# Multi-view Omics Translation with Multiplex Graph Neural Networks

Costa Georgantas
costa.georgantas@chuv.ch
Lausanne University Hospital and University of Lausanne
Lausanne, Switzerland

Jonas Richiardi
jonas.richiardi@chuv.ch
Lausanne University Hospital and University of Lausanne
Lausanne, Switzerland

## ABSTRACT

The rapid development of high-throughput experimental technologies for biological sampling has made the collection of omics data (e.g., genomics, epigenomics, transcriptomics and metabolomics) possible at a small cost. While multi-view approaches to omics data have a long history, omics-to-omics translation is a relatively new strand of research with useful applications such as recovering missing or censored data and finding new correlations between samples. As the relations between omics can be non-linear and exhibit long-range dependencies between parts of the genome, deep neural networks can be an effective tool. Graph neural networks have been applied successfully in many different areas of research, especially in problems where annotated data is sparse, and have recently been extended to the heterogeneous graph case, allowing for the modelling of multiple kinds of similarities and entities. Here, we propose a meso-scale approach to construct multiplex graphs from multi-omics data, which can construct several graphs per omics and cross-omics graphs. We also propose a neural network architecture for omics-to-omics translation from these multiplex graphs, featuring a graph neural network encoder, coupled with an attention layer. We evaluate the approach on the open The Cancer Genome Atlas dataset (N=3023), showing that for MicroRNA expression prediction our approach has lower prediction error than regularized linear regression or modern generative adversarial networks.

## CCS CONCEPTS

• **Computing methodologies** → *Neural networks*; • **Applied computing** → *Life and medical sciences.*

## KEYWORDS

graph representation, heterogeneous graph, machine learning, autoencoder, microRNA, gene expression, methylation

**ACM Reference Format:**
Costa Georgantas and Jonas Richiardi. 2022. Multi-view Omics Translation with Multiplex Graph Neural Networks. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3487553.3524714


*WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France.*

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524714

## 1 INTRODUCTION

In biological analysis, the general term *omics* refers to data gathered from various sources such as genomics, transcriptomics, metabolomics, and others. Omics modelling has a vast range of applications such as disease subtyping, survival analysis and individualised risk prediction. High-throughput biomedical technologies have made possible the collection of large datasets combining various types of omics data. As these data are generally high-dimensional, sometimes incomplete, and exhibit non-linear relationships, it is of particular interest to be able to analyse them jointly using expressive models. Recent deep learning methods for omics data have shown to be effective in many areas of research[6, 10, 34]. As efforts are made to make these types of methods more interpretable, one can assume that their adoption will continue to grow in this field.

Network representations are particularly well suited to this type of data, as they offer a flexible and rich mathematical structure for representing similarities between objects in a non-Euclidean domain, and have seen ever-increasing adoptions in life sciences [19]. With the recent advances in graph representation and deep learning, graph convolutional networks (GCNs) [15] have shown to be effective for a wide variety of tasks in both supervised and unsupervised problems [29]. While these methods were generally initially designed for social networks, they have also been applied to a variety of other data sets [4, 14, 32]. More recently, GCNs have been extended to the heterogeneous and multiplex cases [8, 11], where graphs can have multiple node and edge types. The number of deep graph neural representation methods is growing at a rapid pace and new self-supervised algorithms such as contrastive learning [21] and consensus representation learning [17] can potentially be applied in many areas of research for widely different fields.

In some applications, generating the graph is trivial, for instance two users in a social platform can be connected with a edge if they are friends. However many data sets, especially in biology, come in the form of tabular data; what should be considered a similarity between two nodes is not as well defined. Nevertheless, representing similarities between samples has shown to be an effective strategy for different tasks such as clustering [30], classification [33] and subtyping [2]; more generally, correlation networks representing sample-to-sample similarities are a staple of data analysis in life sciences, including gene co-expression [36] or brain networks [24].

Similarities between two samples can be defined in many ways. State of the art methods for omics analysis [31, 33] generally compute one similarity per omics, such as a measure of the inverse distance between two samples. However we can argue that this choice might be too reductive given the large dimensions of these samples, and that multiple similarities can be computed for each modality. These design decisions will ultimately affect the output

of the network, and thus need to be studied in more detail. To the best of our knowledge, no work has been done to provide a general framework for multiplex graph generation from omics data.

Our main contributions are summarized as follows:

- We propose a method for constructing a multiplex graph from multi-omics data by grouping correlated features and computing an edge type per group, providing a "meso-scale" approach which is more fine-grained than using a single graph per omics.
- We introduce a novel graph encoding architecture able to generate meaningful latent representations of the samples with minimal annotated data.
- We compare our method with the state of the art for omics-to-omics translation and show the effectiveness of our method in both low and well annotated regimes.
- We examine the impact of the multiplex graph construction step, comparing with a well-established method for multi-view, similarity-based omics analysis.

## 2 RELATED WORK

In the multi-omics literature, most similarity-based methods regard the generation of the sample-sample similarity graph as preprocessing [23], and use a single similarity graph per omics, with the notable exception of rMKL-LPP [26], which can use several different kernels per omic. Going back to spectral clustering [5], several results have shown that graph construction impacts clustering [18], so it is of interest to examine the impact of graph construction on predictive algoritms.

