Supplementary Files

Table of Contents

Note S1: The derivation of equation (4) in the paper

Based on the theory of variational inference, we can get the following formula:

$$L(\theta,\phi;\mathbf{x}_i)=\mathrm{E}_{q_\phi(y_i,\mathbf{z}_i|\mathbf{x}_i)}\left(\log p_\theta\left(\mathbf{x}_i\mid y_i,\mathbf{z}_i\right)\right)-D_{KL}\left(q_\phi(y_i,\mathbf{z}_i\mid\mathbf{x}_i)\,\|\,p_\theta\left(y_i,\mathbf{z}_i\right)\right) \tag{i}$$

Firstly, for the first term in the right, we can obtain:

$$\begin{aligned}
\mathrm{E}_{q_\phi(y_i,\mathbf{z}_i|\mathbf{x}_i)}\left(\log p_\theta\left(\mathbf{x}_i\mid y_i,\mathbf{z}_i\right)\right)&=\mathrm{E}_{q_\phi(y_i|\mathbf{x}_i)q_\phi(\mathbf{z}_i|y_i,\mathbf{x}_i)}\left(\log p_\theta\left(\mathbf{x}_i\mid y_i,\mathbf{z}_i\right)\right)\\
&=\mathrm{E}_{q_\phi(y_i|\mathbf{x}_i)}\left(\mathrm{E}_{q_\phi(\mathbf{z}_i|y_i,\mathbf{x}_i)}\left(\log p_\theta\left(\mathbf{x}_i\mid y_i,\mathbf{z}_i\right)\right)\right)\\
&=\mathrm{E}_{q_\phi(y_i|\mathbf{x}_i)}\left(\mathrm{E}_{q_\phi(\mathbf{z}_i|y_i,\mathbf{x}_i)}\left(\sum_{m=1}^{M}\log p_\theta\left(\mathbf{x}_i^m\mid y_i,\mathbf{z}_i^m\right)\right)\right)\\
&=\sum_{m=1}^{M}\left(\mathrm{E}_{q_\phi(y_i|\mathbf{x}_i)}\left(\mathrm{E}_{q_\phi(\mathbf{z}_i^m|y_i,\mathbf{x}_i^m)}\left(\log p_\theta\left(\mathbf{x}_i^m\mid y_i,\mathbf{z}_i^m\right)\right)\right)\right)
\end{aligned} \tag{ii}$$

The derivation of equation (ii) uses the conditional independence of assumed generative model, which can be obtain using d-separate of probabilistic graphical model.

Then, we derive the second term using the same idea:

$$\begin{aligned}
&D_{KL}\left(q_\phi\left(y_i,\mathbf{z}_i\mid\mathbf{x}_i\right)\|\,p_\theta(y_i,\mathbf{z}_i)\right)\\
&=\mathrm{E}_{q_\phi(y_i|\mathbf{x}_i)q_\phi(\mathbf{z}_i|y_i,\mathbf{x}_i)}\left(\log\left(\frac{q_\phi\left(y_i,\mathbf{z}_i\mid\mathbf{x}_i\right)}{p_\theta(y_i,\mathbf{z}_i)}\right)\right)\\
&=\mathrm{E}_{q_\phi(y_i|\mathbf{x}_i)}\left(\mathrm{E}_{q_\phi(\mathbf{z}_i|y_i,\mathbf{x}_i)}\log\left(\frac{q_\phi\left(y_i,\mathbf{z}_i\mid\mathbf{x}_i\right)}{p_\theta(y_i,\mathbf{z}_i)}\right)\right)\\
&=\mathrm{E}_{q_\phi(y_i|\mathbf{x}_i)}\left(\mathrm{E}_{q_\phi(\mathbf{z}_i|y_i,\mathbf{x}_i)}\log\left(\frac{q_\phi\left(\mathbf{z}_i\mid y_i,\mathbf{x}_i\right)q_\phi\left(y_i\mid\mathbf{x}_i\right)}{p_\theta(\mathbf{z}_i\mid y_i)p\left(y_i\right)}\right)\right)\\
&=\mathrm{E}_{q_\phi(y_i|\mathbf{x}_i)}\left(\mathrm{E}_{q_\phi(\mathbf{z}_i|y_i,\mathbf{x}_i)}\sum_{m=1}^{M}\log\left(\frac{q_\phi\left(\mathbf{z}_i^m\mid y_i,\mathbf{x}_i^m\right)}{p_\theta(\mathbf{z}_i^m\mid y_i)}\right)\right)+\mathrm{E}_{q_\phi(y_i|\mathbf{x}_i)}\left(\log\frac{q_\phi\left(y_i\mid\mathbf{x}_i\right)}{p\left(y_i\right)}\right)\\
&=\sum_{m=1}^{M}\left(\mathrm{E}_{q_\phi(y_i|\mathbf{x}_i)}\left(D_{KL}\left(q_\phi\left(\mathbf{z}_i^m\mid y_i,\mathbf{x}_i^m\right)\|\,p_\theta(\mathbf{z}_i^m\mid y_i)\right)\right)\right)+D_{KL}\left(q_\phi\left(y_i\mid\mathbf{x}_i\right)\|\,p\left(y_i\right)\right)
\end{aligned} \tag{iii}$$

Finally, we get the equation (4) in the paper.

Note S2: The derivation of equation (5) in the paper

Following the hypothesis of our study, we mark the $q_\phi\left(y_i=c|\mathbf{x}_i\right)$ as π_c, mark the mean and variance of $q_\phi\left(\mathbf{z}_i^m|y_i=c,\mathbf{x}_i^m\right)$ as $\boldsymbol{\mu}_{ic}^m=\left(\mu_{ijc}^m\right)_{j=1}^{d_m^z}$ and $\boldsymbol{\sigma}_{ic}^{2m}=\left(\left(\sigma_{ijc}^m\right)^2\right)_{j=1}^{d_m^z}$, mark the mean and variance of $q_\phi\left(\mathbf{z}_i^m|y_i=c\right)$ as $\boldsymbol{\mu}_{ic}'^m=\left(\mu_{ijc}'^m\right)_{j=1}^{d_m^z}$ and $\left(\boldsymbol{\sigma}_{ic}'^m\right)^2=\left(\left(\sigma_{ijc}'^m\right)^2\right)_{j=1}^{d_m^z}$, mark the mean of $p_\theta\left(\mathbf{x}_i^m|y_i=c,\mathbf{z}_i^m\right)$ as $\mathbf{x}_i'^m=\left(x_{ijc}'^m\right)_{j=1}^{d_m^x}$ for $c=1,\ldots,C$.

The first right term of equation (4) in the paper is:

$$\begin{aligned}
&\sum_{m=1}^{M}\left(\mathrm{E}_{q_\phi(y_i|\mathbf{x}_i)}\left(\mathrm{E}_{q_\phi(\mathbf{z}_i^m|y_i,\mathbf{x}_i^m)}\left(\log p_\theta\left(\mathbf{x}_i^m\mid y_i,\mathbf{z}_i^m\right)\right)\right)\right)\\
&=\sum_{m=1}^{M}\sum_{c=1}^{C}\pi_{ic}\mathrm{E}_{q_\phi(\mathbf{z}_i^m|y_i,\mathbf{x}_i^m)}\left(\log p_\theta\left(\mathbf{x}_i^m\mid y_i,\mathbf{z}_i^m\right)\right)\\
&=\sum_{m=1}^{M}\mathrm{E}_{q_\phi(\mathbf{z}_i^m|y_i,\mathbf{x}_i^m)}\left(\sum_{c=1}^{C}\pi_{ic}\left(-\frac{d_m^x}{2}\log 2\pi\sigma^2+\frac{1}{2\sigma^2}\sum_{j=1}^{d_m^x}\left(x_{ijc}^m-x_{ijc}'^m\right)^2\right)\right)\\
&=\sum_{m=1}^{M}-\frac{d_m^x}{2}\log 2\pi\sigma^2+\frac{1}{2\sigma^2}\mathrm{E}_{q_\phi(\mathbf{z}_i^m|y_i,\mathbf{x}_i^m)}\left(\sum_{m=1}^{M}\sum_{c=1}^{C}\sum_{j=1}^{d_m^x}\left(x_{ijc}^m-x_{ijc}'^m\right)^2\pi_{ic}\right)
\end{aligned}\tag{iv}$$

