BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

A pan-cancer somatic mutation embedding using autoencoders

Martin Palazzo[1,2,3], Pierre Beauseroy[2] and Patricio Yankilevich[1*]

Abstract

Background: Next generation sequencing instruments are providing new opportunities for comprehensive analyses of cancer genomes. The increasing availability of tumor data allows to research the complexity of cancer disease with machine learning methods. The large available repositories of high dimensional tumor samples characterised with germline and somatic mutation data requires advance computational modelling for data interpretation. In this work, we propose to analyze this complex data with neural network learning, a methodology that made impressive advances in image and natural language processing.

Results: Here we present a tumor mutation profile analysis pipeline based on an autoencoder model, which is used to discover better representations of lower dimensionality from large somatic mutation data of 40 different tumor types and subtypes. Kernel learning with hierarchical cluster analysis are used to assess the quality of the learned somatic mutation embedding, on which support vector machine models are used to accurately classify tumor subtypes.

Conclusions: The learned latent space maps the original samples in a much lower dimension while keeping the biological signals from the original tumor samples. This pipeline and the resulting embedding allows an easier exploration of the heterogeneity within and across tumor types and to perform an accurate classification of tumor samples in the pan-cancer somatic mutation landscape.

Keywords: Autoencoder, Kernel learning, Cancer genomics

Background

Recent years have been characterized by the availability of data repositories providing access to large-scale collaborative cancer projects [1, 2]. These databases contain data from thousands of tumor samples from patients all over the world labeled by tumor type, subtype and other clinical factors such as age and prognosis. The available tumor data includes different layers of biological signals acquired by state-of-the-art omics technologies (e.g., genomics, transcriptomics, proteomics, metabolomics, etc). The information includes somatic mutations, copy number somatic mutations, gene expression, DNA methylation among other data types. Each layer represents the signature of the tumor represented by different macromolecules. Another characteristic is that each omic layer is characterized by tens of thousands of features like gene mutations [3] or gene expression. From a mathematical point of view tumors can be represented as vectors in a high dimensional space. This can be a problem in learning tasks known as the curse of dimensionality. This work focuses on the understanding of the available genomics data containing the somatic point mutations identified in each tumor sample. The availability of a large quantity of samples from the main tumor types and subtypes invites the study of current relations between different tumors and the development of learning algorithms that reduce the complexity of the initial high dimensional environment. The tumor samples are labeled by medical doctors and pathologists based on the tumor primary site and histology. The exploration of tumor mutational profiles can reveal communities of tumors and hidden relations between tumor types and subtypes [4]. This work aims to address the complexity of the pan-cancer somatic mutational data and learn a lower dimension of tumor representations based on the tumor mutational profiles.

*Correspondence: pyankilevich@ibioba-mpsp-conicet.gov.ar
[1] Instituto de Investigación en Biomedicina de Buenos Aires (IBioBA)—CONICET—Partner Institute of the Max Planck Society, Godoy Cruz 2390, C1425FQD Buenos Aires, Argentina
Full list of author information is available at the end of the article

At the same time of the significant growth in cancer biological data, the machine learning and deep learning communities have been developing learning methods such as Artificial Neural Networks with impressive results on image, signal and natural language processing [5]. One type of neural network model is the Auto-encoder (AE) [6]. AE are embeddings built to find reduced and simpler representations of complex data using un-supervised feedforward networks, therefore a non-linear reduction of dimensionality. Different types of Autoencoders have been proposed to generate a reduced latent space with a representative distribution of the original data using different regularization processes like Sparse [7] or contractive autoencoders [8].

The objective of this work is to learn a latent space of reduced dimensionality with autoencoders using mutational data from 14 types of tumors available from the International Cancer Genome Consortium (ICGC) [1] repository. This will allow understanding the similarities between tumors of different types and an improved classification performance of subtypes based on their mutational profile and their corresponding projection in a low dimensional latent space. The Kernel Target Alignment (KTA) score [9] and hierarchical clustering are proposed to measure the quality of the latent space. KTA is computed to measure the similarity between two kernel functions, one learned from the samples projected in the latent space and the second from an ideal target kernel. Kernel functions also lead to the measurement of similarities between training and test samples of the same class once the autoencoder model is trained and observe if the latent space maps similarly independent samples.

Related work

Autoencoders have been used on a wide range of applications in cancer informatics. One application is its use on a single cancer type, such as liver cancer, while combining multi-omics data [10] to learn a latent space and identify new cancer subtypes. A similar case has been proposed for breast cancer to discover subtypes using transcriptomics data [11]. A newer version of AE, the Variational Auto-encoder, has been used to learn a latent space to improve the classification of known subtypes of lung cancer using DNA methylation data [12]. Moreover, instead of learning a latent space from a single type of cancer, a pan-cancer study based on transcriptomics data from The Cancer Genome Atlas (TCGA) [2] using Variational Auto-encoders evidenced a big potential for the use of autoencoders to learn reduced latent space while keeping biological insights [13]. Another work with gene expression data from TCGA applied standard autoencoders and Gene Supersets, which are a priori defined gene sets that retain biological signals in the latent space [14]. On the other hand, a network and graph theory analysis has been done for pan-cancer mutational data to detect communities of tumors [15] and find hidden relations between them using the co-occurrence of mutations as connections. A recent work maps mutated genes instead of the tumor samples to a lower dimension using deep learning techniques to learn a distributed representation [16]. By reviewing the bibliography, it is clear that data from different omics layers require models to simplify the original context and reflect emerging patterns. Autoencoders have shown great adaptability to biological data and are extremely useful for reducing dimensionality.