In the next subsections, we highlight the graph construction stage of a well established and a more recent approach for multi-omics clustering and classification. We also mention other methods that construct graphs out of tabular data.

### 2.1 Similarity Network Fusion

The Similarity Network Fusion [31] (SNF) method consists in generating graphs from multiple modalities and fusioning them iteratively. Given $\mathbf{x}_i$ a feature vector for node $i$, let $\rho(\mathbf{x}_i, \mathbf{x}_j)$ denote the Euclidean distance between nodes $\mathbf{x}_i$ and $\mathbf{x}_j$. For the graph generation process, initial edges weights between nodes $i$ and $j$ are given by :

$$\mathbf{W}_{ij} = \exp\left(-\frac{\rho^2\left(\mathbf{x}_i, \mathbf{x}_j\right)}{\mu\varepsilon_{i,j}}\right) \tag{1}$$

where $\mu$ is a hyperparameter that can be empirically set and $\varepsilon_{i,j}$ is defined as

$$\varepsilon_{i,j} = \frac{\text{mean}\left(\rho\left(\mathbf{x}_i, \mathcal{N}_i\right)\right) + \text{mean}\left(\rho\left(\mathbf{x}_j, \mathcal{N}_j\right)\right) + \rho\left(\mathbf{x}_i, \mathbf{x}_j\right)}{3} \tag{2}$$

where $\mathcal{N}_i$ represent the $N$ closest neighbors of node $i$, Finally, the adjacency matrix entries are given by

$$\mathbf{A}_{ij} = \begin{cases} \dfrac{\mathbf{W}_{ij}}{\Sigma_{k\in N_i}\mathbf{W}_{ik}}, j \in N_i \\ 0, \text{otherwise} \end{cases} \tag{3}$$

This matrix is computed for each modality and used in conjunction with another similarity matrix. This method assumes that local similarities given by the K nearest neighbors are the most reliable. Note that a single graph is computed for each modality.

### 2.2 MOGONET

MOGONET [33] is a method based on graph convolutional networks and a view correlation discovery network. It also builds a single graph for each modality of measurement, but does so based on a similarity metric.

$$\mathbf{A}_{ij} = \begin{cases} s\left(\mathbf{x}_i, \mathbf{x}_j\right), & \text{if } i \neq j \text{ and } s\left(\mathbf{x}_i, \mathbf{x}_j\right) \geq \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $s$ is the cosine similarity

$$s\left(\mathbf{x}_i, \mathbf{x}_j\right) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\|_2 \left\|\mathbf{x}_j\right\|_2} \tag{5}$$

and $\epsilon$ is chosen such that

$$k = \sum_{i,j} I\left(s\left(\mathbf{x}_i, \mathbf{x}_j\right) \geq \epsilon\right)/n \tag{6}$$

where $I(\cdot)$ is the indicator function and $n$ is the number of nodes, and the hyperparameter $k$ dictates the sparsity of the graph.

Again, only one graph is computed for each modality. We argue that as the modalities contain a large amount of features, it could be valuable to model more than one similarity per omic between the nodes.

### 2.3 Other Methods

To the best of our knowledge, TabGNN [7] is the only method using a graph neural network that builds a multiplex graph out of general tabular data. The graph construction step is however largely left to the user. Other methods build single graphs from k nearest neighbors [25, 35], or Pearson correlation [2] and de Resende et al. [3] compute an approximation of the Shapeley values to model relationships. In general, these methods have one or multiple hyper-parameters that dictate the resulting graph properties, such as its sparsity. The common assumption is that appropriate links between the samples will provide meaningful information for the downstream task. These approaches all make sense intuitively, however for high-dimensional data it is worth considering whether similarities between samples should be considered uni-dimensional. Although building one graph per modality is a first step, it is itself an arbitrary choice and other less obvious choices might yield better results.

## 3 A MULTIPLEX GRAPH NEURAL NETWORK FOR MULTI-VIEW OMICS TRANSLATION

### 3.1 Notation

Let $\mathbf{F} \in \mathbb{R}^{N\times d_f}$ be a feature matrix where N is the number of samples and $d_f$ is the dimension of the features, and let $\mathbf{A} \in \mathbb{R}^{N\times N}$ denote an adjacency matrix representing similarities between the samples.

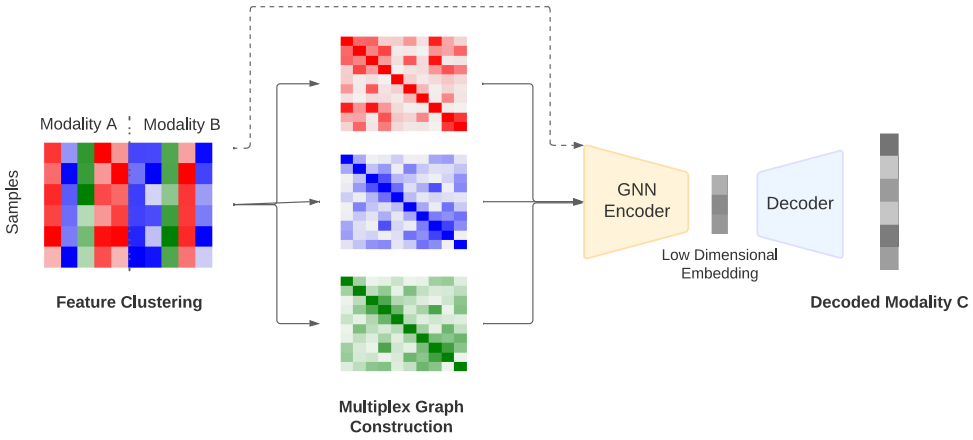


**Figure 1: Overview of the proposed method. The adjacency matrices of the multiplex graph, computed for each feature cluster (represented in colors) are fed to a variational graph neural encoder, and decoded with a regular fully connected network.**