The main part of second right term of equation (4) is the Kullback–Leibler divergence of two multivariate Gaussian distributions. It can be derived that:

$$\begin{aligned}
&D_{KL}\left(q_\phi\left(\mathbf{z}_i^m\mid\mathbf{x}_i^m,y_i=c\right)\|\,p_\theta(\mathbf{z}_i^m\mid y_i=c)\right)\\
&=-\frac{d_m^z}{2}\log 2\pi-\sum_{j=1}^{d_m^z}\log\frac{\sigma_{ijc}^m}{\sigma_{ijc}'^m}-\sum_{j=1}^{d_m^z}\mathrm{E}_{q_\phi(\mathbf{z}_i^m|\mathbf{x}_i^m,y_i=c)}\left(\frac{\left(z_{ijc}^m-\mu_{ijc}^m\right)^2}{2\left(\sigma_{ijc}^m\right)^2}-\frac{\left(z_{ijc}^m-\mu_{ijc}'^m\right)^2}{2\left(\sigma_{ijc}'^m\right)^2}\right)\\
&=-\frac{d_m^z}{2}\log 2\pi-\sum_{j=1}^{d_m^z}\log\frac{\sigma_{ijc}^m}{\sigma_{ijc}'^m}-\sum_{j=1}^{d_m^z}\left(\frac{1}{2}-\mathrm{E}_{q_\phi(\mathbf{z}_i^m|\mathbf{x}_i^m,y_i=c)}\left(\frac{\left(z_{ijc}^m-\mu_{ijc}'^m\right)^2}{2\left(\sigma_{ijc}'^m\right)^2}\right)\right)\\
&=-\frac{d_m^z}{2}\log 2\pi-\sum_{j=1}^{d_m^z}\log\frac{\sigma_{ijc}^m}{\sigma_{ijc}'^m}-\sum_{j=1}^{d_m^z}\left(\frac{1}{2}-\mathrm{E}_{q_\phi(\mathbf{z}_i^m|\mathbf{x}_i^m,y_i=c)}\left(\frac{\left(z_{ijc}^m-\mu_{ijc}^m\right)^2-2\left(\mu_{ijc}'^m-\mu_{ijc}^m\right)z_{ijc}^m+\left(\mu_{ijc}'^m\right)^2-\left(\mu_{ijc}^m\right)^2}{2\left(\sigma_{ijc}'^m\right)^2}\right)\right)\\
&=-\frac{d_m^z}{2}\log 2\pi-\sum_{j=1}^{d_m^z}\log\frac{\sigma_{ijc}^m}{\sigma_{ijc}'^m}-\sum_{j=1}^{d_m^z}\left(\frac{1}{2}-\left(\frac{\left(\sigma_{ijc}^m\right)^2}{2\left(\sigma_{ijc}'^m\right)^2}+\frac{\left(\mu_{ijc}'^m-\mu_{ijc}^m\right)^2}{2\left(\sigma_{ijc}'^m\right)^2}\right)\right)\\
&-\frac{d_m^z}{2}(\log 2\pi+1)+\frac{1}{2}\left(\sum_{j=1}^{d_m^z}-\log\frac{\left(\sigma_{ijc}^m\right)^2}{\left(\sigma_{ijc}'^m\right)^2}+\frac{\left(\sigma_{ijc}^m\right)^2}{\left(\sigma_{ijc}'^m\right)^2}+\frac{\left(\mu_{ijc}'^m-\mu_{ijc}^m\right)^2}{\left(\sigma_{ijc}'^m\right)^2}\right)
\end{aligned}\tag{v}$$

The third term is:

$$D_{KL}\left(q_\phi\left(y_i|\mathbf{x}_i\right)\|\,p\left(y_i\right)\right)=\sum\nolimits_{c=1}^{C}\pi_{ic}\log\frac{\pi_{ic}}{1/C}=\log C+\sum\nolimits_{c=1}^{C}\pi_{ic}\log\pi_{ic}\tag{vi}$$

Then, we omit the constant terms, use reparameterization trick and assume the variance of error is 1 to get the Monte Carlo estimator $L'\left(\theta,\phi;\mathbf{x}_i\right)$ of $L\left(\theta,\phi;\mathbf{x}_i\right)$.

Note S3: The pseudo code of MCluster-VAEs

Algorithm Minibatch stochastic descent training of MCluster-VAEs

Input: expression matrix $X = \{X^m | m = 1, \dots, M\}$ for M omics and N samples, X^m has dimension $N \times d^m$; max number of steps T.

Output: clustering assignments $y = (y_1, \dots, y_N)$

define $KL(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\mu}', \boldsymbol{\sigma}') = -\sum_j \left[2\log(\sigma_j/\sigma_j') - (\sigma_j/\sigma_j')^2 - ((\mu_j - \mu_j')/\sigma_j')^2\right]$

Standardize X^m for $m = 1, \dots, M$.

for $t = 1, \dots, T$ **do**

  Sample minibatch of b samples $\{\{\boldsymbol{x}_1^1, \dots, \boldsymbol{x}_1^M\}, \dots, \{\boldsymbol{x}_b^1, \dots, \boldsymbol{x}_b^M\}\}$.

  Compute $(\pi_1, \dots, \pi_C)^T = \boldsymbol{\pi}_i = q_\phi(y_i | \{\boldsymbol{x}_i^1, \dots, \boldsymbol{x}_i^M\})$.

  if using gumbel softmax trick

    Sample y_i based on formula (8) in the paper.

    Compute $\boldsymbol{\mu}'^m_i$ and $\boldsymbol{\sigma}'^m_i$ of $p_\theta(\boldsymbol{z}_i^m | y_i)$.

    for $m = 1, \dots, M$ **do**

      Compute mean $\boldsymbol{\mu}_i^m$ and $\boldsymbol{\sigma}_i^m$ of $q_\phi(\boldsymbol{z}_i^m | \boldsymbol{x}_i^m, y_i)$.

      Sample $\boldsymbol{z}_i^m$ based on $\boldsymbol{\mu}_i^m$ and $\boldsymbol{\sigma}_i^m$.

      Compute $\boldsymbol{x}'^m_i$ using $p_\theta(\boldsymbol{x}_i^m | y_i, \boldsymbol{z}_i^m)$.

      Compute $L_{rec}^m = 1/b \sum_i \left\| \boldsymbol{x}_i^m - \boldsymbol{x}'^m_i \right\|_2^2$

      Compute $L_{cprior}^m = -1/2b \sum_i KL\left(\boldsymbol{\mu}_i^m, \boldsymbol{\mu}'^m_i, \boldsymbol{\sigma}_i^m, \boldsymbol{\sigma}'^m_i\right)$

    end for

    compute $L_{rec} = \sum_m L_{rec}^m$, $L_{cprior} = \sum_m L_{cprior}^m$.

  else

    for $c = 1, \dots, C$ **do**

      Compute $\boldsymbol{\mu}'^m_{ic}$ and $\boldsymbol{\sigma}'^m_{ic}$ of $p_\theta(\boldsymbol{z}_i^m | y_i = c)$.

      for $m = 1, \dots, M$ **do**

        Compute mean $\boldsymbol{\mu}_{ic}^m$ and $\boldsymbol{\sigma}_{ic}^m$ of $q_\phi(\boldsymbol{z}_i^m | \boldsymbol{x}_i^m, y_i = c)$.

        Sample $\boldsymbol{z}_{ic}^m$ based on $\boldsymbol{\mu}_{ic}^m$ and $\boldsymbol{\sigma}_{ic}^m$.

        Compute $\boldsymbol{x}'^m_{ic}$ using $p_\theta(\boldsymbol{x}_i^m | y_i = c, \boldsymbol{z}_{ic}^m)$.

        Compute $L_{rec}^{mc} = 1/b \sum_i \left\| \boldsymbol{x}_i^m - \boldsymbol{x}'^m_{ic} \right\|_2^2$

        Compute $L_{cprior}^m = -1/2b \sum_i KL\left(\boldsymbol{\mu}_{ic}^m, \boldsymbol{\mu}'^m_{ic}, \boldsymbol{\sigma}_{ic}^m, \boldsymbol{\sigma}'^m_{ic}\right)$

      end for

    end for

    Compute $L_{rec} = \sum_m \sum_c L_{rec}^{mc} \pi_{ic}$, $L_{cprior} = \sum_m \sum_c L_{cprior}^{mc} \pi_{ic}$.