Our work proposes to learn a latent space from somatic mutations of large pan-cancer data using Autoencoders. This embedding is based in a model that projects tumor somatic mutation profiles in a low dimensional latent space where biological signals like tumor subtype persist and facilitates the comparison of tumor samples. For instance, this latent space can be used to explore mutational profiles when the primary tumor is unknown and there is no information on the tumor type, or to better classify tumor subtypes. From our best knowledge up to now, there are no attempts of reproducing a latent space using autoencoders from tumor somatic mutation data. Another important contribution of our work is an accurate tumor classification approach based on one-class Support Vector Machines (SVM) for each of the 40 tumor subtypes.

Results

In this work a neural network maps tumors characterized by mutational profiles from a high dimensional space, built from somatic mutated genes, to a low dimensional space using an Autoencoder as a nonlinear function. The mutational input data, which is highly sparse is considered as multi-modal since it is divided between deleterious and non-deleterious based on the variant type (see Fig. 1). The input tumor mutational profiles are transformed into a latent space as dense vectors.

By training a regularized autoencoder the tumors characterized with 12424 mutated gene as features are mapped to a final latent space of 50 dimensions. Thus, a global compression ratio of 248 is obtained. The learned latent space not only preserves the structural relationship between tumor subtypes but also improves the separability of classes making much easier the identification of a specific tumor phenotype. The resulting Autoencoder architecture has a multi-modal approach with one Encoder-Decoder function for deleterious and non-deleterious input mutations respectively. This allows weighting both types of input vectors (see “Methods” section). Then both models are merged at their respective latent layer level into a single global latent layer known as Latent Space. For regularization Batch Normalization

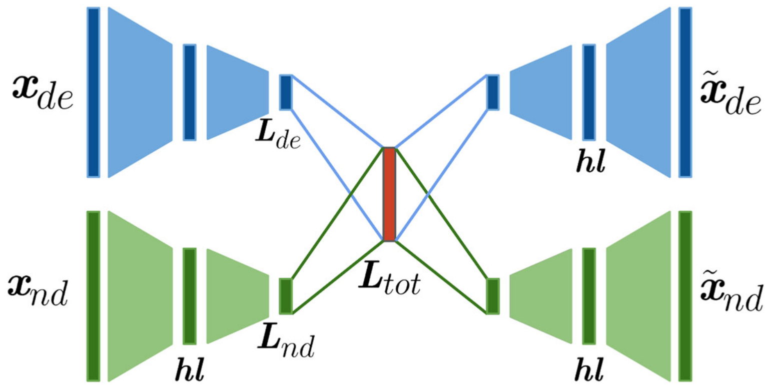

Fig. 1 Model architecture. Scheme of the multi-modal autoencoder architecture for both deleterious and non-deleterious mutational profiles. Input and output dimension have 12424 genes. The encoder and decoder functions contain one hidden layer each of 400 activation functions (neurons). The latent layer of each autoencoder has 50 activation functions. Highlighted in red is the latent space L_{tot} which contains signal from both types of mutational profiles

is implemented after the Encoding hidden layer. Also, L2 norm is imposed to all the encoding weights to regularize their activity and penalize large weights. The learning rate and the L2 norm have been tuned by 5-fold cross validation using the validation loss computed as binary cross entropy. In the resulting latent space 40 one-class SVM models are trained, one for each tumor subtype. Then all the models are evaluated with independent tumor samples from Test set showing promising classification results. Figure 2 shows a scatter plot of a t-distributed stochastic neighbor embedding (t-SNE) as a projection of the resulting latent space after model training and evaluation [17].

Quality assessment of latent space

The validation of the latent space must consider a set of quality assessments. In this work three different approaches are proposed. The first one is the reconstruction error of the autoencoder. Figure 3 shows the convergence of both the Training and Validation loss up to 45 epochs after 5 fold cross validation. This convergence means that the reconstruction quality stabilize. It

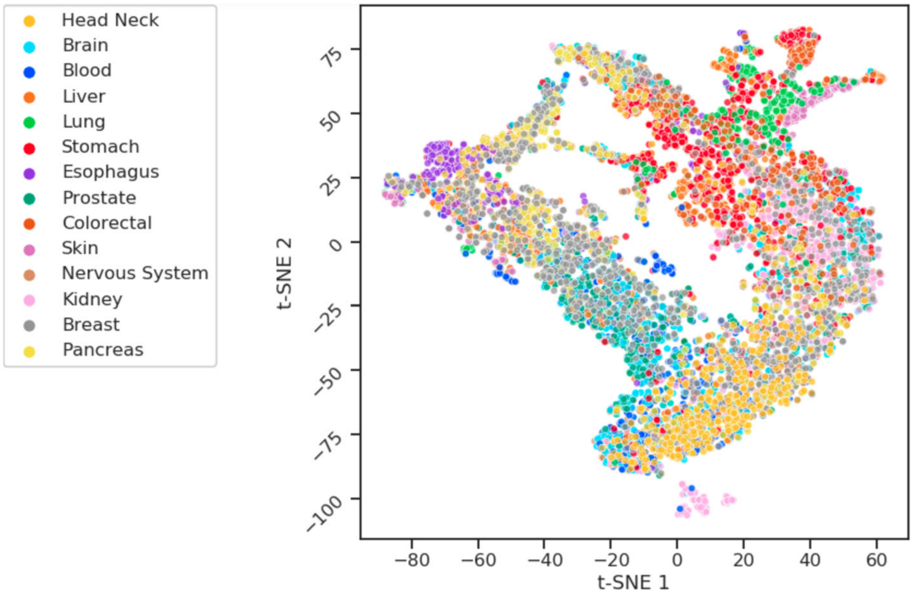

Fig. 2 t-SNE scatter plot. Scatter plot of the projection of the latent space using t-SNE dimensions showing by different colors the 14 tumor types by primary site