An attributed graph $\mathcal{G}(\mathbf{A}, \mathbf{F})$ is composed of these two components. An attributed multiplex graph $\mathcal{G}_{\mathcal{M}} = \{\mathcal{G}^1, \cdots, \mathcal{G}^R\}$ is composed of $R \geq 1$ layers where $\mathcal{G}^r(\mathbf{A}^r, \mathbf{F})$ are individual attributed graphs.

## 3.2 Feature Clustering

As computing one layer of the multiplex graph per feature would be computationally intensive and would result in many redundant graphs, we instead choose to cluster correlated features together and compute a single layer per cluster. As a result, in a multi-modal setting features from different modalities can be in the same cluster. Our motivation for this approach so is that multiplex graphs neural networks can effectively make use of the appropriate similarities (or graph layer) for a given task and discard the others. Formally, we define a number of clusters $K$ that will also represent the number of layers in the multiplex graph. Following Ward's method [20], pairs of clusters that minimally increase within-cluster variance are merged recursively. This procedure is repeated until the number of clusters is equal to $K$. The feature clustering is computed using scikit-learn's [22] FeatureAgglomeration method.

This approach has multiple benefits when compared to SNF and MOGONET. First, the number of layers in the graph is not limited to the number of modalities, making this approach more flexible. Second, the similarities we compute reflect part of the data, as opposed to an entire modality. In cases where a lot of features are highly correlated, computing a single distance will drown out other potentially useful links. Finally, the aforementioned methods are based on the assumption that a link between samples for an entire modality is a good proxy for their respective final tasks, while we only assume this for some sub-groups of features that will be weighted by the neural network.

## 3.3 Multiplex Graph Construction

Let $K$ be the number of clusters generated from the feature clustering, then $\mathbf{X}^k$ denotes the features of cluster $k$ for all sample. We build a multiplex graph with $K$ layers, each layer representing similarities for an associated cluster. Similarities $s_k$ for samples in cluster $k$ are defined as

$$s_k\left(\mathbf{x}_i^k, \mathbf{x}_j^k\right) = e^{-\gamma \|\mathbf{x}_i^k - \mathbf{x}_j^k\|_2} \tag{7}$$

where $\mathbf{x}_i^k$ represents features in cluster $k$ for the $i$-th sample, and $\gamma = 0.5$. As some nodes can be left without any neighbors, we also link singleton nodes with their 10 nearest neighbors. The adjacency matrix $\mathbf{A}^k$ is then computed as in equation 4 with $\epsilon = 0.5$ for each layer. There is a number of other ways we can construct these graphs. We could for instance combine multiple kernel functions for the similarity computation [26], use different thresholds, and many other variants. It is thus difficult to know for certain which similarities to combine and how, and this might cause us to over engineer these similarities based on the dataset. It could be possible to design a neural network making that choice for us, but defining a fully differentiable method for generating a multiplex graph is not trivial.

## 3.4 Encoder Architecture

The network, shown schematically in Figure 1, is composed of two main parts, an encoder and a decoder. The encoder makes use of the pre-computed multiplex graph to encode samples into a low-dimensional embedding space. The embeddings are then fed into a regular decoder composed of two fully connected blocks. Each block is a composed of a fully connected layer, a dropout layer and an activation function. The output of the decoder is the reconstructed modality, and is compared directly with the ground truth.

To allow a wider range of applications for our approach, the head of the graph encoder predicts the mean and variance of a multivariate gaussian distribution rather than the latent representation directly. This can enable us to generate new samples for the output omic directly from the latent space, at no cost of performance for the observed metrics.

The graph encoder, depicted in detail in Figure 2 is similar in stucture to the network used in HDMI [11] and MxGNN [16] as it is a multiplex graph neural network with a similar attention mechanism. However we do not use any contrastive loss or pooling layer. An initial fully connected block reduces the intial dimension of the data (6000 + 5000 in our case) to an embedding size of 512. The embeddings are then passed to a one-layer graph convolution for each cluster defined in equation 8, and recasted by a residual connection.

$$\mathbf{Z}_k = f_k(\mathbf{X}, \hat{\mathbf{A}}_k) = \hat{\mathbf{A}}_k \mathbf{X} \mathbf{W}_k \tag{8}$$

where $\hat{\mathbf{A}}_k = \mathbf{A}_k + \mu\mathbf{I}$, $\mathbf{A}_k$ is the adjacency matrix for cluster $k$, $\mathbf{I}$ is the identity matrix and $\mu$ is a self connection hyper-parameter.