  Compute $L_{centropy} = 1/b \sum_i \sum_c \pi_{ic} \log \pi_{ic}$.

  Compute $\gamma(t) = 1 + 2 \cdot (1 + \cos(t\pi/T))$.

  Compute $Loss = L_{rec} + L_{cprior} + \gamma(t) L_{centropy}$.

  Perform a gradient descent step on $Loss$.

end for

Obtain y by trained $q_\phi(y_i | \{\boldsymbol{x}_i^1, \dots, \boldsymbol{x}_i^M\})$ on X.

Note S4: The definitions of ACC, ARI, NMI and F1

Unsupervised Accuracy (ACC) is defined as:

$$ACC = \max_{m \in \mathrm{M}} \frac{\sum_{i=1}^{N} 1\{l_i = m(c_i)\}}{N} \tag{vii}$$

where N is the total number of samples, l_i is the ground truth label of cancer types, c_i is the cluster assignment obtained by the algorithm, and M is the set of all possible one-to-one mappings between clustering assignments and labels. The best mapping can be obtained by using the KuhnMunkres algorithm (1). Compared to other metrics, ACC is intuitive. It provides the ability to compare with almost any method, even supervised method. The value of ACC lies between 0 and 1 and a high ACC value indicates the good performance of a clustering method.

Adjusted Rand index (ARI) is a widely used metric for measuring the concordance between two clustering results. Given two clustering U and V, we calculate the following four quantities:

- a: number of objects in a pair are placed in the same group in U and in the same group in V;
- b: number of objects in a pair are placed in the same group in U and in different groups in V;
- c: number of objects in a pair are placed in the same group in V and in different groups in U;
- d: number of objects in a pair are placed in different groups in U and in different groups in V.

ARI is defined as follows:

$$ARI = \frac{2(ad - bc)}{(a+b)(b+d)+(a+c)(c+d)} \tag{viii}$$

In the paper, U and V are the ground-truth labels and clustering assignments respectively. The value of ARI lies between 0 and 1, and a high ARI value indicates the good performance of a clustering method.

F measure (F1) is symmetric measure that combines precision and recall, which is equivalent to Dice's measure and is defined as (2):

$$F1 = \frac{2a}{a+b+c} \tag{ix}$$

where a, b and c are defined as in ARI. F1 takes on values between 0 and 1, and a high F1 value indicates the good performance of a clustering method. Because F1 combines precision and recall, it is more capable of evaluating unbalanced data.

Normalized Mutual Information (NMI) is another typical criteria to evaluate the consistency between the obtained clustering and the ground-truth labels of the samples. NMI is defined as

$$NMI = I\left(U,V\right) / \max\left\{H(U), H(V)\right\} \tag{x}$$

where $I\left(U,V\right)$ is the mutual information between U and V, and $H(U)$ represents the entropy of the clustering U. Specifically, assuming that U has P clusters and V has Q clusters, the mutual information is computed as follows:

$$I\left(U,V\right) = \sum_{p=1}^{P}\sum_{q=1}^{Q}\frac{\left|U_p \cap V_q\right|}{N}\log\frac{N\left|U_p \cap V_q\right|}{\left|U_p\right| \times \left|V_q\right|} \tag{xi}$$

where $\left|U_p\right|$ and $\left|V_q\right|$ denote the cardinality of the p-th cluster in U and the q-th cluster in V, respectively. The entropy of each cluster assignment is calculated by $H(U) = -\sum_{p=1}^{P}\left(\left|U_p\right|/N\right)\log\left(\left|U_p\right|/N\right)$ and $H(V) = -\sum_{q=1}^{Q}\left(\left|V_q\right|/N\right)\log\left(\left|V_q\right|/N\right)$. NMI takes on values between 0 and 1, measuring the concordance of two clustering results. In the experiments, we calculated the obtained clustering with respect to the true labels. Therefore, a higher NMI refers to higher concordance with ground-truth, i.e. a more accurate label assignment of each omics data.

Note S5: The definitions of categories of comparison methods

Single Input (SI) is the simplest approach. It concatenates omic matrices to form a single matrix with features from multiple omics, and applies single-omic clustering algorithms on that matrix.

In ***late integration***, each omic is clustered separately and the clustering solutions are integrated to obtain a single clustering solution.

Similarity-based methods use similarities or distances between samples in order to cluster data. These methods compute the similarities between samples in each omic separately, and vary in the way these similarities are integrated.

Dimension reduction-based methods assume the data have an intrinsic low dimensional representation, with that low dimension often corresponding to the number of clusters.

Statistical-based methods model the probabilistic distribution of the data. Some of these methods view samples as originating from different clusters, where each cluster defines a distribution for the data, while other methods do not explicitly use the cluster structure in the model.

Deep Learning-based (two-steps) methods use non-linear neural networks to learn an integrated representation of multi-omics data by the unsupervised framework (representation learning step) and then apply a traditional clustering algorithm to this representation (clustering step). Gaussian Mixture Model (GMM) or k-means usually are applied in the second step.

Note S6: The detail introduction and parameter setting of comparison methods

Single Input (SI) methods

K-means is a widely used clustering algorithm which uses a simple iterative optimization algorithm based on the objective function of the distance to the cluster center. We used *kmeans* function from R *stats* package with the parameters *iter.max = 10* and *nstart = 1*.

Spectral clustering is a widely used similarity-based method. First it calculate the affinity matrix and the spectral clustering objective is shown to be a relaxation of the discrete normalized cut in a graph, providing an intuitive explanation for the clustering. We used *spectralClustering* function from R *SNFtool* package with parameter *type = 3*.

Late Integration methods

COCA takes as input the binary vectors that represent each of the omic-specific cluster-groups and re-clusters the samples according to those vectors. One advantage of the method is that data are combined without the need for normalization steps. In addition, each omic influences the final integrated result with weight proportional to the number of distinct subtypes reproducibly found by Consensus Clustering. COCA is executed by *coca* function from R *coca* package with parameters *pItem = 0.8, choiceKmethod = "silhouette"* and *ccClMethod = "kmeans"*.

Dimension Reduction-based methods

CCA finds two projection vectors of dimensions, such that the projected data has maximum correlation. CCA only supports integration of two types of omics and **MCCA** expands it to more, which maximizes the sum of pairwise correlations between projections. MCCA depends on *MultiCCA* function from R *PMA* package with parameters *niter = 25, type = "standard"* and *ncomponents = 1*.

Similarity-based methods

SNF first constructs a similarity network for every omic separately then fuses together using an iterative procedure based on message passing. This process converges to a single similarity network, summarizing the similarity between samples across all omics. This network is partitioned using spectral clustering. SNF depends on *SNF* function from R *SNFtool* package with *arguments K = 20* and *t = 20*.

Similar to SNF, **ANF** first constructs a patient affinity network from each view, and then fuses all individual networks to get a more robust one for spectral clustering. ANF requires much less computation while generating as good as or even better results than those from SNF. ANF is executed by *ANF* function from R *ANF* package with arguments *K = 20, type = "two-step"* and *alpha = (1, 1, 0, 0, 0, 0, 0, 0)*.

CIMLR learns a measure of similarity between each pair of samples in a multi-omic dataset by combining multiple gaussian kernels per data type, corresponding to different, complementary representations of the data. It enforces a block structure in the resulting similarity matrix, which is then used for dimension reduction and k-means clustering. CIMLR relies on *CIMLR* function from R *CIMLR* package with arguments *k = 10* and *cores.ratio = 1*.

NEMO works in three phases. First, an inter-patient similarity matrix is built for each omic. Next, the matrices of different omics are integrated into one matrix. Finally, that network is clustered. NEMO can be applied to partial datasets in which some patients have data for only a subset of the omics, without performing data imputation. We performed *NEMO* using *nemo.clustering* function from *NEMO* package with argument *num.neighbors = 6*.

