used the Keras library with TensorFlow background to implement the four distinct autoencoders compared in this paper. The autoencoders were trained on a Quadro P4000 GPU with 8 Gb RAM. For subtyping and survival analysis, we applied the CancerSubtype R package [28].

For vanilla autoencoder, we used three hidden layers with 500, 100, 500 features, respectively, where input and output layers consist of 1000 features. These input features are selected based on the maximum variance of three data types. We used 500 features from gene expression data, 400 features for DNA methylation, and 100 features for miRNA expression. For denoising autoencoder, we used three hidden layers with 500, 100, and 500 units, respectively, where input and output layers consist of 1000 features. We applied a noise factor of 0.5 in the network input data. Sparse autoencoder was implemented with three hidden layers with 500, 100, and 500 units, respectively, where input and output layers consist of 1000 features. To avoid overfitting, we used an L1 regularization penalty of 0.000001, and L2 set as 0.00001, on the activities of the nodes. We implemented the variational autoencoder with four hidden layers with 1000, 500, 250 and 100 units. The decoder was implemented using the sequential model method and the encoder with the functional model. We used the log variance and lambda layer to convert the standard deviation when necessary for numerical stability.

An extension to the stochastic gradient descent (*adam*) algorithm was utilized to optimize all the autoencoder implementations. In vanilla, sparse, and denoising autoencoder, we used the hyperbolic tangent (*tanh*) activation function on the input layer and hidden layers and sigmoid on the output layer. In variational autoencoder, we applied a rectified linear activation function (*ReLU*) as the activation function between the layers and sigmoid in the output layer. We chose the mean square error function for vanilla and denoising autoencoder, binary cross-entropy function for sparse autoencoder, and negative log-likelihood function for variational autoencoder to measure the loss between the input layers ($X$) and the output layer ($X'$).

### *2.4. Clustering and subtyping*

The autoencoder transforms multidimensional features to a reduced number of features in the bottleneck layer. On this reduced feature set, we next applied the standard subtyping method to subtype patients. First, we calculated the similarity of each patient pair considering these reduced set of features. We used two different similarity measurements: euclidean distance and Spearman correlation, as a similarity measure between two patients. Next, we employed an unsupervised clustering algorithm to cluster similar groups of patients. In this work, we used an unsupervised subtypes discovery method combined with K-means [32] and Partitioning around medoids (PAM) [33] as our clustering methods. We run the two algorithms (K-means and PAM) in a window between 2 and 6 clusters.

### *2.5. Evaluation metrics for subtyping*

We utilized two different metrics to evaluate the performance of different autoencoders on the TCGA dataset. First, we performed a survival analysis to evaluate the survival patterns from different subtypes. Next, we calculated the p-value of the log-rank test to identify the difference of Kaplan-Meier survival curves between different subtypes. Here,

 doi:10.20944/preprints202102.0365.v1



a low p-value ($< 0.05$) ensure high confidence of different survival times for the different identified subtypes.

We also used the silhouette width of the clusters to benchmark the performance of Clustering. Silhouette scores quantify the inside group versus outside the group to measure how well a patient is matched to its identified cluster compared to other clusters. A high Silhouette value indicates a proper group distribution.

### *2.6. COX model for feature selection*

To validate the data fusion, we selected the two datasets (COAD and KRCC) that obtained the worst result with the selection by the variance, and we made a new selection of features based on the Cox proportional hazards model [34]. Cox proportional hazards model is a regression model to predict the relationship between the predictor variable and patients' survival. Using the univariate Cox model with cutoff $p < 0.05$, we selected 8,788 features from the methylation data, 400 feautures from DNA expression data, and 16 features from the miRNA data from COAD patients. From KRCC datasets, we selected 565 features from methylation data, 33 features from miRNA, and 419 from DNA expression data. Next, we fed these selected features as input for vanilla, sparse, denoising, and variational autoencoder implementations.

### *2.7. Comparison with other data integration methods*

We compare our results with other data fusion methods namely, Principal component analysis (PCA) and Similarity network fusion (SNF). SNF is a computational method on the fusion similarity network for aggregating multi-omics data as proposed in [8]. In this method, we used the methylation and mRNA from GBM datasets. Before applying SNF, we made a feature selection using the cox regression model. We selected 2,806 features from the methylation dataset and 3,309 from the mRNA dataset. The SNF algorithm and the survival analysis were implemented with clusters from 3 to 6 using the CancerSubtype package.

PCA allowed a linear dimensionality reduction to project the data in lower-dimensional spaces. We implemented PCA in Python using the sklearn package, and the features were selected based on the variance (0.90) in the GBM dataset. The final dataset was used as input to the Consensus Clustering for cancer subtype identification using the CancerSubtype package.

### *2.8. Differential expression and enrichment analysis on detected subtypes*

Lastly, we performed a differential expression (DE) and functional enrichment analysis of the clusters and compared the DE genes and enriched processes among the clusters. The DE genes were detected using the linear method LIMMA [35], while the functional enrichment analysis has been performed using the CluterProfiler [36] package in R. This can identify the critical genes belonging to a subtype and identify the functional processes which may lead to this outcome.

To explore the organization of the clusters, we performed a differential expression analysis using the GBM dataset.

For the analysis, the gene expression data were downloaded, for each one of the clusters obtained from the different types of autoencoders, and the clustering algorithms (PAM and Kmeans) were used from the HT_HG-U133A platform, using the GDCquery, GDCdownload and, GDCprepare functions. Samples with Primary Tumor and samples with solid tissue normal were compared to get differential expression utilizing the TCGA-analyze_DEA function with $fdr.cut = 0.01$ and $logFC.cut = 1$.