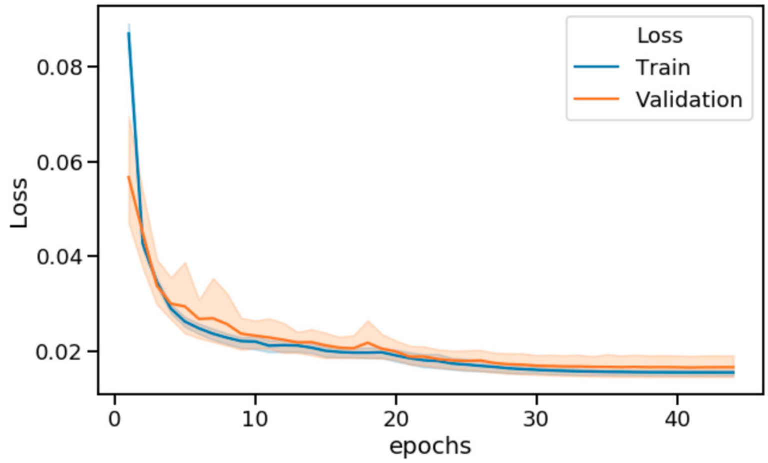

Fig. 3 Validation loss. Autoencoder training and validation loss during training epochs after cross validation

serves as a way to measure how information is preserved from the input to the latent space until the output of the autoencoder. If the autoencoder loss is small means the reconstruction $\hat{x}$ is similar to the input x then the compressed latent vector preserves the salient features of the input space.

A second approach to assess the quality of the latent space is via Kernel Target Alignment. The KTA measures the kernel performance in finding high similarity between tumors of the same type, and low similarity between tumors of different types. The higher the KTA, the better the similarity between tumors of the same type. Given a gaussian kernel built in the latent space K_l, a second kernel in the original input space K_{in}, and the tumor type labels Y_{ty}, the resulting alignment KTA_l obtained in the latent space outperforms the KTA_{in} obtained from the initial input space. The obtained results show that the autoencoder keeps the original input properties in the latent space while cleaning the noise, making a better space for pattern recognition tasks.

Finally, the latent space is evaluated by cluster analysis. By performing hierarchical clustering in the input and in latent space separately, is possible to asses the quality of the resulting clusters by measuring how well tumors of the same type are clustered together. This is done by computing the mutual information score MI. This score consider the probability of a set of samples belonging to a class to be clustered together given a number of k clusters. As expected, the MI scores are better in the latent space when compared to the original input space. Figure 4 shows the results of KTA evaluation for different values of sigma parameter and the MI scores for different number of clusters. In order to evaluate different architectures of the autoencoder, other dimensions L of the latent space were evaluated, $L = 100$ and $L = 200$. As the assessment for different values of L leads to similar results, the $L = 50$ is used in the final architecture since it has associated less model parameters or weights to fit during training and means a simpler model.

Tumor sub-type classification

One-class SVMs is used to test if the low dimensional latent space learned captures mutational signals from the original input space and improves the tumor sub-types classification. A one-class SVM classifier is built for each of the 40 tumor subtypes. Every one-class classifier is trained with the samples corresponding to its subtype label and validated with the rest of the training samples. Finally, the one-class model is tested with an independent test set of samples preserved for autoencoder evaluation. The area under the ROC curve (AUC-ROC) is computed using the test samples to assess how well the classifier detects the True Positive samples, which in this case means tumors of the same subtype, therefore a correct classification. The same classification approach is also applied on the input space in both deleterious and non deleterious mutational data as a method to benchmark the obtained results. Table 1 shows the classification performance measured by the area under the ROC curve for each class on the latent space and the two input spaces X_{de} and X_{nd}. Results equal to 0.5 indicates that the classifier can not find any difference between one subtype and the rest of the samples. On the other hand, results close to 1 correspond to classifiers that separates well the corresponding subtype class from the rest of the samples. The classification performance presents an improvement in the latent space in 35 out of 40 tumor subtypes (highlighted in the Table 1). In all these cases the area under the curve is higher in the classifier trained on the latent space than the ones trained on the input space with

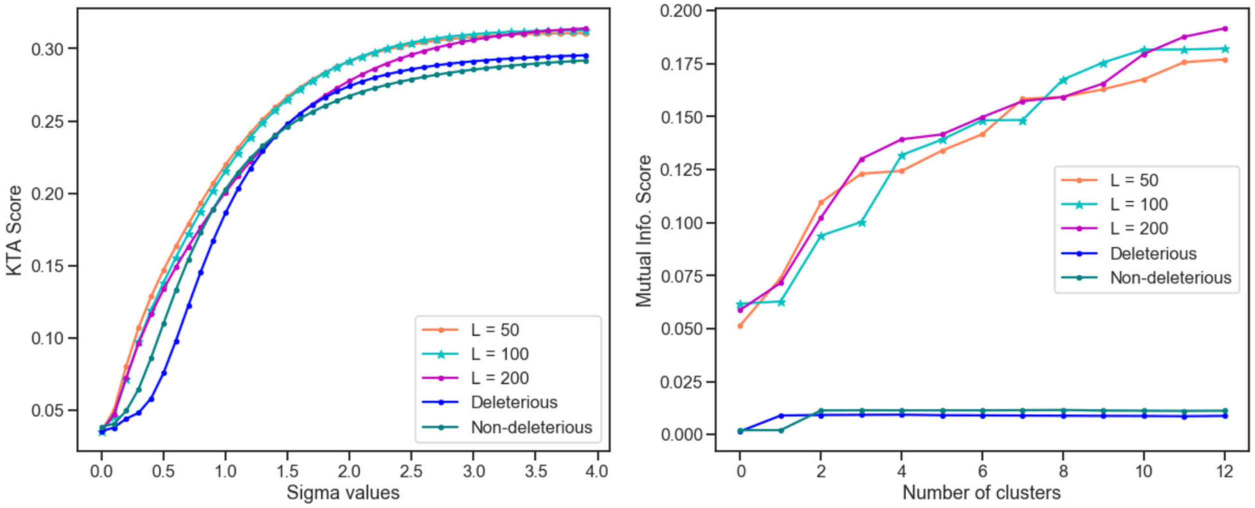

Fig. 4 Latent Space evaluation. Left: Kernel target alignment score for different values of sigma parameter. Right: Mutual Information score for different number of clusters

deleterious and non-deleterious mutational profiles. The tumor subtypes LAML-KR, KIRC-US, KIRP-US, LUSC-US, ESAD-UK, LIRI-JP and PRAD-CA show promising results with AUC-ROC close to 1 while the performance on the input space is close to 0.5.