Statistical-based methods

iClusterBayes assumes that the data originate from a low dimension representation, which determines the cluster membership for each sample. Under this model iClusterBayes maximizes the likelihood of the observed data with a Bayesian regularization for sparse matrices and optimization is performed using an EM-like algorithm. iClusterBayes relies on *iClusterBayes* function from *iClusterPlus* package with arguments *type = "gaussian"*, *n.burnin=1000* and *n.draw=1200*.

Deep Learning-based methods (two-steps)

MAUI uses a multimodal, stacked VAE to extract latent factors which explain the variation across the different data modalities, capturing important aspects of cancer biology. The latent factors also can be used to identify disease subtypes and predict patient survival. MAUI has been implemented as a python package *MAUI* (https://github.com/BIMSBbioinfo/maui), but it would raise error after installation. We have rewritten the code using *pytorch* based its source code as method comparison. The hyperparameters used the default arguments from the python package.

DCAP inputs the multi-omics data into the unsupervised denoising Autoencoder (AE), obtains the representative features for the high dimensional input data, and then utilizes these learned features to accurately estimate cancer risks through the Cox proportional hazard model. At last, the patients are classified into two risk subgroups based on the median predicted risk value. We only used its autoencoder part for multi-omics representation extraction and then use K-means for clustering. The code was from https://github.com/Hua0113/DCAP.

Subtype-GAN is a deep adversarial learning approach based on the multiple-input multiple-output neural network to model the complex omics data accurately. The multiple input layers of the Subtype-GAN are relatively independent and are connected to the same shared layer simultaneously. Then, through the shared layer's hidden factor, Subtype-GAN used consensus clustering to obtain the number of subtypes and the subtyping label of each sample. Codes from https://github.com/haiyang1986/Subtype-GAN was used to implement Subtype-GAN. The hyperparameters were the default arguments in the code.

MCluster-VAEs (Deep Learning-based methods, one-steps)

The network architectures of MCluster-VAEs were shown in Table S3. The activation function used in MCluster-VAEs was GELU (3). The number of training epochs was 500. The learning rate varied with cosine schedule (4), whose initial value was 0.0008. Due to different sample size of datasets, we used different batch size for different dataset to improve the training speed. For the Pan Cancer dataset, the batch size was

512. For GBM and UVM, the batch size was 32. For other datasets, the batch size was 64.

Note S7: The relationship of MCluster-VAEs and other categories of comparison methods

MCluster-VAEs can be considered as a method with the excellent characteristics of statistics-based approaches, dimension reduction-based approaches and deep learning-based approaches. Firstly, MCluster-VAEs and most statistics-based approaches consider the clustering assignments as a latent categorical variable and try to infer this variable with Bayesian approaches. The difference between them is that the statistics-based approaches often have a strict distribution hypothesis, while the hypothesis of MCluster-VAEs is more relaxed. This moderate prior makes MCluster-VAEs can identify complicated relationships. Secondly, MCluster-VAEs could be considered as a dimension reduction model, which is similar to most dimension reduction-based methods, like MCCA and NMF. However, MCluster-VAEs uses more flexible non-linear embedding instead of linear project vector of these dimension reduction-based methods, which makes MCluster-VAEs learn rich representations. Thirdly, MCluster-VAEs is implemented by neural networks and trained by the standard mini-batch stochastic gradient descent algorithm, same as all deep learning-based models. However, as mentioned before, the new probabilistic model with the common latent clustering assignments, compatible with multiple data sources, leads better adaptability for multi-omics clustering task, enabling MCluster-VAEs to perform better for identifying cancer subtypes.

Table S1: The sample sizes and abbreviation of each cancer types in the Pan Cancer dataset

Table S2. The sample sizes and abbreviation of each cancer types in the Pan Cancer dataset

Full name	Abbreviation	Sample size
breast invasive carcinoma	BRCA	757
head & neck squamous cell carcinoma	HNSC	506
brain lower grade glioma	LGG	506
thyroid carcinoma	THCA	494
prostate adenocarcinoma	PRAD	484
lung adenocarcinoma	LUAD	448
uterine corpus endometrioid carcinoma	UCEC	411
bladder urothelial carcinoma	BLCA	401
stomach adenocarcinoma	STAD	365
liver hepatocellular carcinoma	LIHC	357
lung squamous cell carcinoma	LUSC	356
skin cutaneous melanoma	SKCM	351
kidney clear cell carcinoma	KIRC	306
cervical & endocervical cancer	CESC	291
colon adenocarcinoma	COAD	285
kidney papillary cell carcinoma	KIRP	268
sarcoma	SARC	250
esophageal carcinoma	ESCA	180
pancreatic adenocarcinoma	PAAD	176
acute myeloid leukemia	LAML	163
pheochromocytoma & paraganglioma	PCPG	161
testicular germ cell tumor	TGCT	133
thymoma	THYM	119

rectum adenocarcinoma	READ	91
mesothelioma	MESO	87
uveal melanoma	UVM	80
adrenocortical cancer	ACC	76
kidney chromophobe	KICH	65
uterine carcinosarcoma	UCS	55
diffuse large B-cell lymphoma	DLBC	47
cholangiocarcinoma	CHOL	36
ovarian serous cystadenocarcinoma	OV	9

Table S2: The categories and references of all methods

Table S2. The categories and references of all methods.

Method	Categories[1]	Reference
k-means	Single Input (SI)	(5)
spectral clustering	Single Input (SI)	(6)
MCCA	Dimension Reduction	(7)
COCA	Late Integration	(8)
ANF	Similarity-based	(9)
SNF	Similarity-based	(10,11)
CIMLR	Similarity-based	(12)
NEMO	Similarity-based	(13)
iClusterBayes	Statistical-based	(14)
MAUI (*VAE*)	Deep Learning-based (two-steps)	(15,16)
DCAP (*AE*)	Deep Learning-based (two-steps)	(17,18)
SubtypeGAN	Deep Learning-based (two-steps)	(19)
MCluster-VAEs	Deep Learning-based (one-steps)	

[1] These categories were from (20). The definitions of the categories are in Note S5.

Table S3: The architecture used in this study

Table S3. The architecture used in this study.

Module	Omics	Architecture
$q_\phi\left(y_i \vert \mathbf{x}_i\right)$	methylation	Feature extraction: [3139]-FC[100]
		Attention score: [100]-BN-GELU-FC[1]
	mRNA	Feature extraction: [3217]-FC[100]
		Attention score: [100]-BN-GELU-FC[1]
	CNA	Feature extraction: 3105-FC[100]
		Attention score: [100]-BN-GELU-FC[1]
	miRNA	Feature extraction: [383]-FC[100]
		Attention score: [100]-BN-GELU-FC[1]
	integration	100-BN-GELI-FC[C]
$q_\phi\left(\mathbf{z}_i^m \vert \mathbf{x}_i^m, y_i\right)$	methylation	[3139+C]-FC[250]-BN-GELU-FC[100, 100]
	mRNA	[3217+C]-FC[250]-BN-GELU-FC[100, 100]
	CNA	[3105+C]-FC[250]-BN-GELU-FC[100, 100]
	miRNA	[383+C]-FC[250]-BN-GELU-FC[30, 30]
$p_\theta(\mathbf{x}_i^m \mid y_i, \mathbf{z}_i^m)$	methylation	[100]-FC[100]-BN-GELU-FC[100]-BN-GELU-FC[3139]
	mRNA	[100]-FC[100]-BN-GELU-FC[100]-BN-GELU-FC[3217]
	CNA	[100]-FC[100]-BN-GELU-FC[100]-BN-GELU-FC[3105]
	miRNA	[30]-FC[100]-BN-GELU-FC[100]-BN-GELU-FC[383]
$p_\phi\left(y_i \vert \mathbf{x}_i\right)$	methylation	[C]-FC[100]
	mRNA	[C]-FC[100]
	CNA	[C]-FC[100]
	miRNA	[C]-FC[100]

Table S4: The -log10 *P*-values of differential survival of all methods in the ten specific cancer datasets

Table S4. The -log10 *P*-values of differential survival of all methods in the ten specific cancer datasets.