For the enrichment analysis of the gene sets, the TCGAanalyze_EAcomplet function was used, which allowed us to obtain the biological processes, cellular components, and molecular functions of Gene Ontology (GO) [37], in addition to the enrichment of the pathways.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 February 2021 doi:10.20944/preprints202102.0365.v1



## 3. Results and Discussion

### 3.1. Performance of Different autoencoders

We ran the survival analysis for 3 to 6 number of clusters for each autoencoder (Tables 1 and 2) . We noticed that the silhouette score differs depending upon the regularization methods . Hence, we chose the optimal cluster number for a disease based on counting the number of autoencoders that achieved a high silhouette score ($> 0.80$). Next, we performed a log-rank test to check if the identified clusters have different survival profile. Lowest p-values with high silhouette score ($> 0.8$) for the optimal cluster number was considered as the final cluster prediction. Though the performance of different autoencoders varies depending upon the dataset, we found that the sparse autoencoder performance was more impoverished compared to the other autoencoders in terms of log-rank test and mean silhouette score of the clusters (supplementary data 1).

**Table 1.** p-value of survival analysis results for the clusters generated with the autoencoder output.

| Dataset | Number of cluster | Autoencoder Vanilla | | Autoencoder Denoising | | Autoencoder Sparse | | Autoencoder Variational | |
|---|---|---|---|---|---|---|---|---|---|
| | | PAM/ Spearman | K-means/ Euclidean | PAM/ Spearman | K-means/ Euclidean | PAM/ Spearman | K-means/ Euclidean | PAM/ Spearman | K-means/ Euclidean |
| GBM | 3 | **0.002** | **0.001** | **9e-05** | **9e-04** | **0.02** | 0.543 | **5e-05** | **0.001** |
| | 4 | **0.002** | **2e-04** | 0.06 | **2e-05** | **0.0155** | **0.55** | **0.006** | **6e-05** |
| | 5 | **2e-04** | **1e-04** | **0.001** | **1e-05** | 0.114 | 0.219 | **5e-05** | **3e-05** |
| | 6 | **3e-04** | **2e-05** | **0.003** | **4e-05** | 0.267 | 0.865 | **1e-04** | **2e-05** |
| BIC | 3 | 0.0667 | 0.664 | 0.193 | 0.508 | 0.359 | 0.693 | 0.271 | 0.443 |
| | 4 | **0.0049** | 0.183 | 0.145 | **0.0275** | 0.531 | 0.58 | 0.0659 | 0.194 |
| | 5 | 0.322 | **0.0273** | **0.0481** | **0.0476** | **0.0066** | 0.713 | 0.103 | 0.219 |
| | 6 | 0.212 | 0.621 | **0.0306** | **0.0457** | **0.0083** | **4e-05** | 0.367 | 0.441 |
| COAD | 3 | **0.00524** | **0.00581** | **0.0275** | **0.00011** | **0.0478** | **0.0499** | **0.00871** | **0.0053** |
| | 4 | **0.0144** | **0.0135** | **0.044** | **0.0007** | **0.0855** | 0.0793 | 0.054 | **0.0181** |
| | 5 | **0.0309** | **0.031** | **0.0159** | **0.0041** | 0.097 | 0.082 | 0.0951 | **0.0006** |
| | 6 | **0.0241** | **0.0336** | **0.0341** | **0.00547** | 0.0897 | 0.143 | 0.0802 | **0.014** |
| KRCC | 3 | 0.288 | 0.392 | 0.165 | 0.135 | 0.37 | 0.346 | **0.00608** | **0.0266** |
| | 4 | 0.471 | 0.6144 | 0.437 | 0.47 | 0.663 | 0.663 | **0.0353** | **0.0393** |
| | 5 | 0.665 | 0.347 | 0.691 | **0.036** | 0.518 | 0.518 | 0.131 | **0.0141** |
| | 6 | 0.369 | 0.527 | 0.268 | 0.068 | 0.541 | 0.51 | 0.0669 | **0.0324** |

**Table 2.** Silhouette index results for the clusters generated with the autoencoder output.

| Dataset | Number of cluster | Autoencoder Vanilla | | Autoencoder Denoising | | Autoencoder Sparse | | Autoencoder Variational | |
|---|---|---|---|---|---|---|---|---|---|
| | | PAM/ Spearman | K-means/ Euclidean | PAM/ Spearman | K-means/ Euclidean | PAM/ Spearman | K-means/ Euclidean | PAM/ Spearman | K-means/ Euclidean |
| GBM | 3 | **1** | **0.91** | **0.98** | **0.91** | 0.72 | 0.22 | **0.98** | **0.87** |
| | 4 | **0.84** | 0.58 | 0.77 | 0.6 | 0.55 | 0.13 | **0.95** | 0.6 |
| | 5 | **0.8** | 0.62 | **0.82** | 0.73 | 0.64 | 0.25 | **0.88** | 0.51 |
| | 6 | 0.73 | 0.57 | 0.77 | 0.73 | 0.55 | 0.26 | **0.85** | 0.64 |
| BIC | 3 | **0.96** | **0.86** | 0.53 | 0.65 | 0.72 | **0.88** | **0.95** | **0.81** |
| | 4 | **0.91** | **0.87** | 0.67 | **0.81** | **0.84** | **0.81** | **0.85** | 0.78 |
| | 5 | 0.69 | 0.63 | 0.63 | 0.67 | 0.79 | 0.7 | 0.65 | 0.74 |
| | 6 | 0.67 | 0.74 | 0.61 | 0.6 | 0.71 | 0.65 | 0.59 | 0.74 |
| COAD | 3 | **0.97** | **0.82** | 0.7 | 0.67 | 0.66 | 0.46 | **0.83** | **0.82** |
| | 4 | 0.65 | 0.7 | 0.74 | 0.57 | 0.74 | 0.7 | 0.6 | 0.67 |
| | 5 | **0.8** | 0.68 | 0.72 | 0.59 | 0.73 | 0.68 | **0.96** | 0.73 |
| | 6 | **0.89** | 0.69 | 0.59 | 0.527 | 0.71 | 0.62 | 0.69 | 0.65 |
| KRCC | 3 | **0.83** | 0.77 | 0.58 | 0.48 | 0.69 | 0.66 | **0.95** | 0.63 |
| | 4 | 0.78 | **0.8** | 0.65 | 0.56 | **0.8** | **0.8** | **0.95** | 0.49 |
| | 5 | 0.55 | 0.67 | 0.59 | 0.46 | **0.81** | **0.8** | 0.78 | 0.58 |
| | 6 | 0.7 | 0.59 | 0.65 | 0.53 | 0.74 | 0.74 | 0.67 | 0.68 |