Discussion

Simple Somatic mutation data can be used to train an Autoencoder and build a latent space of lower dimensionality that keeps the biological signals of tumors. This study carries out a pan-cancer mapping by an Autoencoder trained with 8946 tumor samples from 40 tumor subtypes and evaluated with 2236 independent samples. The deleterious and non-deleterious variants in tumor mutational profiles are merged by a multi-modal autoencoder architecture allowing the weighting of each variant type differently. Although most pipelines for identification of disease-causing mutations filter out non-deleterious variants at the earliest stages, there is growing evidence that this type of variants affect protein splicing, expression and ultimately function, and some of these contribute to disease. This is not the case for tumor mutational profile exploration and classification, where non-deleterious variants showed to be very informative [4]. In this work deleterious and non-deleterious variant profiles equally contribute to the final latent space, with a mixture weight of $\alpha = 0.5$. It has been shown that Kernel Target Alignment and hierarchical clustering analysis exhibits an improvement on the latent space over these two input spaces regarding the capacity to group samples in clusters. Finally, a classification task using one-class approach is implemented in the latent space for each tumor subtype. The evaluation of the classifiers using independent samples for each class shows improvements in the vast majority of the tumor subtypes.

Conclusions

This work presents a tumor mutation profile analysis pipeline which is from our best knowledge, the first attempt to learn a low dimensional latent space using autoencoders from mutational profiles of large pan-cancer tumor data. The latent space evidences biological signals in 50 dimensions after reducing the input dimension more than 200 times for 40 tumor subtypes. The use of kernel learning for latent space validation and assessment of the resulting cluster structures proved to be a useful approach. The use of a multi-modal approach to differentiate deleterious and non-deleterious variants let the autoencoder to learn a more realistic representation of the tumor somatic profiles. Classification at a tumor subtype level in the learned representation shows a clear improvement in comparison to the original input space. The quality of learned embedding has been assessed by different methods and proved to be a powerful tool for analysis of the pan-cancer mutational landscape.

This work is a first milestone and lay the foundations for future work on the learned somatic mutation autoencoder embedding to serve as a reference for biomarker discovery using feature selection techniques based on kernel learning and neural networks. If enough clinical data is available, the identified biomarkers with a further exploration of the latent space for cancer subtypes stratification could inform of patient expected prognosis and what are the most beneficial treatments. Future work should also consider further model validation and the inclusion of multi-omic input layers.

Table 1 Classification results for 40 cancer subtypes

Primary site	Project name	Test samples	AUC latent	AUC De.	AUC Nd.
Head and neck	**ORCA-IN**	36	0.75	0.50	0.55
Brain	**LGG-US**	56	0.81	0.62	0.80
Blood	CLLE-ES	102	0.83	0.84	0.76
Head and neck	**THCA-SA**	28	0.82	0.66	0.81
Liver	**LINC-JP**	79	0.54	0.50	0.50
Lung	**LUSC-KR**	34	0.77	0.57	0.50
Skin	**GACA-CN**	24	0.79	0.50	0.50
Stomach	**LICA-FR**	49	0.67	0.50	0.51
Blood	**ESCA-CN**	65	0.77	0.5	0.61
Lung	**EOPC-DE**	40	0.81	0.74	0.55
Colorectal	**COCA-CN**	63	0.60	0.50	0.51
Skin	**SKCM-US**	67	0.84	0.50	0.50
Liver	**LICA-CN**	79	0.61	0.50	0.50
Blood	ALL-US	15	0.83	0.91	0.94
Skin	**SKCA-BR**	20	0.84	0.50	0.50
Brain	**GBM-US**	55	0.87	0.50	0.66
Nervous system	NBL-US	18	0.96	0.95	0.96
Blood	**LAML-KR**	41	0.97	0.73	0.64
Prostate	**PRAD-UK**	28	0.65	0.5	0.50
Prostate	**PRAD-US**	51	0.83	0.58	0.74
Blood	**MALY-DE**	49	0.82	0.50	0.50
Kidney	**KIRC-US**	82	0.90	0.50	0.58
Brain	**PBCA-DE**	90	0.79	0.77	0.59
Kidney	**RECA-EU**	49	0.84	0.5	0.5
Blood	AML-US	18	0.94	0.96	0.96
Breast	**BRCA-UK**	28	0.59	0.50	0.50
Kidney	**KIRP-US**	33	0.90	0.50	0.57
Prostate	**PRAD-CA**	58	0.88	0.71	0.50
Stomach	**STAD-US**	58	0.84	0.50	0.50
Stomach	**GACA-JP**	115	0.80	0.50	0.50
Liver	**LIRI-JP**	32	0.88	0.50	0.75
Breast	**BRCA-US**	189	0.75	0.60	0.50
Lung	**LUSC-US**	39	0.91	0.50	0.50
Esophagous	**ESAD-UK**	61	0.90	0.50	0.50
Colorectal	**COAD-US**	51	0.83	0.50	0.50
Breast	**BRCA-FR**	15	0.82	0.50	0.56
Pancreas	**PACA-AU**	73	0.72	0.60	0.50
Pancreas	PACA-CA	54	0.87	0.50	0.89
Head & Neck	THCA-US	76	0.85	0.85	0.50
Breast	**BRCA-EU**	114	0.79	0.56	0.50

The number of the test samples for the corresponding class is detailed. Area under the Roc curve is detailed for classifiers on Latent Space, Deleterious and Non-Deleterious input data. Tumor subtypes where the classification performance is improved in the latent space are highlighted in bold

Methods

Pan-cancer somatic mutation data

Data has been downloaded from the International Cancer Genome Consortium [1]. Only Simple Somatic Mutation (SSM) data of the Release 27 has been considered for this work. Tumor data is labeled by type and subtype. There are 14 tumor types composed by 40 subtypes. There are a total of 11183 samples from whole exome sequecing and each one is characterized by more than 20.000 protein coding genes annotated with Variant Effect Predictor tool [18].