method	BLCA	BRCA	GBM	KIRC	LUAD	PAAD	SKCM	STAD	UCEC	UVM
ANF	1.8159	1.4608	1.5138	4.9506	1.4081	2.5610	7.0397	0.0539	5.2666	3.3797
CIMLR	2.3759	0.4716	0.8642	6.4446	1.1760	0.1073	3.4402	0.2190	3.3405	2.4458
COCA	2.8371	0.6305	2.3437	1.4509	1.3075	0.0495	0.7186	0.5722	3.2939	1.9414
DCAP(AE)	0.0888	0.1102	1.1542	2.5579	0.7328	0.0702	0.0090	0.0667	0.4334	2.8155
K-means	0.4023	0.1642	0.9606	5.2918	0.8173	2.3764	1.1572	0.0702	6.5030	1.6726
MAUI(VAE)	2.0530	0.3036	1.1864	6.4659	1.4144	2.3316	2.0814	0.0465	6.3615	2.9692
MCCA	0.1342	0.1028	2.8393	5.2952	0.5707	0.8841	0.7484	0.0661	3.6864	1.4618
MCluster-VAEs	4.1194	3.0046	4.6077	10.9667	2.9486	4.3510	9.9278	2.4809	9.0763	7.0163
NEMO	2.4987	1.0129	1.5215	5.6820	2.1347	1.8067	5.3235	1.0520	5.8855	2.1548
SNF	0.6634	2.2392	1.9938	9.8267	2.4157	2.7821	5.0360	0.8337	7.1849	1.8710
Spectral	0.9594	2.1258	0.8308	5.4081	1.5487	3.5806	5.0586	1.1563	3.5624	3.2524
SubtypeGAN	2.4905	1.9629	2.1812	7.0643	2.4668	1.5854	5.2628	1.3201	5.8270	4.9520
iCluster	0.2738	0.1804	0.0429	2.1646	0.0847	0.5394	0.5774	0.0964	5.8977	1.4007

Table S5: The number of significant enrichment clinical parameters of all methods in the ten specific cancer datasets

Table S5. The number of significant enrichment clinical parameters of all methods in the ten specific cancer datasets.

method	BLCA	BRCA	GBM	KIRC	LUAD	PAAD	SKCM	STAD	UCEC	UVM
ANF	5	5	0	4	2	1	4	1	2	1
CIMLR	6	5	0	5	2	0	4	2	2	1
COCA	5	4	2	3	1	0	0	3	2	0
DCAP(AE)	0	1	0	1	0	0	0	0	0	1
K-means	1	2	0	4	0	1	0	1	2	0
MAUI(VAE)	5	3	0	5	1	1	0	1	2	1
MCCA	1	2	1	4	0	0	0	0	2	0
MCluster-VAEs	6	6	2	7	4	4	4	4	2	1
NEMO	6	4	0	4	2	1	4	1	2	1
SNF	5	6	1	6	3	1	4	1	2	0
Spectral	1	6	0	6	3	1	4	2	2	1
SubtypeGAN	6	6	2	7	5	0	4	1	2	1
iCluster	0	0	0	2	0	1	1	1	2	0

Table S6: Marker genes for each BRCA subtypes

Table S6. Marker genes for each BRCA subtypes.

C1	C2	C3	C4	C5
NPY1R	STAC2	PRAME	PROM1	TFF1
AGR3	C4orf7	A2ML1	ELF5	AGR3
CPB1	SOX10	ONECUT2	SLC34A2	TFF3
LPPR3	KRT16	MMP1	GABRP	AGR2
LRP2	KRT15	GLDC	STAC2	C1orf64
ELOVL2	KRT5	VGF	CALML5	CYP2B7P1
PGR	FABP7	MAGEA6	LTF	ANKRD30A
SERPINA11	SFRP1	PRR11	KIF1A	FOXA1
DOK7	GABRP	CASP14	FABP7	SLC44A4
SERPINA6	KRT14	3-Sep	A2ML1	GP2
S100A9	CEACAM5	AGR3	AGR2	MIA
LBP	PPP2R2C	HMGCS2	FSIP1	C4orf7
CASP14	VSTM2A	PGR	LPPR3	FABP7
S100A7	CACNA1H	SCGB2A2	TMPRSS6	ROPN1B
GLYATL2	EEF1A2	PIP	NEURL	MSLN
S100A8	CPB1	NEK10	BMPR1B	KRT16
C2orf54	HS6ST3	TFF1	NKAIN1	HORMAD1
TDRD1	CPLX2	TFAP2B	AGR3	A2ML1
MUCL1	CYP2B7P1	ANKRD30A	KCNJ3	ROPN1
CLCA2	RIMS4	CYP4Z1	CPB1	GABRP

Table S7: Top 18 biological process items

Table S7. Top 18 biological process items.

GO	Category	Description	Count	%	Log10(P)	Log10(q)
GO:0030855	GO Biological Processes	epithelial cell differentiation	12	15.19	-7.48	-3.13
GO:0050786	GO Molecular Functions	RAGE receptor binding	3	3.8	-5.69	-1.82
GO:0048469	GO Biological Processes	cell maturation	6	7.59	-5.34	-1.7
GO:0046660	GO Biological Processes	female sex differentiation	5	6.33	-4.84	-1.29
GO:0008289	GO Molecular Functions	lipid binding	10	12.66	-4.36	-1.07
GO:0048871	GO Biological Processes	multicellular organismal homeostasis	7	8.86	-4.33	-1.07
GO:0022412	GO Biological Processes	cellular process involved in reproduction in multicellular organism	7	8.86	-4.05	-1.02
GO:0004175	GO Molecular Functions	endopeptidase activity	7	8.86	-3.85	-0.9
GO:0030510	GO Biological Processes	regulation of BMP signaling pathway	4	5.06	-3.77	-0.87
GO:0016324	GO Cellular Components	apical plasma membrane	6	7.59	-3.43	-0.69
GO:0052548	GO Biological Processes	regulation of endopeptidase activity	6	7.59	-3.05	-0.51
GO:0033674	GO Biological Processes	positive regulation of kinase activity	6	7.59	-2.75	-0.36
GO:0046903	GO Biological Processes	secretion	6	7.59	-2.74	-0.36
GO:0001676	GO Biological Processes	long-chain fatty acid metabolic process	3	3.8	-2.51	-0.21
GO:0008202	GO Biological Processes	steroid metabolic process	4	5.06	-2.35	-0.11
GO:0008285	GO Biological Processes	negative regulation of cell population proliferation	7	8.86	-2.35	-0.11
GO:0051046	GO Biological Processes	regulation of secretion	6	7.59	-2.28	-0.06
GO:0006820	GO Biological Processes	anion transport	5	6.33	-2.19	0

Table S8: MCluster-VAEs clusters (C1~C5) and previous subtypes on BRCA dataset

Table S8. MCluster-VAEs clusters (C1~C5) and previous subtypes (Basal: basal-like, Normal: normal-like, Lumb: luminal-B, LumA: luminal-A, Her2: HER2-enriched) on BRCA dataset.

Previous Subtypes	C5	C1	C3	C2	C4
Basal	170	6	0	0	0
Her2	3	65	1	11	0
LumA	0	8	309	60	171
LumB	0	8	16	92	91
Normal	23	19	51	19	25

Figure S1: Heatmaps of confusion matrix of clustering performance using MCluster-VAEs and twelve other methods