### 3.2. Performance of different autoencoders for GBM

GBM is the most studied cancer for subtype detection using multiview learning. However, a different number of subtypes has been detected by different computational methods on different datasets (Figures 3 and 4). Authors in [8] discovered three subtypes from 215 patients from TCGA using mRNA, miRNA, and DNA methylation data. While [38] classified GBM into the following four subtypes: a) Classical, b) Mesenchymal, c) Neural and d) Proneural. The authors in [39] also found three subtypes for the GBM dataset. Our method identified three as the optimal cluster number. Six autoencoders achieved a high silhouette score ($> 0.8$), while the variational autoencoder with PAM/Spearman achieved the lowest p-value in the log-rank test.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 February 2021 doi:10.20944/preprints202102.0365.v1



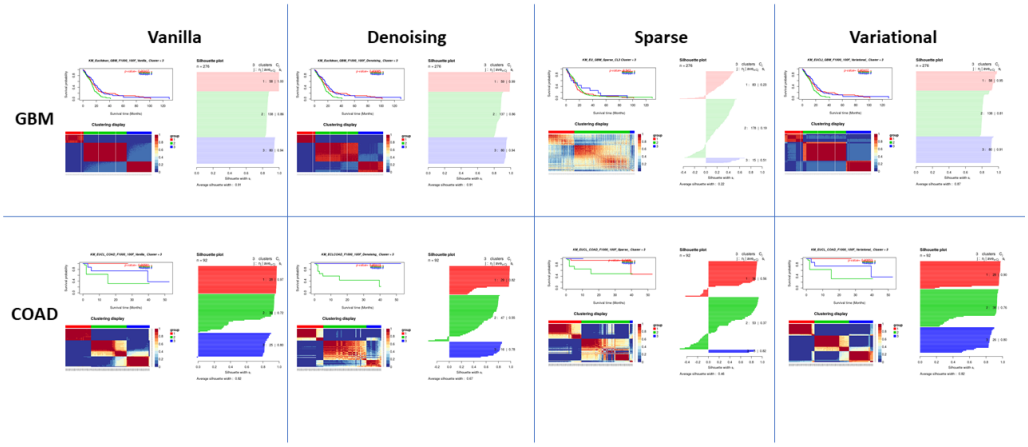


**Figure 3.** K-means survival analysis on Datasets. In each subfigure, left-top: Kaplan-Meier survival curves of three identified clusters. The log-rank test confirmed a difference in survival profiles among clusters; Left-dow: Patient to patient similarity and identified clusters on the dataset and Right: Silhouette score of the identified clusters.

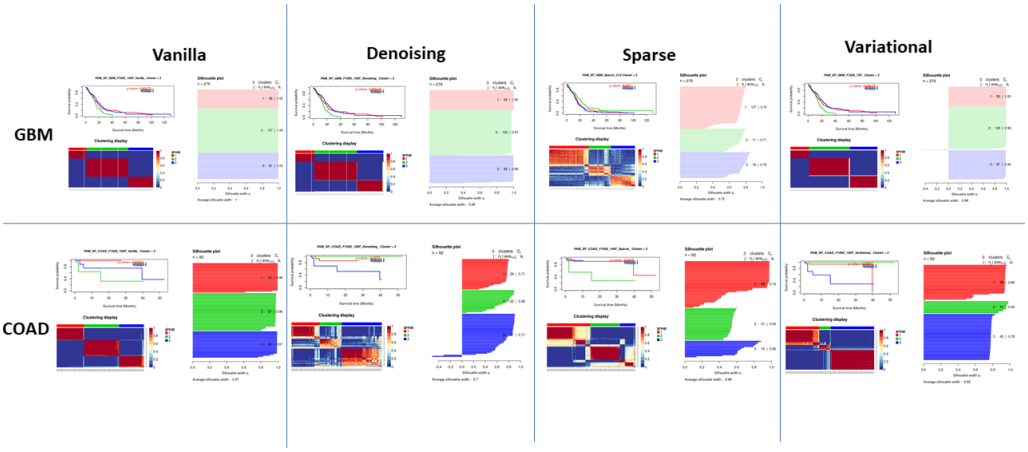


**Figure 4.** PAM survival analysis on Datasets. In each subfigure, left-top: Kaplan-Meier survival curves of three identified clusters; Left-dow: Patient to patient similarity and identified clusters on the dataset and Right: Silhouette score of the identified clusters.