The pre-processing step consists of first counting separately the number of deleterious and non-deleterious somatic mutations per patient protein coding gene according to Cosmic notation [19]. Deleterious mutations are Inframe, Frameshift, Missense, Start Lost, Stop Gained and Stop Lost and the Non-deleterious are Synonimous, Splice, UTR 5 prime and UTR 3 prime. This results in two data matrices X_{mn} for each mutation type where m is the number of samples and n the number of genes or features. The value of each position X_{ij} corresponds to the number of somatic mutations a sample i has in gene j. Then each gene is zero-one normalized. It is important to remark the high sparsity of the data matrix and the curse of dimensionality. Initially the sample to feature ratio is 0.55. From this initial context only the features with non-zero values in at least 50 samples are retained and the rest that are only present in less than 50 samples are discarded. This decreases the feature set to a total of 12.424 genes and the resulting sample-to-feature ratio is 1.1 now. The data matrix is partitioned in train and test sets where train samples represent the 80% of the total data set. Within train set data where split in 5 folds to perform 5 training and validation iterations to tune the hyper-parameters of the model like learning rate, hidden layer size, regularization parameters, the number of training epochs and the mixture weight parameter.

Autoencoders

Autoencoders are feedforward networks that learn two functions simultaneously: an encoder and decoder. The encoder maps the original input domain $\mathcal{X}$ to a new domain named latent space $\mathcal{Z}$ of dimension L. The decoder then maps from $\mathcal{Z}$ to the original input space $\mathcal{X}$. The mapping from $\mathcal{X}$ to $\mathcal{Z}$ is created by a neural network with one or multiple hidden layers [20]. The output of the decoder is also a reconstruction feedforward network. Since we aim to have a lower dimensionality at the latent space $\mathcal{Z}$, the autoencoder is forced to build an encoder function that captures all the salient features from the training data as much as possible [21]. The encoder and decoder functions are defined as $\boldsymbol{z} = f(\boldsymbol{x})$ and $\tilde{\boldsymbol{x}} = g(\boldsymbol{z})$ respectively where $\boldsymbol{z}$ are the samples at the learned latent space and $\tilde{\boldsymbol{x}}$ are the reconstructed samples on $\mathcal{X}$. With the previous definition, the autoencoder loss function to minimize is formalized as

$$E\left(\boldsymbol{x}, \tilde{\boldsymbol{x}}\right) = E\left(\boldsymbol{x}, g\left(f\left(\boldsymbol{x}\right)\right)\right)$$

where E penalizes $g\left(f\left(\boldsymbol{x}\right)\right)$ to be different to $\boldsymbol{x}$. In this work the measure of this loss function is the cross entropy score. Then the encoder F and decoder G functions can be defined as [22]

$$\begin{aligned} \boldsymbol{z} &= F\left(\boldsymbol{x}, \mathbf{W}_F\right) = \sigma\left(\mathbf{W}_F \boldsymbol{x} + \mathbf{b}_F\right) \\ \tilde{\boldsymbol{x}} &= G\left(\boldsymbol{z}, \mathbf{W}_G\right) = \sigma\left(\mathbf{W}_G \mathbf{z} + \mathbf{b}_G\right) \end{aligned}$$

where $F\left(\cdot, \mathbf{W}_F\right)$ and $G\left(\cdot, \mathbf{W}_G\right)$ correspond to the encoding and decoding functions respectively and $\sigma\left(\cdot\right)$ is an activation function. The original input sample is $\mathbf{x} \in \mathcal{X}$, $\tilde{\boldsymbol{x}} \in \mathcal{X}$ is the reconstructed samples and $\boldsymbol{z}$ the corresponding latent ones which dimension is lower than $\boldsymbol{x}$. The tensors $\mathbf{W}$ and $\mathbf{b}$ corresponds to the trained weights and biases of the encoder and decoder networks. These parameters are learned by backpropagation in order to minimize the loss function by the optimizer. This work uses Adaptive Moment Estimation (Adam) [23] optimizer to learn the weights of the network that minimizes the loss function. Adam is a novel first-order stochastic optimization technique. It computes an adaptive learning rate depending on the gradient mean.

Training an autoencoder to solely make $\tilde{\boldsymbol{x}}$ a copy of $\boldsymbol{x}$ does not ensure the learned latent space $\mathbf{z}$ is representative of the input $\boldsymbol{x}$. Without any constrain or penalization term, the encoder and decoder functions can result into a function that only copies the input in an output, but that is not useful to learn a latent space. For that reason different regularization strategies are evaluated which are L2 norm and Batch Normalization. L2 norm consists in a constraint term added to the loss function E where β is the regularization parameter.

$$E'\left(\boldsymbol{x}, g\left(f\left(\boldsymbol{x}\right)\right), \beta\left(f\left(\boldsymbol{x}\right)\right)\right) = E\left(\boldsymbol{x}, g\left(f\left(\boldsymbol{x}\right)\right)\right) + \beta \sum_i \left|w_i\right|_2^2$$

The regularization term penalizes the functions f and g to have large weights leading to a simpler model and reducing overfitting [24]. To improve even more the generalization capacity the other regularization policy is used during the encoding process just after the first hidden layer of the encoding function. Batch Normalization [25] consists in auto-scaling the activation units to zero mean and unit variance at each mini-batch iteration.