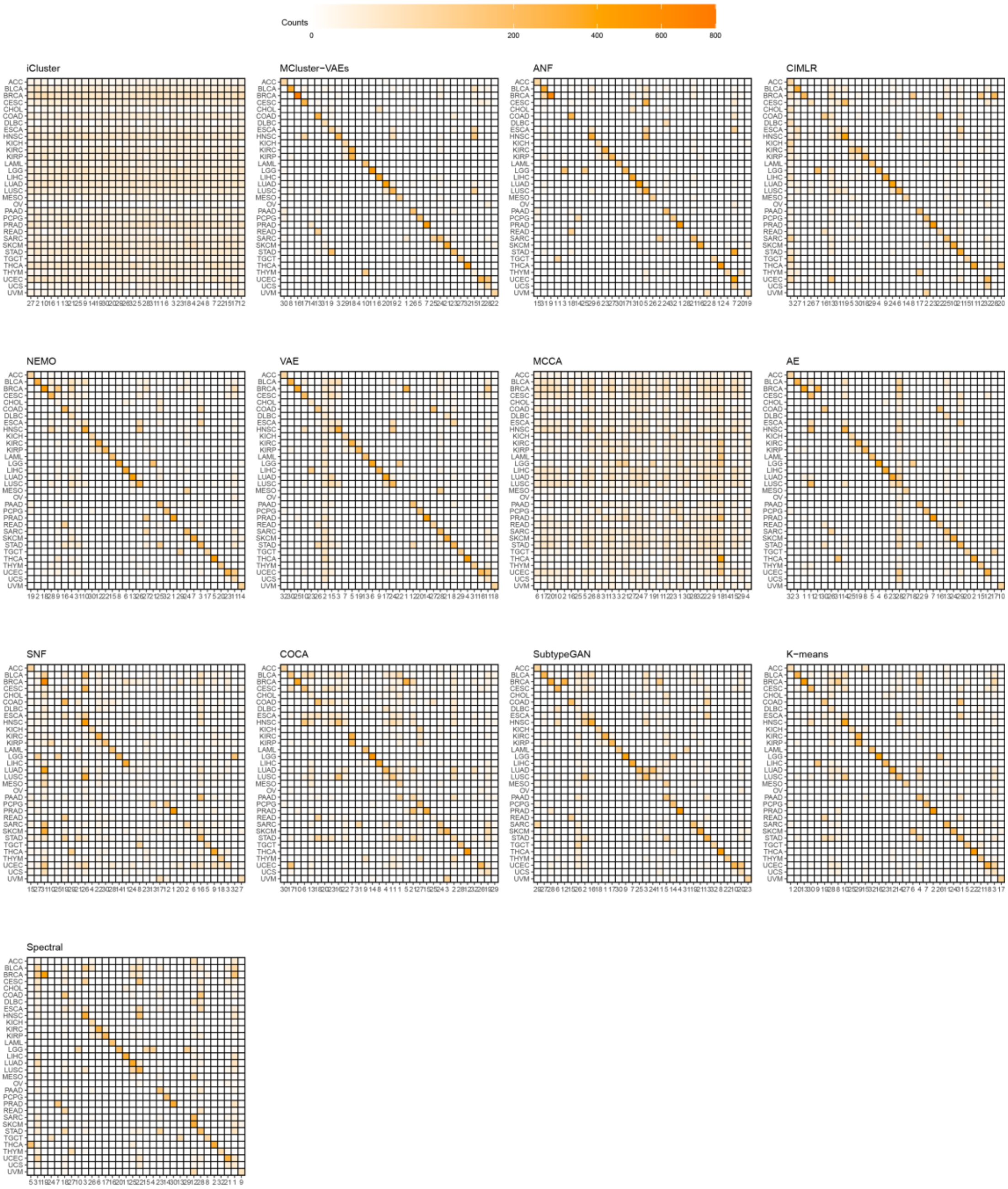

Figure S1. Heatmaps of confusion matrix of clustering performance using MCluster-VAEs and twelve other methods.

Figure S2: Sankey plots between clustering assignments and true cancer using MCluster-VAEs and twelve other methods

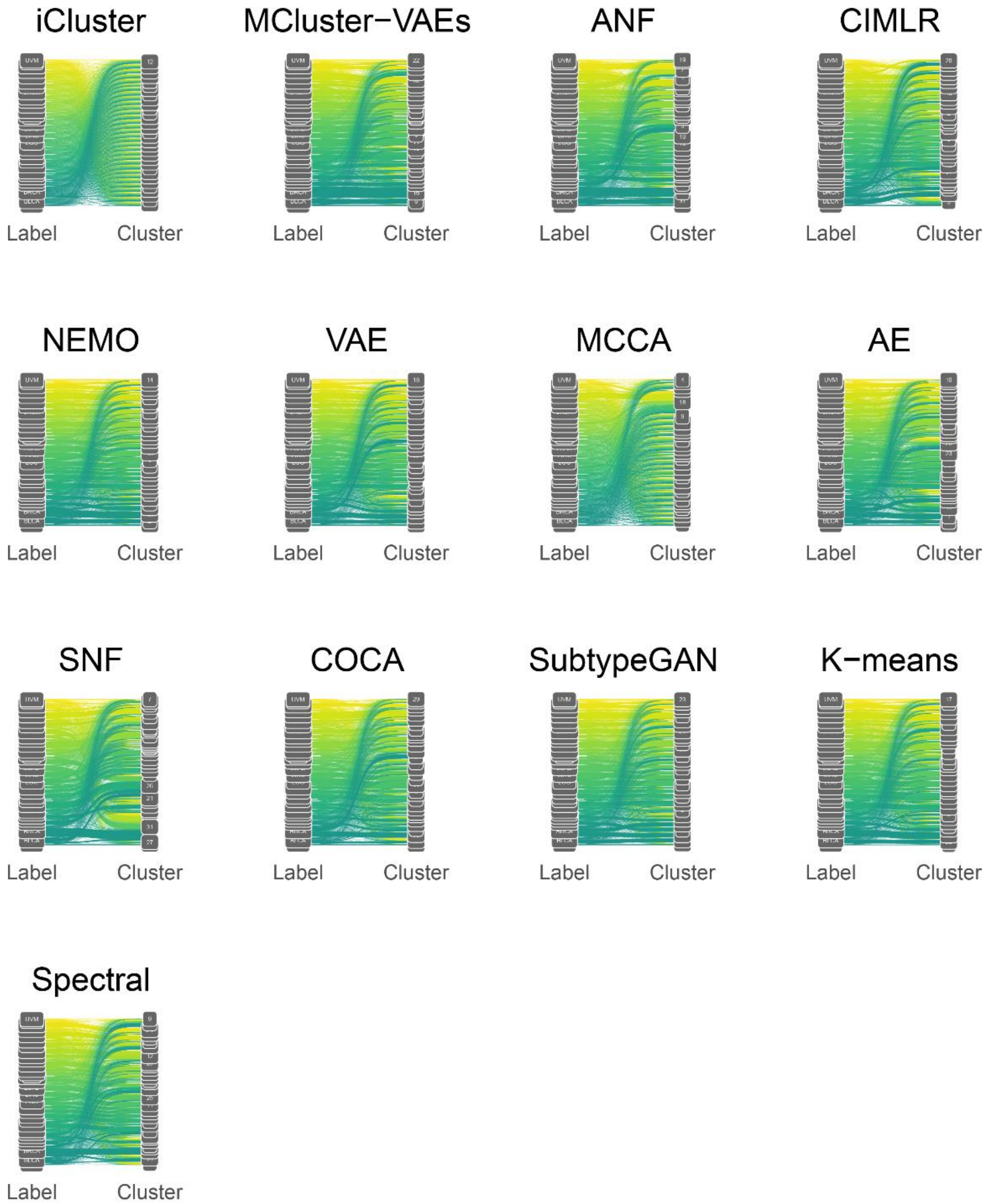

Figure S2. Sankey plots between clustering assignments and true cancer using MCluster-VAEs and twelve other methods.

Figure S3: Performance of MCluster-VAEs with (w/ attention) or without (w/o attention) attention mechanism on the Pan Cancer dataset