### *3.3. Performance of different autoencoders for COAD*

For COAD, based on the count of silhouette score cutoff, we predicted the optimum number of clusters as three (Figures 3 and 4). Four different autoencoders constriction (Vanilla and variational autoencoders) achieved a high silhouette score for three clusters. The vanilla autoencoder with PAM/Spearman achieved the highest silhouette score of 0.96. We also observed a significant difference in survival profiles between these clusters $p = 0.005$. Moreover, all the other autoencoders also detected a difference in survival time for $K = 3$. It should be noted here that [8] also found three clusters in COAD based on the eigen distance.

### *3.4. Effect of different similarity measures*

Calculating patient to patient similarity measure is a crucial step in subtype detection. We can use various similarity measures for subtype detection, in which performance can vary depending on the dataset. Here we observed that PAM clustering with Spearman distance usually performed favorably than K-mean clustering with Euclidean distance. PAM with Spearman usually achieved better clustering based on the silhouette score. However, the identified clusters using K-Mean/Euclidean distance commonly showed lower p-value for the survival difference between the identified clusters.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 February 2021 doi:10.20944/preprints202102.0365.v1



### 3.5. Effect of supervised feature selection

For the KRCC and COAD datasets, we were unable to find a significant difference in survival profile between clusters. Hence, we chose a supervised feature selection algorithm Cox to select the input features. The Cox model is a supervised model, which selects the genes based on the survival status of patients. We observed a significant improvement of $p$ value for survival difference between the clusters using this method (Table 3). However, we noticed a decrease in silhouette score compared to the most variance (VAR) feature reduction method. Based on the silhouette score cutoff, variational autoencoder with Spearman distance performed best and optimal cluster number chosen was 3. It identified 3 different clusters with significant survival difference $p = 1e - 08$. Also for KRCC, the variational autoencoders achieved the highest silhouette score with three clusters. It also revealed a significant difference in survival profile between clusters.

**Table 3.** Result of autoencoder with data filtered by Cox Index

| Survival Analysis p-Value | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Dataset | Number of cluster | Autoencoder Vanilla | | Autoencoder Denoising | | Autoencoder Sparse | | Autoencoder Variational | |
| | | PAM/ Spearman | K-means/ Euclidean | PAM/ Spearman | K-means/ Euclidean | PAM/ Spearman | K-means/ Euclidean | PAM/ Spearman | K-means/ Euclidean |
| COAD | 3 | 0.0002 | 0.0027 | 0.0025 | 0.0025 | 0.005 | 0.005 | 0.0024 | 0.0027 |
| | 4 | 0.0081 | 0.0067 | 0.0076 | 0.0076 | 0.0162 | 0.0072 | 9.09e-05 | 0.012 |
| | 5 | 0.016 | 0.016 | 0.0097 | 0.0097 | 0.0253 | 0.0017 | 0.0032 | 0.026 |
| | 6 | 0.0323 | 0.0217 | 0.0205 | 0.015 | 0.048 | 0.0007 | 0.0082 | 0.051 |
| KRCC | 3 | 4e-09 | 7e-08 | 1e-08 | 8e-05 | 2e-06 | 1e-08 | 0.006 | 0.026 |
| | 4 | 5e-09 | 3e-07 | 9e-12 | 1e-06 | 9e-08 | 2e-10 | 0.035 | 0.039 |
| | 5 | 9e-11 | 3e-08 | 1e-10 | 2e-08 | 9e-08 | 1e-11 | 0.1 | 0.014 |
| | 6 | 3e-10 | 9e-07 | 1e-12 | 6e-08 | 3e-08 | 1e-10 | 0.67 | 0.0324 |
| Silhoutte Index Results | | | | | | | | | |
| Dataset | Number of cluster | Autoencoder Vanilla | | Autoencoder Denoising | | Autoencoder Sparse | | Autoencoder Variational | |
| | | PAM/ Spearman | K-means/ Euclidean | PAM/ Spearman | K-means/ Euclidean | PAM/ Spearman | K-means/ Euclidean | PAM/ Spearman | K-means/ Euclidean |
| COAD | 3 | 0.99 | 0.91 | 1 | 0.85 | 1 | 0.9 | 0.88 | 0.96 |
| | 4 | 0.95 | 0.76 | 0.98 | 0.76 | 0.98 | 0.76 | 0.85 | 0.78 |
| | 5 | 0.98 | 0.67 | 0.83 | 0.68 | 0.82 | 0.65 | 0.93 | 0.78 |
| | 6 | 0.87 | 0.63 | 0.87 | 0.6 | 0.77 | 0.63 | 0.81 | 0.6 |
| KRCC | 3 | 0.74 | 0.82 | 0.77 | 0.83 | 0.7 | 0.64 | 0.95 | 0.63 |
| | 4 | 0.68 | 0.74 | 0.69 | 0.8 | 0.63 | 0.62 | 0.95 | 0.49 |
| | 5 | 0.64 | 0.71 | 0.66 | 0.64 | 0.64 | 0.49 | 0.78 | 0.58 |
| | 6 | 0.54 | 0.62 | 0.75 | 0.6 | 0.64 | 0.46 | 0.066 | 0.68 |

### 3.6. Comparison with other subtype detection methods

Next, we compared the autoencoder subtype detection result with two other commonly used data fusion techniques: principal component analysis (PCA) and similarity network fusion (SNF) (Table 4). PCA is a commonly used method for dimensionality reduction. Unfortunately, PCA performed poorly for subtype detection. The clusters identified by PCA using Spearman correlation did not show a significant difference in survival time (Figure 5). However, the Euclidean distance performed better than the spearman correlation. SNF is another popular approach for data fusion. SNF showed comparable performance to autoencoders for subtype detection. However, SNF has a few additional hyperparameters, and the result is sensitive to hyperparameter selection.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 February 2021 doi:10.20944/preprints202102.0365.v1