Similarly to HDMI [11], the output embeddings from the GCN are passed to a attention layer. The attention weights are computed as :

$$\alpha_n^r = tanh(\mathbf{y}^r \cdot LeakyRelu(\mathbf{W}_a^r \mathbf{h}_n^r)) \tag{9}$$

where $\mathbf{y}^r$ and $\mathbf{W}_a^r$ are learnable parameters, and a LeakyRelu activation function with a slope 0.2. The weights are the normalized using the softmax function

$$\alpha_n^r = \frac{\exp\left(\alpha_n^r\right)}{\sum_{r'=1}^{R} \exp\left(\alpha_n^{r'}\right)} \tag{10}$$

The final embedding of the $n$-th node is then obtained by a weighted average

$$\mathbf{h}_n = \sum_{r=1}^{R} \alpha_n^r \, \mathbf{h}_n^r \tag{11}$$

The role of this attention mechanism is to discern meaningful modalities for the downstream task. Unlike graph attention networks [28], this attention mechanism works solely on layers and not on nodes in each layer. In contrast to social or molecular graphs, the graph structure of our layers is not as relevant, and thus a simple GCN layer was chosen as our message passing paradigm.

## 3.5 Variational Encoder Decoder

Variational autoencoders (VAE) [13] are a class of deep neural networks capable of learning meaningful latent representations from high dimensional data. Let some dataset $\{\mathbf{x}^{(i)}\}_{i=1}^{N}$ with $N$ samples and $\mathbf{x}^{(i)} \in \mathbb{R}^d$ where $d$ is large. We assume that $\mathbf{x}^{(i)}$ are i.i.d. samples from some distribution $\mathbf{x}$ and that this data is itself generated by some random process involving a latent representation $\mathbf{z} \in \mathbb{R}^l$ where $l << d$. Latent representations from samples come from a prior distribution $p_{\boldsymbol{\theta}}(\mathbf{z})$ where $\boldsymbol{\theta}$ is a vector of learnable parameters. From a known $\mathbf{z}$, each sample is generated from a conditional distribution $p_\theta(\mathbf{x}|\mathbf{z})$. The approximate modelled probability is then

$$p_\theta(\mathbf{x}) = \int_z p_\theta(\mathbf{x} \mid \mathbf{z}) p_\theta(\mathbf{z}) dz \tag{12}$$

This integral is intractable, to remedy this we approximate the true posterior $p_\theta(\mathbf{z} \mid \mathbf{x})$ by an approximation $q_\phi(\mathbf{z} \mid \mathbf{x})$ where $\phi$ is another set of parameters. Maximizing the Kullback-Leibler divergence between these two quantities is equivalent to minimzing the evidence lower bound, or ELBO :

$$\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x} \mid \mathbf{z}) - D_{\mathrm{KL}}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z})\right) \tag{13}$$

where $D_{\mathrm{KL}}$ is the Kullback-Leibler divergence. This term however can be analytically derived as both $q_\phi(\mathbf{z} \mid \mathbf{x})$ and $p_\theta(\mathbf{z})$ are known distributions. We use a variation of this method, a variational encoder-decoder [13], to generate suitable latent representations for our omics translation task. The only difference being that the input sample is a different vector from the ground truth target.

The ELBO minimization is composed of the reconstruction loss between the reconstructed and the original sample, and the KL divergence between the estimated $q_\phi(\mathbf{z}|\mathbf{x})$ and the normal distribution. As the choice of prior for $p_{\boldsymbol{\theta}}(\mathbf{z})$, we simply use a normal distribution. Our loss function, initially defined in [13], is described in equation 14, we use the binary cross-entropy for our reconstruction loss and average over all training samples.

$$\mathcal{L}_{vae} = \frac{1}{N} \sum_{j=1}^{N} BCE\left(\mathbf{y}, \hat{\mathbf{y}}\right) + \mathcal{L}_{KL} \tag{14}$$

where $\mathbf{y}$ is the output modality and $\hat{\mathbf{y}}$ is the reconstructed approximation. $\mathcal{L}_{KL}$ is the regularization loss, which is the KL-divergence

$$\mathcal{L}_{KL} = D_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}) \| \mathcal{N}(\mathbf{0}, \mathbf{I})) \tag{15}$$

As the two distributions in the KL-divergence are Gaussian, an analytical formulation of this loss can be computed. Table 1 provides an overview of the most important hyperparameters.

# 4 EXPERIMENTS

## 4.1 Dataset

We use the TCGA [27] multi-omics data (available for download here http://acgt.cs.tau.ac.il/multi_omic_benchmark/download.html) for all of our experiments, and use all ten available cancer types. Data for each cancer type comprises of three omics layers: gene expression, DNA methylation, and MicroRNA (miRNA) expression. We use the same proprocessing as Cantini et al. [1], by using the natural logarithm of the miRNA expression level and gene expressions. We also only keep the first 6000 features of gene expression with the highest variance. Finally, only samples for which all three modalities were available were kept. The number of samples ranges from 170 for Acute Myeloid Leukemia (AML) to 621 for Breast cancer for a total of 3023 samples and 6000, 5000, 1508 features for gene expression, DNA methlyation and miRNA respectively. For all our results, we choose to reconstruct miRNA from gene expression and DNA methylation.