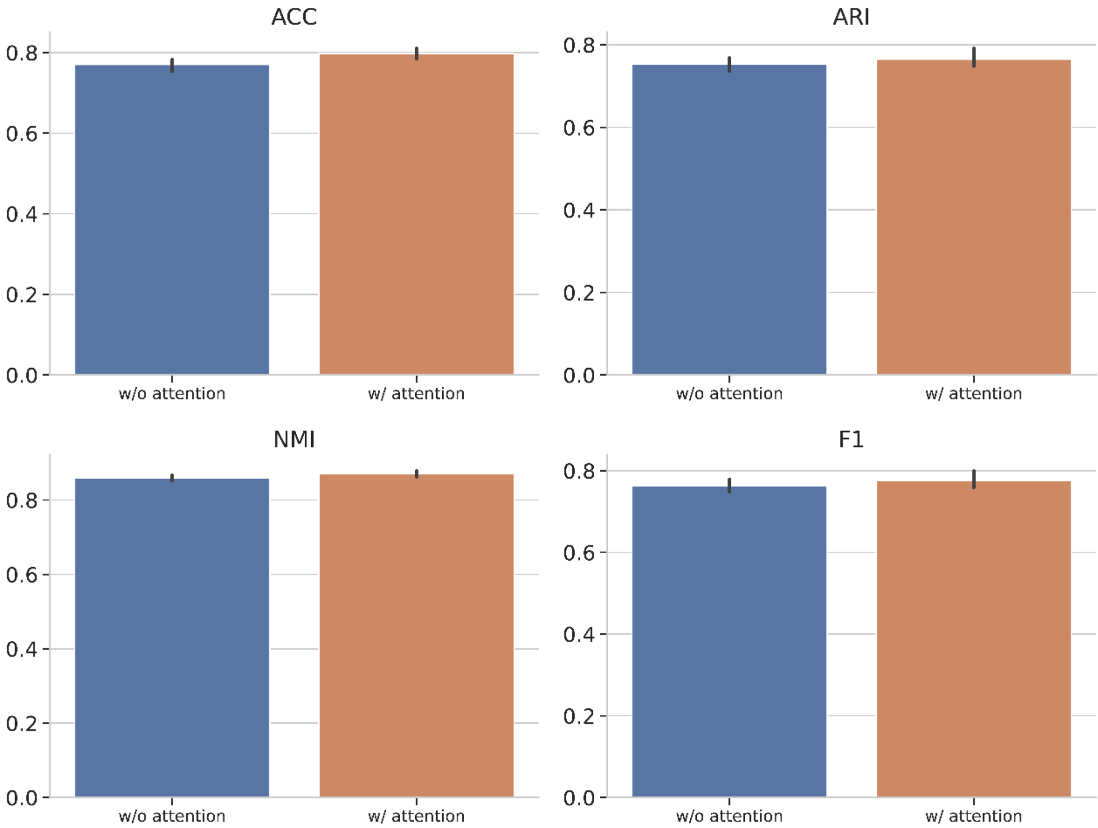

Figure S3. Performance of MCluster-VAEs with (w/ attention) or without (w/o attention) attention mechanism on the Pan Cancer dataset.

Figure S4: The distribution of attention scores of each cancer type for each omics data

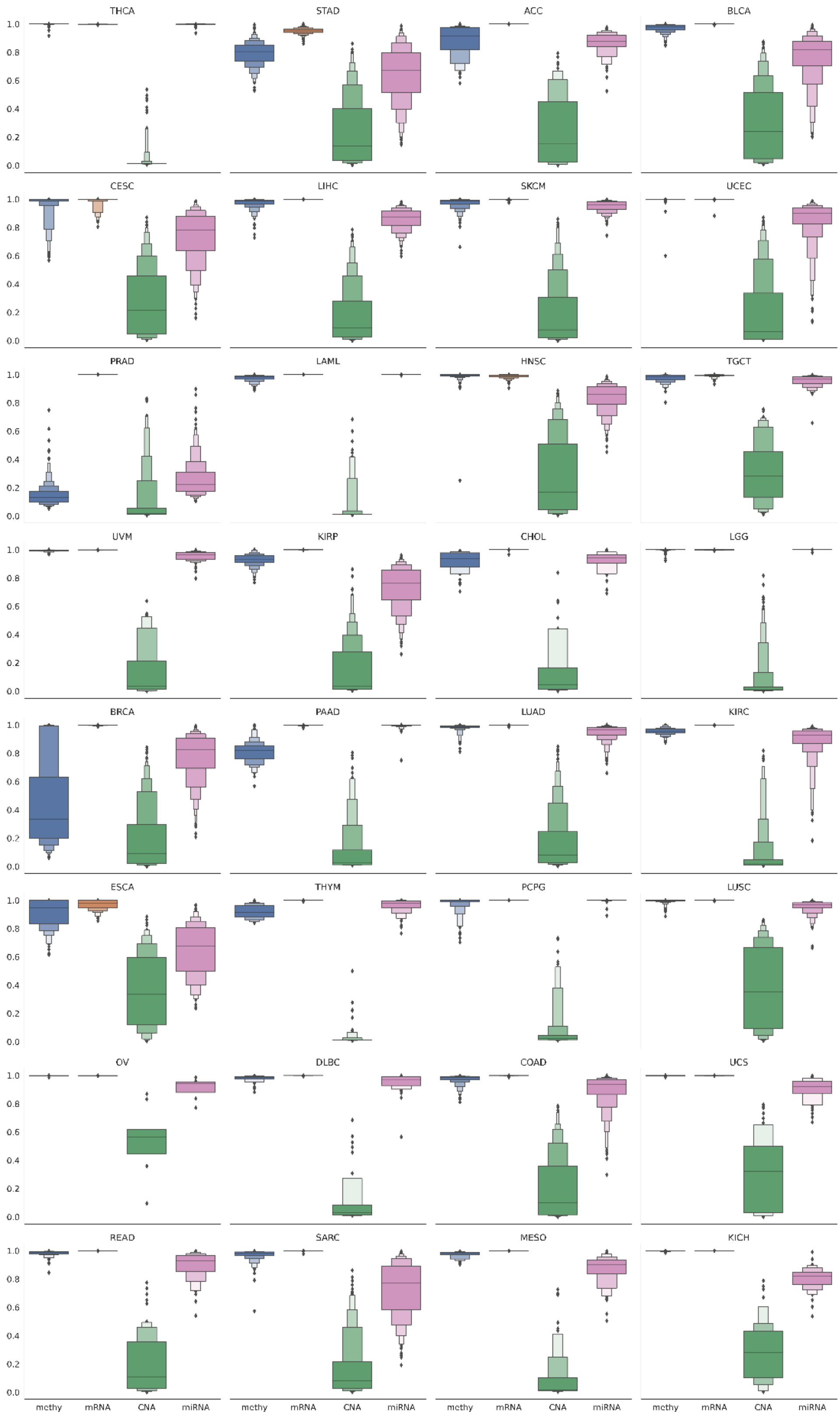

Figure S4. The distribution of attention scores of each cancer type for each omics data.

Figure S5: Running time of MCluster-VAEs with or without gumbel softmax trick

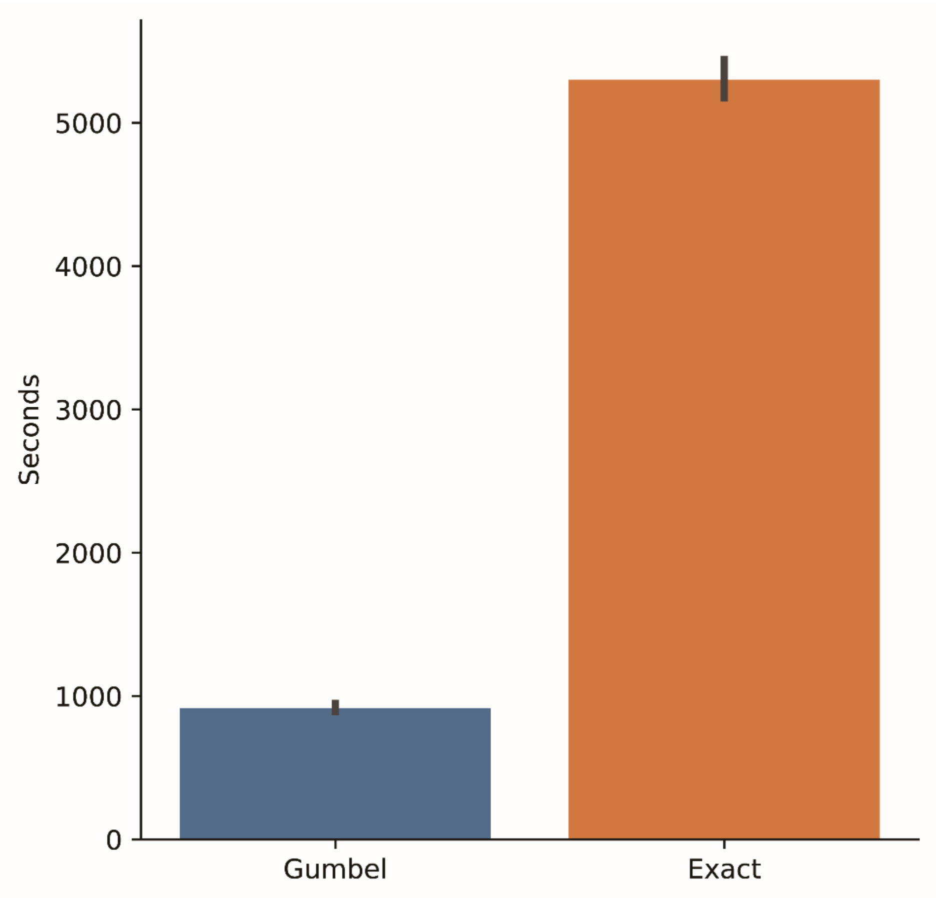

Figure S5. Running time of MCluster-VAEs with (Gumbel) or without (Exact) gumbel softmax trick on the Pan Cancer dataset.