Since the input data is characterized by two mutational data types and is represented in two matrices X_{de} and X_{nd} corresponding to deleterious and non-deleterious mutations respectively, the Autoencoder model must have two inputs and two outputs. Then, a multi-modal approach is proposed on the autoencoder architecture [26, 27]. A multi-modal autoencoder consists of two input networks and two output networks, each one with one Encoder and Decoder function. The network layers L_{de} and L_{nd} correspond to the latent representation of each model and are merged into one L_{tot} after the encoding function. This latent representation, which includes the signals of the two models, it is decomposed in two decoding functions. Figure 1 shows the proposed architecture. Since two models are participating in the construction of the final latent space, the final loss function is determined as follows

$$E_{tot} = \alpha E_{de} + (1-\alpha)E_{nd}$$

where α is a mixture weight parameter that represents the contribution of each model in the final latent representation, E_{de} is the loss of the deleterious model and E_{nd} is the non-deleterious. This approach allows to implement a weighting system on the input data and gives relative importance to deleterious and non deleterious mutational data. The best value of the α mixture weight parameter was found by a grid search of values $\alpha =[0.1, 0.3, 0.5, 0.7, 0.9]$, using Mutual Information (MI) from clustering results to evaluate the performance. During the cross validation task for each weight configuration a latent space is obtained, and based on the 14 tumor type classes a hierarchical clustering model with 14 clusters is implemented. For each clustering result the mutual information is calculated between the obtained cluster labels and the ground truth tumor labels. The final α parameter corresponds to the highest MI score obtained after cross validation which is $\alpha = 0.5$ (Additional file 1: Figure S1).

For the experiments, the architecture used consists in one hidden layer of 400 neurons (activation units) in both the encoding and decoding functions, named as hl_e and hl_d respectively. The latent space is obtained from the latent layer $\boldsymbol{z}$ with dimensionality L and represents an information bottleneck with the lowest dimension within the network. Different dimensions of latent space are evaluated to observe how the structure of the tumor data changes and is retained for each latent dimensionality. The pipeline's objective is to reveal biological structures of the input data while reducing the dimensionality as much as possible. Figure 1 shows the proposed multi-modal architecture of the auto-encoder trained with both deleterious and non deleterious somatic mutational data from tumors.

During training the L2 norm and learning rate have been selected by 5-fold cross-validation on the train set using L_2 =[0.00005, 0.00002, 0.00001] and L_r = [0.005, 0.0025, 0.001]. The final values are $L_2 = 0.00002$ and $L_r = 0.001$. The number of epochs and the learning rate have been determined by an early stopping policy when the validation loss changes to lower than a certain threshold between each epoch.

Kernel learning

In this work Kernel Learning is used to measure the structure of the learned latent space by the autoencoder and as the function used for the support vector classification step. Kernel functions can be thought as similarity functions between vectors. These functions indicate the dot product between those vectors mapped in a high dimensional Hilbert feature space. A Kernel is a function $k : \mathcal{X} \times \mathcal{X} \mapsto R$ where $\mathcal{X} \subseteq R^n$ is an n-dimensional space $\mathcal{X}$. The function k is symmetric and describes implicitly the mapping ϕ from $\mathcal{X}$ to a Reproducing Kernel Hilbert Space $\mathcal{H}$ by an inner product [28] $K\left(x_i, x_j\right) = \left\langle \phi(x_i), \phi(x_j) \right\rangle_{\mathcal{H}}$. The mapping from $\mathcal{X}$ to a feature space $\mathcal{H}$ is done by the function $\phi : X \mapsto \phi(X) \in \mathcal{H}$.

In this work, a good kernel finds high similarity between tumors of the same type and low similarity between tumors of different types. The kernel used is the Gaussian Kernel where the σ parameter functions as an exponential scaling factor.

$$k(x_i, x_j) = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right); \sigma > 0 \qquad (1)$$

The Gaussian kernel is one of the most common kernel functions. The parameter σ controls the size of the neighborhood of any x_i such that $k(x_i, x)$ is significantly larger than zero. The bigger the σ parameter, the more constant the function and thus the lower its ability to learn non-trivial patterns. On the other hand, low values of σ allow the kernel to fit complex patterns and be more sensitive to details [29].

Once the kernel is defined, it can be compared with other kernels via the Kernel Alignment [9]. Given two valid kernels K_1 and K_2 over a set of samples M, the alignment A between both kernels is defined as

$$A\left(K_1, K_2\right) = \frac{\langle K_1, K_2\rangle_F}{\sqrt{\langle K_1, K_1\rangle_F \langle K_2, K_2\rangle_F}} \qquad (2)$$

and means the similarity between the two kernels using the same sample set M where $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product between both kernel matrices. In other words, it can be thought as how similar both kernels map the samples. Considering the set S of labeled samples such that $S = \left\{(x_1, y_1) ..., (x_m, y_m)\right\}$ where $x_i \in R^n$ and $y_i \in \{-1, +1\}$, when $K_2 = yy^T$ represents an ideal Kernel matrix or target K_{yy} with each position $K_{ij} = 1$ if $y_i = y_j$ and $K_{ij} = -1$ if $y_i \neq y_j$. The alignment of a kernel K and the target K_{yy} is known as the Kernel Target Alignment. The higher the KTA score, the bigger the inter-class distance, therefore the classes are more separated between each other and thus, well mapped to their corresponding target label. The σ parameter of the Gaussian Kernel has been tuned to maximize the corresponding KTA. In this work KTA is used to assess the quality of the latent space by using the tumor type labels and to evaluate the improvement of it in comparison with the original input space. It is expected to observe a higher KTA in the latent space and a lower one in the input high dimensional space.