## 4.2 Experimental Setup

We test our method for two different splits of training and testing data. We define split A as random split in (5%, 5%, 90%) for train, validation and test data percentage respectively. For split B, we choose (80%, 5%, 15%). In both cases we pick the best performing model on validation data and evaluate it on the test set. We average the performance over 5 runs, and to ensure that our comparisons are fair we use the exact same splits for all methods. Our network is trained with the Adam optimizer and a stepwise learning rate scheduler of coefficient 0.5 applied every 1000 iterations. The values of important hyper-parameters are summarized in table 1. All models were trained on an NVIDIA GTX 3090 and an Intel(R) Core(TM) i9-10900X CPU @ 3.70GHz.

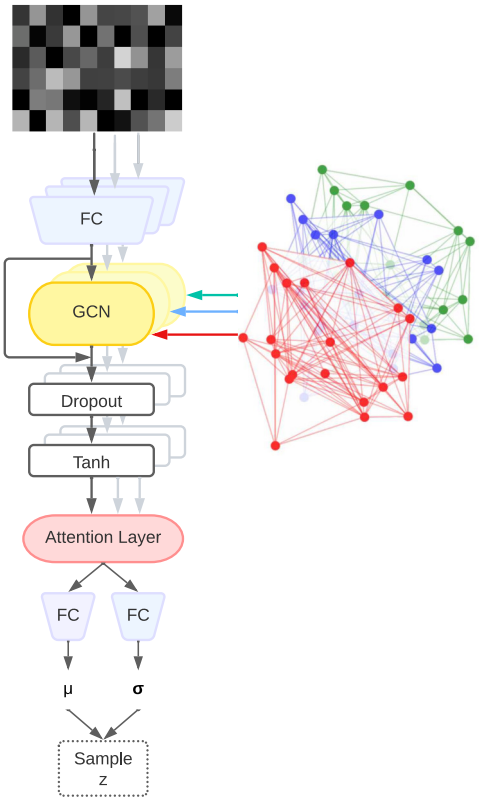


**Figure 2: Encoder Architecture. The initial fully connected layer condensed the initial high dimensional space to the network dimension. The K layers of the multiplex graph are fed to individual GCNs and combined by an attention mechanism. The final fully connected blocks project the output of the attention layer to a lower dimensional space (Embedding dimension 1 to 2 referenced in Table 1)**

### 4.3 Comparison Methods and Performance Assessment

For our baselines, we compare our method with multivariate LASSO regression (with regularization parameter $\alpha = 1$), as well as a traditional variational encoder decoder (named VED). The VED encoder is a simple MLP composed of two fully connected blocks and has the exact same decoder as our proposed approach. In addition, we compare our method with OmiTrans [37], a GAN-based neural network that, to the best of our knowledge, is the state of the art for omics-to-omics translation. OmiTrans was slightly modified to accommodate for our experiments, and the two input omics were simply concatenated to form the network input. We also compare the performance when generating one graph per modality, with graphs computed with the same method as SNF.

| Hyper-parameter | Value |
|---|---|
| $K$ | 10 |
| Embedding dimension 1 | 512 |
| Embedding dimension 2 | 256 |
| Dropout Rate | 0.1 |
| Maximum number of iterations | 15000 |
| Initial learning rate | 0.001 |
| Self connection coefficient $\mu$ | 3 |

**Table 1: Values of hyper-parameters for our method. $K$ denotes the number of feature clusters. Embedding dimensions 1 and 2 refer to the dimension of the graph encoder and the final embeddings, respectively.**

We use three different metric to assess the quality of our reconstruction. We compute the average mean square error (MSE), the mean absolute error (MAE) and the coefficient of determination ($R^2$) .

## 5 RESULTS

Our results for the reconstruction of miRNA expression from gene expression and DNA methylation are summarized in Table 2. For split A, both our graph-based methods perform better than OmiTrans (Wilcoxon signed-rank test on samples mean squared error, p = $2 \times 10^{-19}$). However there seems to be no significant gain or loss in performance when comparing the graph construction methods, and LASSO also performs similarly as our graph encoder.

For split B, our approach also outperforms OmiTrans on MSE (p = $7 \times 10^{-4}$). The increased amount of training data also validates our graph construction method (p = $1.6 \times 10^{-2}$) compared to SNF. Interestingly however, compared to OmiTrans we do not observe better results for MAE (non-significant, p=0.12). Our approach seem better equipped to handle outliers while OmiTrans does slightly better on average.

To better understand why our graph encoder performs significantly better than VED, we plotted the t-SNE projections of the target modality and of latent representation of each sample for the two methods on figure 3. We can see that most of the types of cancer are distinguishable for both the raw targets and our approach, while they collapse in VED. As the decoders are exactly the same in both methods, we conclude that encoding with fully connected blocks results in less expressive latent encodings. Admittedly, our architecture for VED might not be optimal for this problem, but we have not found a single combination of layers that would provide a better suited latent representation.

To summarize, we showed that our graph construction method can be advantageous in cases where enough data is provided. We also showed that our multiplex graph based approach can produce state of the art results for a multi-omics to omics translation task.