**Table 4.** Principal Component Analysis (PCA) and Similarity Network Fusion (SNF) Results

| Principal Component Analysis Results | | | | | |
|---|---|---|---|---|---|
| Dataset | Number of cluster | PAM/ Spearman | | K-means/ Euclidean | |
| | | p-value | Silhouette Index | p-value | Silhouette Index |
| GBM | 3 | 0.542 | 0.56 | 0.0052 | -0.02 |
| | 4 | 0.514 | 0.42 | 0.0098 | 0 |
| | 5 | 0.989 | 0.35 | 0.0341 | -0.02 |
| | 6 | 0.731 | 0.38 | 0.131 | -0.04 |
| **Similarity Network Fusion Results** | | | | | |
| Dataset | Number of cluster | p-value | | Silhouette Index | |
| GBM | 3 | 2.43e-05 | | 0.46 | |
| | 4 | 0.0001 | | 0.47 | |
| | 5 | 3.39e-05 | | 0.47 | |
| | 6 | 1.92e-05 | | 0.46 | |

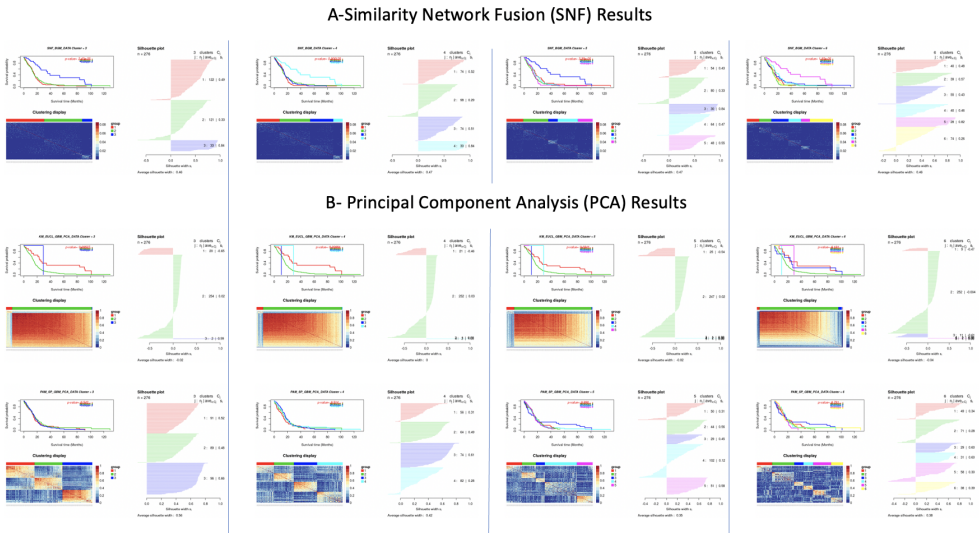


**Figure 5.** Similarity Network Fusion (SNF) and Principal component analysis results

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 17 February 2021 doi:10.20944/preprints202102.0365.v1



### 3.7. Differential expression and enrichment analysis on detected subtypes

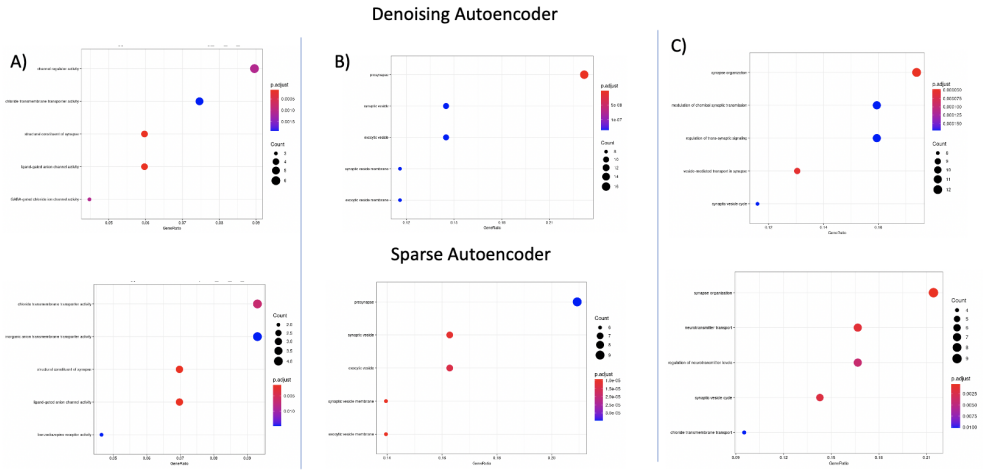


**Figure 6.** Gene ontology (GO) analysis was performed on the differentially expressed genes identified in the Denoising and Sparse autoencoders' results in patients cluster 1 using the K-means algorithm. The GO results were used to analyze the A) molecular functions, B) Cellular components, and C) Biological processes, identified by the Denoising Autoencoder and the Sparse Autoencoder data.

The GO and KEGG pathways' enrichment showed numerous eligible genes with differential expression between GBM and control samples considering the four autoencoders (vanilla, denoising, Sparse, variational) and their identified subgroups. The genes were related to cellular components, biological processes, and molecular function as shown in figure 6 (supplementary data), which is similar to previous studies [40,41]. Some selection criteria were applied to increase the reliability and precision of the results as follows: (i) p-value $< 0.05$, (ii) the reads count number $\geq 6$ (0 to 12), (iii) shared in the results from all autoencoders, and (iv) belong to at least two clusters.