Once the autoencoder is trained and the latent dimensions are finally defined, Kernel Alignment and support vector classification are used for latent space evaluation. It is important to remark that since there are 40 tumor subtypes, a one vs all approach using a binary classification is not the best option since classes are highly unbalanced. For this reason classification is done by the one class ν-SVM model [30]. It is used to classify each tumor sub-type against the rest of the tumor samples and is commonly used to define a decision boundary of only one class versus the rest of the sample set. This approach is applied to each tumor subtype and serves as a way to perform multi-class classification, where a one-class model is used instead of using a binary classifier. Its objective function is

$$\min_{w,\xi,\rho} \frac{1}{2}\|w\|^2 + \frac{1}{n\nu}\sum_{i=1}^{n}\xi_i - \rho$$
$$\text{s.t.}(w \cdot \phi(x_i)) \geq \rho - \xi_i, \xi_i \geq 0$$

The hyperparameter $\nu \in (0, 1)$ functions as a lower bound on the number of samples characterized as support vectors and an upper one for the miss-classified samples that lie on the wrong side of the hyperplane. A set of slack variables $\xi = (\xi_1, ..., \xi_m)$ are introduced to allow the possibility of miss classifications when a sample fall on the wrong side of the margin. Then the decision function is defined as follows

$$f(x) = \boldsymbol{sgn}((w \cdot \phi(x_i)) - \rho)$$

Note that a Kernel function can shape the decision function by the participation of the $\phi(\cdot)$. The $f(x)$ function will be positive for most of the samples in the training set in a small region which are going to be samples of the same tumor subtype, and -1 elsewhere. The closer ν parameter to 0 the penalization of miss-classified samples increases. If the training samples are separable from the origin, then the ν-SVM model will find a unique hyperplane that separates all the samples from the origin, and the distance from it to the origin is the maximal.

Once the classifiers are trained on the latent space these are evaluated with independent test samples. Classification performance is reported on Table 1.

Cluster analysis

Latent space quality assessment is done also by cluster analysis as a complement of the KTA. Once the autoencoder network is trained, tumors are mapped from the original high dimensional space to a latent space $\mathcal{Z}$ with lower dimensionality. Given a latent space of dimension L and the original input space X_{de} and X_{nd}, Hierarchical Clustering with a k number of clusters is applied separately to samples in $\mathcal{Z}$ on one side and to X_{de} and X_{nd} on the other. Clusters labels c_i are assigned to each sample i belonging to cluster c. Then by considering the real tumor types labels Y_{ty} as the ground truth, a mutual information score [31] is computed for each value of k to evaluate the quality of the obtained clusters in both cases. Every time the clustering algorithm is executed a set of k cluster labels $\Lambda = \{\lambda_1, ..., \lambda_k\}$ and a set of ground truth labels $Y = \{y_1, ..., y_j\}$ are defined. Then the mutual information score is defined as follows

$$\text{MI}(\Lambda, C) = \sum_k \sum_j P\left(\lambda_k \cap y_j\right) \log \frac{P\left(\lambda_k \cap y_j\right)}{P(\lambda_k)P(y_j)}$$

where $P(\lambda_k)$ is the probability of a sample to be located in cluster λ_k, $P(y_j)$ the probability to belong to class y_j and $P(\lambda_k \cap y_j)$ the one to be at the intersection of both. The possible results of the score are $\text{MI} \in (0, 1)$. The higher the MI score the better the quality of the obtained cluster result. In this work, the MI score is computed for different values of clusters k in both the obtained latent space $\mathcal{Z}$ and the original input space X_{in}. If the quality of clusters is better in the latent space than the input space then MI_z will be higher than MI_{de} and MI_{nd}. A higher value of MI can be interpreted as samples of the same tumor type tend to be grouped together in the same cluster.

Computational tools

Data preprocessing, clustering analysis and kernel learning have been implemented with Python 3.6 and Scikit Learn [32]. The autoencoder model has been built with Keras an Tensorflow backend. Training has been performed with a GPU N-Vidia GTX 1060 4GB.

Supplementary information

Supplementary information accompanies this paper at https://doi.org/10.1186/s12859-019-3298-z.

Additional file 1: Supplementary Figure S1.

Abbreviations

AE: Autoencoders; AUC-ROC: Area under the ROC curve; ICGC: International cancer genome consortium; KTA: Kernel target alignment; MI: Mutual information; SSM: Simple somatic mutation; SVM: Support vector machines; TCGA: The cancer genome atlas; t-SNE: t-distributed stochastic neighbor embedding

Acknowledgements

The authors would like to thank the International Cancer Genome Consortium who have provided the data.

Authors' contributions

MP, PY and PB designed research; MP performed research and analyzed data; and MP, PB and PY wrote the paper. All authors reviewed the final version of the manuscript.

Funding
This work was supported by the doctoral program of the Universidad Tecnologica Nacional in Argentina, the UTN FRBA doctoral school of Signal Processing and the Universite de Technologie de Troyes. Also it was funded by grants from CONICET, ANPCyT and FOCEM-Mercosur.

Availability of data and materials
The data that support the findings of this study are available from the International Cancer Genome Consortium (ICGC) but restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly available. Data are however available from the authors upon reasonable request and with permission of ICGC.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Instituto de Investigación en Biomedicina de Buenos Aires (IBioBA)—CONICET—Partner Institute of the Max Planck Society, Godoy Cruz 2390, C1425FQD Buenos Aires, Argentina. [2]Institut Charles Delaunay, Universite de Technologie de Troyes, 12 Rue Marie Curie, 10300 Troyes, France. [3]Universidad Tecnologica Nacional, Facultad Regional Buenos Aires, Av. Medrano 951, C1179AAQ Buenos Aires, Argentina.