## 6 DISCUSSION AND CONCLUSION

We proposed a new method for multi-view omics-to-omics translation using a multiplex graph neural network encoder. While we showed that this type of method can do well for one type of omics it is still not clear to what extent this method can apply to other

| | Split A | | | Split B | | |
|---|---|---|---|---|---|---|
| | MSE ↓ | MAE ↓ | $R^2$ ↑ | MSE ↓ | MAE ↓ | $R^2$ ↑ |
| LASSO | 0.3768 ± 0.0015 | 0.2626 ± 0.0017 | 0.9546 ± 0.0002 | 0.3759 ± 0.0028 | 0.2609 ± 0.0016 | 0.9548 ± 0.0003 |
| VED | 0.7548 ± 0.0281 | 0.3609 ± 0.0042 | 0.8971 ± 0.0040 | 0.7751 ± 0.0080 | 0.3634 ± 0.0048 | 0.8949 ± 0.0020 |
| OmiTrans | 0.3876 ± 0.0039 | 0.2674 ± 0.0022 | 0.9538 ± 0.0005 | 0.2775 ± 0.0045 | **0.2117 ± 0.0026** | 0.9670 ± 0.0004 |
| Ours (SNF) | **0.3639 ± 0.0058** | **0.2570 ± 0.0010** | **0.9558 ± 0.0008** | 0.2537 ± 0.0086 | 0.2226 ± 0.0042 | 0.9685 ± 0.0010 |
| Ours | 0.3688 ± 0.0050 | 0.2577 ± 0.0017 | 0.9557 ± 0.0006 | **0.2401 ± 0.0046** | 0.2163 ± 0.0026 | **0.9702 ± 0.0006** |

**Table 2: Reconstruction performance of the tested methods for different metrics. Split A and B correspond to (train, validation, test) sample percentages of (5%, 5%, 90%) and (80%, 5%, 15%), respectively. Ours(SNF) denotes using our architecture with a multiplex graph computed like SNF (one graph per modality), while Ours denotes using the feature clustering step for multiplex graph construction.**

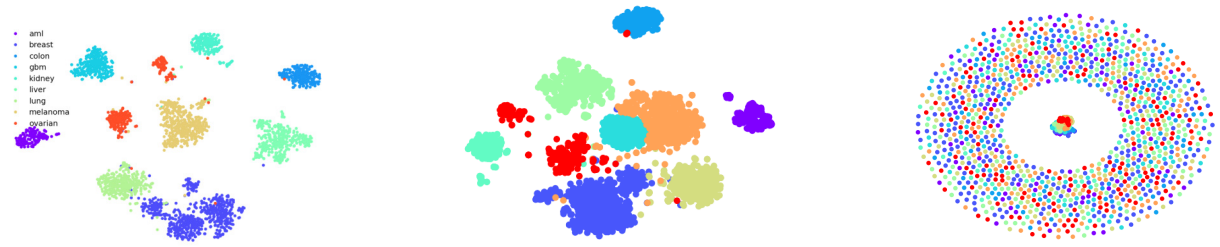


**Figure 3: Two-dimensional t-SNE projections from our best model trained on split B using (left) the original high-dimensional target miRNA expression data, (middle) the latent space of our approach, and (right) the latent space of a VED without a graph architecture.**

datasets and modalities, and further experimentation is required. On the other hand there is no reason for this approach to be limited to omics translation, and it would be worth to investigate other problems such as survival prediction, disease sub-typing, or other types of predictions.

There is room for improvement in our pipeline. Our method for splitting the features into clusters is rather simple, and generating the graph based on other factors that do not stem directly from the data (isolating important genes or clustering them based on some exterior criteria such as biological pathways or ontologies) could be an informative and interpretable way of injecting prior knowledge to the neural network. Moreover, our encoder decoder structure, while simple and easy to train, may lack the flexibility of other architectures such as GANs.

Despite its apparent advantages our method still suffers from some limitations. First, we have no a priori estimation of a suitable number of clusters $K$. This means that $K$ has to be treated like a hyper-parameter which, given the computational load of computing the multiplex graph, may take a significant amount of time. Second, we treat the variables in each cluster as equally important. This is also an issue as some individual variables might have a lot more expressive power than others in the same cluster; relatedly, our evaluation was limited to the few thousand features with the highest variance in gene expression, meaning that scaling issue could be present when moving to whole-genome coverage. Finally, the performance of our algorithm would significantly decrease when increasing $K$, meaning that our attention mechanism might not be sufficient to efficiently weight the graphs after a certain point.

A potential solution would be to compute similarities directly during training by doing some sort of similarity learning, such as Huai et al. [9]. Our approach could also be extended with some form of contrastive method such as Deep Graph Infomax [29] or Graph-MVP [12] to further improve performance on self-supervised tasks.

## 7 ACKNOWLEDGEMENTS

This work is supported by the Swiss National Science Foundation under Sinergia grant CRSII5_202276 / 1.

## 8 CODE AVAILABILITY

The source code for generating the graph and training the network can be downloaded from GitLab at :
https://gitlab.com/CGeorgantasCHUV/mgnn-omics-translation.