First, we found that synaptic organization is the only one present among the three clusters (CL1, CL2, and CL3). According to the GO, the cell function called synaptic organization is a process that results in the assembly, an arrangement of constituent parts or disassembly of a synapse, the junction between a neuron and a target (neuron, muscle, or secretory cell).

For the immune synapse to occur, a conjugate of T cells and their targets are formed and triggers the reorganization of surface receptors. There is then an accumulation of actin at the contact site, forming the peripheral ring that delivers cytotoxic granules to the cytolytic synapse. The authors in [42] showed that impaired synaptic organization affects cell adhesion in T cells.

Secondly, we identified presynaptic and vesicle-mediated transport in cellular synapse components in at least two clusters of the three evaluated. These findings were similar to the study by [43] when analyzing targets of genes differentially expressed in GBM samples from in silico analysis using the Gene Expression Omnibus (GEO) database.

A pre-synaptic terminal in a synapse secretes neurotransmitters while the postsynaptic terminal feels neurotransmitters in its receptors [44]. This process is orchestrated by multiple and complex signaling pathways that differentiate the excitatory from the inhibitory pre-synapse; however, this process is still mostly unknown [45].

The authors in [46] identified that SYN1 (considered a pre-synaptic marker) is expressed outside neural tissues that can mimic neurotransmission. Furthermore, glutamate self-stimulation in malignant cells favors proliferation, motility, excitotoxic cell death, and seizures in peritumor brain tissues [47]. Therefore, pre-synaptic hyperexpression is unfavorable to a good prognosis.

Vesicles have been extensively investigated as a repository and transport of proteins, RNAs, and lipids between local and distant cells [48]. Vesicle-mediated intercellular com-

 doi:10.20944/preprints202102.0365.v1



munication, also known as surrounding tumor microenvironment (TME), composed of malignant, benign cells and non-cellular components, can interfere with gene expression by favoring a pro-tumorigenic microenvironment that modulates tumor behavior, aggressiveness, recurrence, and progression [49,50]. In GBM, the TME plays a crucial role in the progression of the GBM, with the vesicles being identified in the bidirectional communication between the tumor and the TME, in addition to favoring avoidance of apoptosis and therapeutic resistance [51], and also unfavorable to a good prognosis.

## 4. Conclusion

Recently, deep learning autoencoders are showing huge promise for multiview data fusion and cancer subtype detection. Here, we have compared four regularized autoencoders for subtype detection in four cancer types from the TCGA database. Though the performance of different autoencoders varied on the dataset, usually vanilla autoencoder and variational autoencoder showed the better performance to detect the subtypes. We also observed that the PAM/Spearman similarity showed better performance than K-mean/Euclidean clustering. By comparing the result of four autoencoders, we predicted the optimum number of subtypes in four cancer types. Moreover, DE analysis of the identified subtypes discovered critical genes and pathways in each subtype. Overall, we demonstrated that multi-omics data fusion combined with subtype detection as proposed here can improve cancer patient care.

**Author Contributions:** Study conceptualization, P.G., R.T.J.R, E.F.,and PR .; methodology, P.R and E.F.; software, E.F. V.C and P.R.; validation, E.F., V.C. P.R and A.M.P.C.R.; resources, V.A., R.T.J.R. and P.G.; data curation and prepossessing , E.F., V.C. and P.R.; writing–original draft preparation, E.F., P.R. and A.M.P.C.R.; writing–review and editing, E.F., V.A .,R.T.J.R.,P.R.,P.G. and A.M.P.C.R..; visualization,E.F. and P.R.; supervision, V.A.,R.T.J.R.,P.G.; funding acquisition, V.A, R.T.J.R, P.G

**Funding:** This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 88881.187658/2018-01.The funding sources had no role in the study design, data collection, data analyses, data interpretation, or writing of the report.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data and the codes are available online at https://github.com/edianfranklin/autoencoder_for_cancer_subtype .

**Acknowledgments:** The present study was conducted with support of the Brazilian Coordination for the Improvement of Higher Education Personnel (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES) and the Brazilian National Council for Scientific, Technological Development (Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq) and Pró-reitoria de Pesquisa e Pós-graduação(PROPESP) - UFPA. Biological Engineering Laboratory, Federal University of Pará (Universidade Federal do Pará – UFPA) and the Biological Networks Lab at Virginia Commonwealth University-VA.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; nor in the decision to publish the results.

**Sample Availability:** Samples of the compounds ... are available from the authors.

## Abbreviations

The following abbreviations are used in this manuscript:

 doi:10.20944/preprints202102.0365.v1



| | |
|---|---|
| TCGA | The Cancer Genome Atlas |
| SNF | Similarity network fusion |
| DL | Deep learning |
| GBM | Glioblastoma multiforme |
| COAD | Colon Adenocarcinoma |
| KRCC | Kidney renal clear cell carcinoma |
| BIC | Breast invasive carcinoma |
| VAR | Maximum variance |
| PCA | Principal Component Analysis |
| PAM | Partitioning around medoids |
| DE | Differential expression |
| GO | Gene Ontology |
| CL1 | Cluster 1 |
| GEO | Gene Expression Omnibus |
| TME | surrounding tumor microenvironmen |

## Appendix A

*Appendix A.1*

The appendix is an optional section that can contain details and data supplemental to the main text—for example, explanations of experimental details that would disrupt the flow of the main text but nonetheless remain crucial to understanding and reproducing the research shown; figures of replicates for experiments of which representative data are shown in the main text can be added here if brief, or as Supplementary Data. Mathematical proofs of results not central to the paper can be added as an appendix.