Received: 8 May 2019 Accepted: 27 November 2019
Published online: 11 December 2019

References
1. Consortium ICG, et al. International network of cancer genome projects. Nature. 2010;464(7291):993.
2. Weinstein JN, Collisson EA, Mills GB, Shaw KRM, Ozenberger BA, Ellrott K, Shmulevich I, Sander C, Stuart JM, Network CGAR, et al. The cancer genome atlas pan-cancer analysis project. Nat Genet. 2013;45(10):1113.
3. Alexandrov LB, Nik-Zainal S, Wedge DC, Aparicio SA, Behjati S, Biankin AV, Bignell GR, Bolli N, Borg A, Børresen-Dale A-L, et al. Signatures of mutational processes in human cancer. Nature. 2013;500(7463):415.
4. Goncearenco A, Rager SL, Li M, Sang Q-X, Rogozin IB, Panchenko AR. Exploring background mutational processes to decipher cancer genetic heterogeneity. Nucleic Acids Res. 2017;45(W1):514–22.
5. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436.
6. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science. 2006;313(5786):504–7.
7. Xu J, Xiang L, Liu Q, Gilmore H, Wu J, Tang J, Madabhushi A. Stacked sparse autoencoder (ssae) for nuclei detection on breast cancer histopathology images. IEEE Trans Med Imaging. 2016;35(1):119–30.
8. Rifai S, Mesnil G, Vincent P, Muller X, Bengio Y, Dauphin Y, Glorot X. Higher order contractive auto-encoder. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer; 2011. p. 645–60. https://doi.org/10.1007/978-3-642-23783-6_41.
9. Cristianini N, Shawe-Taylor J, Elisseeff A, Kandola JS. On kernel-target alignment. In: Advances in Neural Information Processing Systems. Cambridge: MIT Press; 2002. p. 367–73.
10. Chaudhary K, Poirion OB, Lu L, Garmire LX. Deep learning–based multi-omics integration robustly predicts survival in liver cancer. Clin Cancer Res. 2018;24(6):1248–59.
11. Guo Y, Shang X, Li Z. Identification of cancer subtypes by integrating multiple types of transcriptomics data with deep learning in breast cancer. Neurocomputing. 2019;324:20–30.
12. Wang Z, Wang Y. Exploring dna methylation data of lung cancer samples with variational autoencoders. In: 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE; 2018. p. 1286–9. https://doi.org/10.1109/bibm.2018.8621365.
13. Way GP, Greene CS. Evaluating deep variational autoencoders trained on pan-cancer gene expression. 2017. arXiv preprint arXiv:1711.04828.
14. Chen H-IH, Chiu Y-C, Zhang T, Zhang S, Huang Y, Chen Y. Gsae: an autoencoder with embedded gene-set nodes for genomics functional characterization. BMC Syst Biol. 2018;12(8):142.
15. Iranzo J, Martincorena I, Koonin EV. Cancer-mutation network and the number and specificity of driver mutations. Proc Natl Acad Sci. 2018;115(26):6010–9.
16. Kim S, Lee H, Kim K, Kang J. Mut2vec: distributed representation of cancerous mutations. BMC Med Genom. 2018;11(2):33.
17. Maaten Lvd, Hinton G. Visualizing data using t-sne. J Mach Learn Res. 2008;9(Nov):2579–605.
18. McLaren W, Gil L, Hunt SE, Riat HS, Ritchie GR, Thormann A, Flicek P, Cunningham F. The ensembl variant effect predictor. Genome Biol. 2016;17(1):122.
19. Forbes SA, Bindal N, Bamford S, Cole C, Kok CY, Beare D, Jia M, Shepherd R, Leung K, Menzies A, et al. Cosmic: mining complete cancer genomes in the catalogue of somatic mutations in cancer. Nucleic Acids Res. 2010;39(suppl_1):945–50.
20. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P-A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res. 2010;11(Dec):3371–408.
21. Goodfellow I, Bengio Y, Courville A. Deep Learning: MIT press; 2016.
22. Kampffmeyer M, Løkse S, Bianchi FM, Jenssen R, Livi L. Deep kernelized autoencoders. In: Scandinavian Conference on Image Analysis. Springer; 2017. p. 419–30. https://doi.org/10.1007/978-3-319-59126-1_35.
23. Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014. arXiv preprint arXiv:1412.6980.
24. Vapnik V. Estimation of Dependences Based on Empirical Data: Springer; 2006.
25. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015. arXiv preprint arXiv:1502.03167.
26. Ngiam J, Khosla A, Kim M, Nam J, Lee H, Ng AY. Multimodal deep learning. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11); 2011. p. 689–96.
27. Ma M, Sun C, Chen X. Deep coupling autoencoder for fault diagnosis with multimodal sensory data. IEEE Trans Ind Inf. 2018;14(3):1137–45.
28. Friedman J, Hastie T, Tibshirani R. The Elements of Statistical Learning, vol. 1: Springer; 2001.
29. Zhong S, Chen D, Xu Q, Chen T. Optimizing the gaussian kernel function with the formulated kernel target alignment criterion for two-class pattern classification. Pattern Recogn. 2013;46(7):2045–54.
30. Schölkopf B, Williamson RC, Smola AJ, Shawe-Taylor J, Platt JC. Support vector method for novelty detection. In: Advances in Neural Information Processing Systems. Cambridge: MIT Press; 2000. p. 582–8.
31. Vinh NX, Epps J, Bailey J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. J Mach Learn Res. 2010;11(Oct):2837–54.
32. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al. Scikit-learn: Machine learning in python. J Mach Learn Res. 2011;12(Oct):2825–30.

Publisher's Note