## REFERENCES

[1] Laura Cantini, Pooya Zakeri, Celine Hernandez, Aurelien Naldi, Denis Thieffry, Elisabeth Remy, and Anaïs Baudot. 2021. Benchmarking joint multi-omics dimensionality reduction approaches for the study of cancer. *Nature Communications* 12, 1 (Jan. 2021), 124. https://doi.org/10.1038/s41467-020-20430-7

[2] Wei Dai, Wenhao Yue, Wei Peng, Xiaodong Fu, Li Liu, and Lijun Liu. 2022. Identifying Cancer Subtypes Using a Residual Graph Convolution Model on a Sample Similarity Network. *Genes* 13, 1 (Jan. 2022), 65. https://doi.org/10.3390/genes13010065

[3] Bruno Messias F. de Resende, Eric K. Tokuda, and Luciano da Fontoura Costa. 2021. Unraveling the graph structure of tabular datasets through Bayesian and spectral analysis. *arXiv:2110.01421 [cs]* (Oct. 2021). http://arxiv.org/abs/2110.01421 arXiv: 2110.01421.

[4] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S. Yu. 2020. Enhancing Graph Neural Network-based Fraud Detectors against Camouflaged Fraudsters. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (Oct. 2020), 315–324. https://doi.org/10.1145/3340531.3411903 arXiv: 2008.08692.

[5] Miroslav Fiedler. 1973. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*. 23 (1973), 298–305. Place: London : Publisher: Kluwer/Plenum Publishers.

[6] Edian F. Franco, Pratip Rana, Aline Cruz, Víctor V. Calderón, Vasco Azevedo, Rommel T. J. Ramos, and Preetam Ghosh. 2021. Performance Comparison of Deep Learning Autoencoders for Cancer Subtype Detection Using Multi-Omics Data. *Cancers* 13, 9 (April 2021), 2013. https://doi.org/10.3390/cancers13092013

[7] Xiawei Guo, Yuhan Quan, Huan Zhao, Quanming Yao, Yong Li, and Weiwei Tu. 2021. TabGNN: Multiplex Graph Neural Network for Tabular Data Prediction. *arXiv:2108.09127 [cs]* (Aug. 2021). http://arxiv.org/abs/2108.09127 arXiv: 2108.09127.

[8] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous Graph Transformer. *arXiv:2003.01332 [cs, stat]* (March 2020). http://arxiv.org/abs/2003.01332 arXiv: 2003.01332.

[9] Mengdi Huai, Chenglin Miao, Qiuling Suo, Yaliang Li, Jing Gao, and Aidong Zhang. 2018. Uncorrelated Patient Similarity Learning. In *Proceedings of the 2018 SIAM International Conference on Data Mining (SDM)*. Society for Industrial and Applied Mathematics, 270–278. https://doi.org/10.1137/1.9781611975321.31

[10] Zhi Huang, Xiaohui Zhan, Shunian Xiang, Travis S. Johnson, Bryan Helm, Christina Y. Yu, Jie Zhang, Paul Salama, Maher Rizkalla, Zhi Han, and Kun Huang. 2019. SALMON: Survival Analysis Learning With Multi-Omics Neural Networks on Breast Cancer. *Frontiers in Genetics* 10 (2019). https://www.frontiersin.org/article/10.3389/fgene.2019.00166

[11] Baoyu Jing, Chanyoung Park, and Hanghang Tong. 2021. HDMI: High-order Deep Multiplex Infomax. *Proceedings of the Web Conference 2021* (April 2021), 2414–2424. https://doi.org/10.1145/3442381.3449971 arXiv: 2102.07810.

[12] Baoyu Jing, Yuejia Xiang, Xi Chen, Yu Chen, and Hanghang Tong. 2021. GraphMVP: Multi-View Prototypical Contrastive Learning for Multiplex Graphs. *arXiv:2109.03560 [cs]* (Oct. 2021). http://arxiv.org/abs/2109.03560 arXiv: 2109.03560.

[13] Diederik P. Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. (Dec. 2013). https://arxiv.org/abs/1312.6114v10

[14] Amy C. Kinsley, Gianluigi Rossi, Matthew J. Silk, and Kimberly VanderWaal. 2020. Multilayer and Multiplex Networks: An Introduction to Their Use in Veterinary Epidemiology. *Frontiers in Veterinary Science* 7 (2020), 596. https://doi.org/10.3389/fvets.2020.00596

[15] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv:1609.02907 [cs, stat]* (Feb. 2017). http://arxiv.org/abs/1609.02907 arXiv: 1609.02907.

[16] Yanyan Liang, Yanfeng Zhang, Dechao Gao, and Qian Xu. 2021. An End-to-End Multiplex Graph Neural Network for Graph Representation Learning. *IEEE Access* 9 (2021), 58861–58869. https://doi.org/10.1109/ACCESS.2021.3070690

[17] Changshu Liu, Liangjian Wen, Zhao Kang, Guangchun Luo, and Ling Tian. 2021. Self-supervised Consensus Representation Learning for Attributed Graph. *Proceedings of the 29th ACM International Conference on Multimedia* (Oct. 2021), 2654–2662. https://doi.org/10.1145/3474085.3475416 arXiv: 2108.04822.

[18] Markus Maier, Ulrike von Luxburg, and Matthias Hein. 2008. Influence of graph construction on graph-based clustering measures. In *Proceedings of the 21st International Conference on Neural Information Processing Systems (NIPS'08)*. Curran Associates Inc., Red Hook, NY, USA, 1025–1032.