## References


1. Rana, P.; Berry, C.; Ghosh, P.; Fong, S.S. Recent advances on constraint-based models by integrating machine learning. *Current Opinion in Biotechnology* **2020**, *64*, 85–91.
2. Martini, P.; Chiogna, M.; Calura, E.; Romualdi, C. MOSClip: multi-omic and survival pathway analysis for the identification of survival associated gene and modules. *Nucleic acids research* **2019**, *47*, e80–e80.
3. Ramazzotti, D.; Lal, A.; Wang, B.; Batzoglou, S.; Sidow, A. Multi-omic tumor data reveal diversity of molecular mechanisms that correlate with survival. *Nature communications* **2018**, *9*, 1–14.
4. Ritchie, M.D.; Holzinger, E.R.; Li, R.; Pendergrass, S.A.; Kim, D. Methods of integrating data to uncover genotype–phenotype interactions. *Nature Reviews Genetics* **2015**, *16*, 85–97.
5. Chung, R.H.; Kang, C.Y. A multi-omics data simulator for complex disease studies and its application to evaluate multi-omics data analysis methods for disease classification. *GigaScience* **2019**, *8*, giz045.
6. Huang, S.; Chaudhary, K.; Garmire, L.X. More is better: recent progress in multi-omics data integration methods. *Frontiers in genetics* **2017**, *8*, 84.
7. Ebrahim, A.; Brunk, E.; Tan, J.; O'brien, E.J.; Kim, D.; Szubin, R.; Lerman, J.A.; Lechner, A.; Sastry, A.; Bordbar, A.; others. Multi-omic data integration enables discovery of hidden biological regularities. *Nature communications* **2016**, *7*, 13091.
8. Wang, B.; Mezlini, A.M.; Demir, F.; Fiume, M.; Tu, Z.; Brudno, M.; Haibe-Kains, B.; Goldenberg, A. Similarity network fusion for aggregating data types on a genomic scale. *Nature methods* **2014**, *11*, 333.
9. Chiu, Y.C.; Chen, H.I.H.; Zhang, T.; Zhang, S.; Gorthi, A.; Wang, L.J.; Huang, Y.; Chen, Y. Predicting drug response of tumors from integrated genomic profiles by deep neural networks. *BMC medical genomics* **2019**, *12*, 18.
10. Luck, M.; Sylvain, T.; Cardinal, H.; Lodi, A.; Bengio, Y. Deep learning for patient-specific kidney graft survival analysis. *arXiv preprint arXiv:1705.10245* **2017**.
11. Ng, A.; Ngiam, J.; Foo, C.Y.; Mai, Y.; Suen, C.; Coates, A.; Maas, A.; Hannun, A.; Huval, B.; Wang, T.; others. Stanford deep learning tutorial, 2015.
12. Marivate, V.N.; Nelwamodo, F.V.; Marwala, T. Autoencoder, principal component analysis and support vector regression for data imputation. *arXiv preprint arXiv:0709.2506* **2007**.
13. Mirza, B.; Wang, W.; Wang, J.; Choi, H.; Chung, N.C.; Ping, P. Machine learning and integrative analysis of biomedical big data. *Genes* **2019**, *10*, 87.
14. Zhang, Z.; Zhao, Y.; Liao, X.; Shi, W.; Li, K.; Zou, Q.; Peng, S. Deep learning in omics: a survey and guideline. *Briefings in functional genomics* **2018**, *18*, 41–57.
15. Wang, S.; Ding, Z.; Fu, Y. Feature selection guided auto-encoder. Thirty-First AAAI Conference on Artificial Intelligence, 2017.
16. Chaudhary, K.; Poirion, O.B.; Lu, L.; Garmire, L.X. Deep learning–based multi-omics integration robustly predicts survival in liver cancer. *Clinical Cancer Research* **2018**, *24*, 1248–1259.