[19] Patrick McGillivray, Declan Clarke, William Meyerson, Jing Zhang, Donghoon Lee, Mengting Gu, Sushant Kumar, Holly Zhou, and Mark Gerstein. 2018. Network Analysis as a Grand Unifier in Biomedical Data Science. *Annual Review of Biomedical Data Science* 1, 1 (July 2018), 153–180. https://doi.org/10.1146/annurev-biodatasci-080917-013444

[20] Fionn Murtagh and Pierre Legendre. 2014. Ward's Hierarchical Clustering Method: Clustering Criterion and Agglomerative Algorithm. *Journal of Classification* 31, 3 (Oct. 2014), 274–295. https://doi.org/10.1007/s00357-014-9161-z arXiv: 1111.6285.

[21] Erlin Pan and Zhao Kang. 2021. Multi-view Contrastive Graph Clustering. https://openreview.net/forum?id=NlB8:hXkbby

[22] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 85 (2011), 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html

[23] Nimrod Rappoport and Ron Shamir. 2018. Multi-omic and multi-view clustering algorithms: review and cancer benchmark. *Nucleic Acids Research* 46, 20 (Nov. 2018), 10546–10562. https://doi.org/10.1093/nar/gky889

[24] Jakob Seidlitz, František Váša, Maxwell Shinn, Rafael Romero-Garcia, Kirstie J. Whitaker, Petra E. Vértes, Konrad Wagstyl, Paul Kirkpatrick Reardon, Liv Clasen, Siyuan Liu, Adam Messinger, David A. Leopold, Peter Fonagy, Raymond J. Dolan, Peter B. Jones, Ian M. Goodyer, Armin Raznahan, and Edward T. Bullmore. 2018. Morphometric Similarity Networks Detect Microscale Cortical Organization and Predict Inter-Individual Cognitive Variation. *Neuron* 97, 1 (Jan. 2018), 231–247.e7. https://doi.org/10.1016/j.neuron.2017.11.039

[25] Qianqian Song, Jing Su, and Wei Zhang. 2021. scGCN is a graph convolutional networks algorithm for knowledge transfer in single cell omics. *Nature Communications* 12, 1 (June 2021), 3826. https://doi.org/10.1038/s41467-021-24172-y

[26] Nora K. Speicher and Nico Pfeifer. 2015. Integrating different data types by regularized unsupervised multiple kernel learning with application to cancer subtype discovery. *Bioinformatics* 31, 12 (June 2015), i268–i275. https://doi.org/10.1093/bioinformatics/btv244

[27] The Cancer Genome Atlas Research Network. 2008. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature* 455, 7216 (Oct. 2008), 1061–1068. https://doi.org/10.1038/nature07385

[28] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *arXiv:1710.10903 [cs, stat]* (Feb. 2018). http://arxiv.org/abs/1710.10903 arXiv: 1710.10903.

[29] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2018. Deep Graph Infomax. *arXiv:1809.10341 [cs, math, stat]* (Dec. 2018). http://arxiv.org/abs/1809.10341 arXiv: 1809.10341.

[30] Bo Wang, Aziz M. Mezlini, Feyyaz Demir, Marc Fiume, Zhuowen Tu, Michael Brudno, Benjamin Haibe-Kains, and Anna Goldenberg. 1993. Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods* 11, 3 (November 1993), 333–337.

[31] Bo Wang, Aziz M. Mezlini, Feyyaz Demir, Marc Fiume, Zhuowen Tu, Michael Brudno, Benjamin Haibe-Kains, and Anna Goldenberg. 2014. Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods* 11, 3 (March 2014), 333–337. https://doi.org/10.1038/nmeth.2810

[32] Jianian Wang, Sheng Zhang, Yanghua Xiao, and Rui Song. 2021. A Review on Graph Neural Network Methods in Financial Applications. *arXiv:2111.15367 [cs, q-fin, stat]* (Nov. 2021). http://arxiv.org/abs/2111.15367 arXiv: 2111.15367.

[33] Tongxin Wang, Wei Shao, Zhi Huang, Haixu Tang, Jie Zhang, Zhengming Ding, and Kun Huang. 2021. MOGONET integrates multi-omics data using graph convolutional networks allowing patient classification and biomarker identification. *Nature Communications* 12, 1 (June 2021), 3445. https://doi.org/10.1038/s41467-021-23774-w

[34] Eloise Withnell, Xiaoyu Zhang, Kai Sun, and Yike Guo. 2021. XOmiVAE: an interpretable deep learning model for cancer classification using high-dimensional omics data. *Briefings in Bioinformatics* 22, 6 (Nov. 2021), bbab315. https://doi.org/10.1093/bib/bbab315 arXiv: 2105.12807.

[35] Chen Xu and Zhengchang Su. 2015. Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics (Oxford, England)* 31, 12 (June 2015), 1974–1980. https://doi.org/10.1093/bioinformatics/btv088

[36] Bin Zhang and Steve Horvath. 2005. A General Framework for Weighted Gene Co-Expression Network Analysis. *Statistical Applications in Genetics and Molecular Biology* 4, 1 (Jan. 2005). https://doi.org/10.2202/1544-6115.1128

[37] Xiaoyu Zhang and Yike Guo. 2021. OmiTrans: generative adversarial networks based omics-to-omics translation framework. *arXiv:2111.13785 [cs, q-bio]* (Nov. 2021). http://arxiv.org/abs/2111.13785 arXiv: 2111.13785.