17. Tan, J.; Ung, M.; Cheng, C.; Greene, C.S. Unsupervised feature construction and knowledge extraction from genome-wide assays of breast cancer with denoising autoencoders. Pacific Symposium on Biocomputing Co-Chairs. World Scientific, 2014, pp. 132–143.
18. Ronen, J.; Hayat, S.; Akalin, A. Evaluation of colorectal cancer subtypes and cell lines using deep learning. *Life Science Alliance* **2019**, *2*.
19. Zhang, X.; Zhang, J.; Sun, K.; Yang, X.; Dai, C.; Guo, Y. Integrated multi-omics analysis using variational autoencoders: application to pan-cancer classification. 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2019, pp. 765–769.
20. Simidjievski, N.; Bodnar, C.; Tariq, I.; Scherer, P.; Andres Terre, H.; Shams, Z.; Jamnik, M.; Liò, P. Variational autoencoders for cancer data integration: design principles and computational practice. *Frontiers in genetics* **2019**, *10*, 1205.
21. Sheet, S.; Ghosh, A.; Ghosh, R.; Chakrabarti, A. Identification of Cancer Mediating Biomarkers using Stacked Denoising Autoencoder Model-An Application on Human Lung Data. *Procedia Computer Science* **2020**, *167*, 686–695.
22. Makki, J. Diversity of breast carcinoma: histological subtypes and clinical relevance. *Clinical Medicine Insights: Pathology* **2015**, *8*, CPath–S31563.
23. Siegel, R.L.; Miller, K.D.; Jemal, A. Cancer statistics, 2016. *CA: a cancer journal for clinicians* **2016**, *66*, 7–30.
24. Society, A.C. Colorectal cancer facts & figures 2014–2016. *American Cancer Society* **2014**.
25. Acs, A. Cancer facts and figures 2010. *American Cancer Society, National Home Office, Atlanta* **2010**, pp. 1–44.
26. Chow, W.H.; Dong, L.M.; Devesa, S.S. Epidemiology and risk factors for kidney cancer. *Nature Reviews Urology* **2010**, *7*, 245.
27. Colaprico, A.; Silva, T.C.; Olsen, C.; Garofano, L.; Cava, C.; Garolini, D.; Sabedot, T.S.; Malta, T.M.; Pagnotta, S.M.; Castiglioni, I.; others. TCGAbiolinks: an R/Bioconductor package for integrative analysis of TCGA data. *Nucleic acids research* **2016**, *44*, e71–e71.
28. Xu, T.; Le, T.D.; Liu, L.; Su, N.; Wang, R.; Sun, B.; Colaprico, A.; Bontempi, G.; Li, J. CancerSubtypes: an R/Bioconductor package for molecular cancer subtype identification, validation and visualization. *Bioinformatics* **2017**, *33*, 3131–3133.
29. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *nature* **2015**, *521*, 436–444.
30. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* **2013**.
31. Doersch, C. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908* **2016**.
32. MacQueen, J.; others. Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Oakland, CA, USA, 1967, Vol. 1, pp. 281–297.
33. Kaufman, L.; Rousseeuw, P.J. Partitioning around medoids (program pam). *Finding groups in data: an introduction to cluster analysis* **1990**, *344*, 68–125.
34. Cox, D.R. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)* **1972**, *34*, 187–202.
35. Smyth, G.K. Limma: linear models for microarray data. In *Bioinformatics and computational biology solutions using R and Bioconductor*; Springer, 2005; pp. 397–420.
36. Yu, G.; Wang, L.G.; Han, Y.; He, Q.Y. clusterProfiler: an R package for comparing biological themes among gene clusters. *Omics: a journal of integrative biology* **2012**, *16*, 284–287.
37. Ashburner, M.; Ball, C.A.; Blake, J.A.; Botstein, D.; Butler, H.; Cherry, J.M.; Davis, A.P.; Dolinski, K.; Dwight, S.S.; Eppig, J.T.; others. Gene ontology: tool for the unification of biology. *Nature genetics* **2000**, *25*, 25–29.
38. Verhaak, R.G.; Hoadley, K.A.; Purdom, E.; Wang, V.; Qi, Y.; Wilkerson, M.D.; Miller, C.R.; Ding, L.; Golub, T.; Mesirov, J.P.; others. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer cell* **2010**, *17*, 98–110.
39. Wang, H.; Zheng, H.; Wang, J.; Wang, C.; Wu, F.X. Integrating omics data with a multiplex network-based approach for the identification of cancer subtypes. *IEEE transactions on nanobioscience* **2016**, *15*, 335–342.
40. Ji'an Yang, L.W.; Xu, Z.; Wu, L.; Liu, B.; Wang, J.; Tian, D.; Xiong, X.; Chen, Q. Integrated analysis to evaluate the prognostic value of signature mRNAs in glioblastoma multiforme. *Frontiers in genetics* **2020**, *11*.
41. Zhang, M.; Lv, X.; Jiang, Y.; Li, G.; Qiao, Q. Identification of aberrantly methylated differentially expressed genes in glioblastoma multiforme and their association with patient survival. *Experimental and therapeutic medicine* **2019**, *18*, 2140–2152.
42. Zhao, F.; Cannons, J.L.; Dutta, M.; Griffiths, G.M.; Schwartzberg, P.L. Positive and negative signaling through SLAM receptors regulate synapse organization and thresholds of cytolysis. *Immunity* **2012**, *36*, 1003–1016.
43. Xiong, D.D.; Xu, W.Q.; He, R.Q.; Dang, Y.W.; Chen, G.; Luo, D.Z. In silico analysis identified miRNA-based therapeutic agents against glioblastoma multiforme. *Oncology reports* **2019**, *41*, 2194–2208.
44. Südhof, T.C. Towards an understanding of synapse formation. *Neuron* **2018**, *100*, 276–293.
45. Dabrowski, A.; Terauchi, A.; Strong, C.; Umemori, H. Distinct sets of FGF receptors sculpt excitatory and inhibitory synaptogenesis. *Development* **2015**, *142*, 1818–1830.
46. Yool, A.J.; Ramesh, S.A. Molecular targets for combined therapeutic strategies to limit glioblastoma cell migration and invasion. *Frontiers in pharmacology* **2020**, *11*, 358.
47. Corsi, L.; Mescola, A.; Alessandrini, A. Glutamate receptors and glioblastoma multiforme: an old "Route" for new perspectives. *International journal of molecular sciences* **2019**, *20*, 1796.
48. Graner, M.W. Roles of extracellular vesicles in high-grade gliomas: tiny particles with outsized influence. *Annual review of genomics and human genetics* **2019**, *20*, 331–357.

 doi:10.20944/preprints202102.0365.v1




49. Van der Pol, E.; Böing, A.N.; Harrison, P.; Sturk, A.; Nieuwland, R. Classification, functions, and clinical relevance of extracellular vesicles. *Pharmacological reviews* **2012**, *64*, 676–705.
50. Yáñez-Mó, M.; Siljander, P.R.M.; Andreu, Z.; Bedina Zavec, A.; Borràs, F.E.; Buzas, E.I.; Buzas, K.; Casal, E.; Cappello, F.; Carvalho, J.; others. Biological properties of extracellular vesicles and their physiological functions. *Journal of extracellular vesicles* **2015**, *4*, 27066.
51. Simon, T.; Jackson, E.; Giamas, G. Breaking through the glioblastoma micro-environment via extracellular vesicles. *Oncogene* **2020**, *39*, 4477–4490.