Figure S6: Performance of the algorithms on the ten cancer datasets

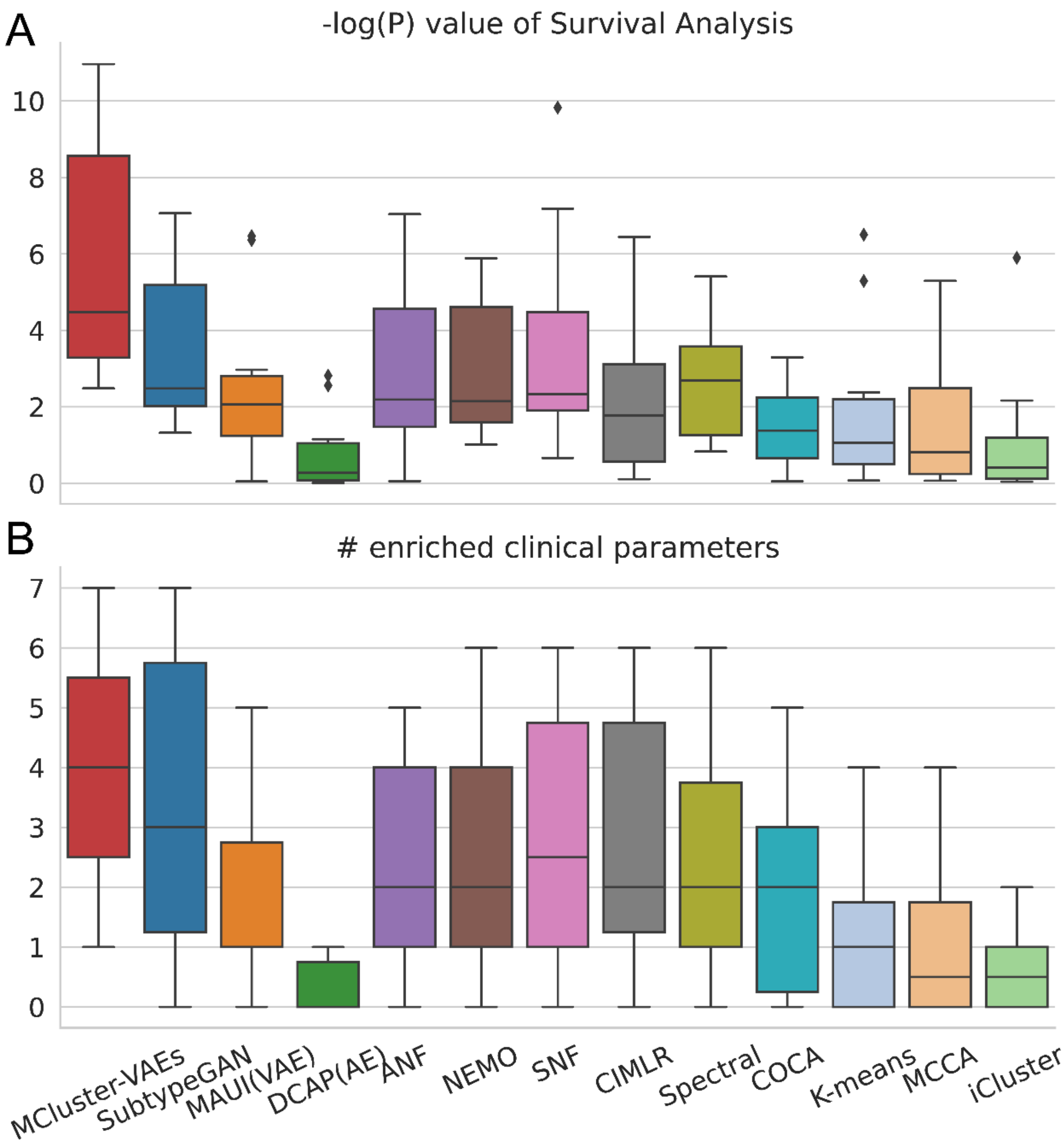

Figure S6. Performance of the algorithms on the ten cancer datasets. The x-axis was the multi-omics clustering methods used. The y-axis of first subfigure (A) measures the differen-tial survival between clusters (-log10 of permutated logrank's test P values), and the y-axis of the second (B) is the number of clinical parameters enriched in the clusters.

Figure S7: Performance of MCluster-VAEs based on four omics or single-omics data on the ten cancer datasets

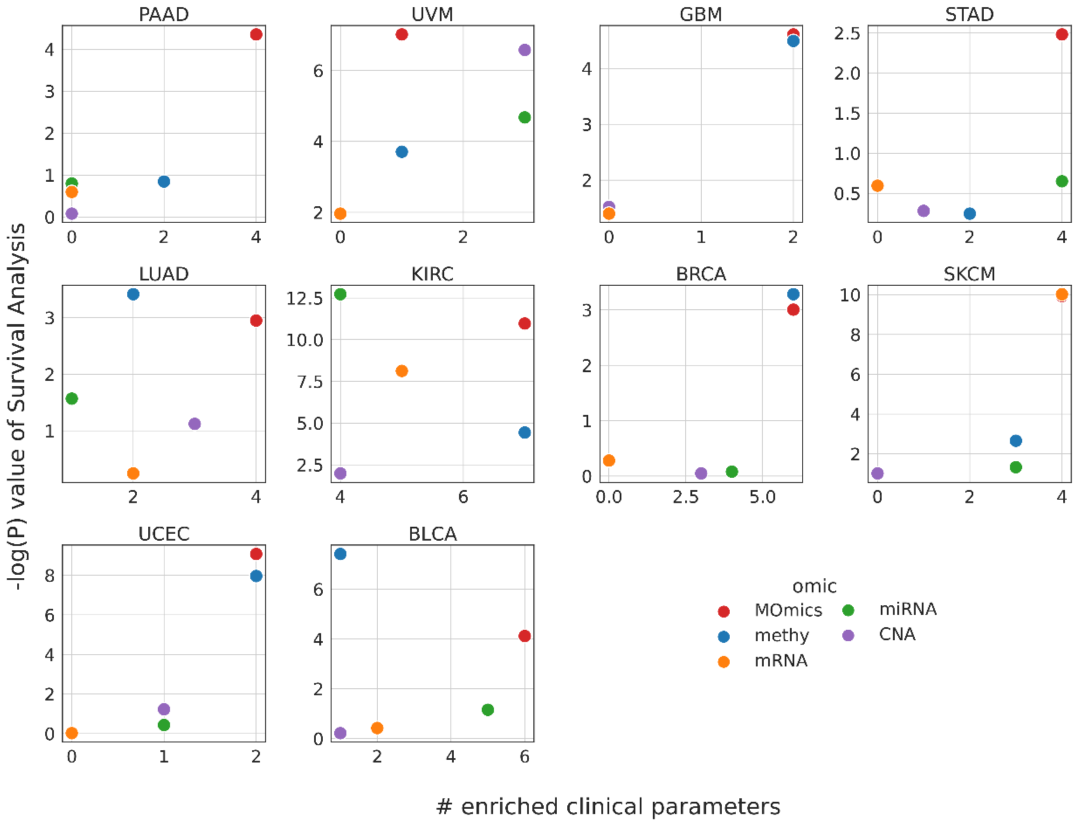

Figure S7. Performance of MCluster-VAEs based on four omics or single-omics data on the ten cancer datasets. The x-axis was the number of clinical parameters enriched in the clusters, and the y-axis measured the differential survival between clusters (-log10 of permutated logrank's test P values). Colors indicated the omics data applied. Here, MOmics represents four omics data, mRNA represents mRNA expression, methy denotes DNA methylation (450K), miRNA represents miRNA expression and CNA represents copy number alterations.

Figure S8: Silhouette scores MCluster-VAEs achieved based on different number of clusters on ten cancer datasets

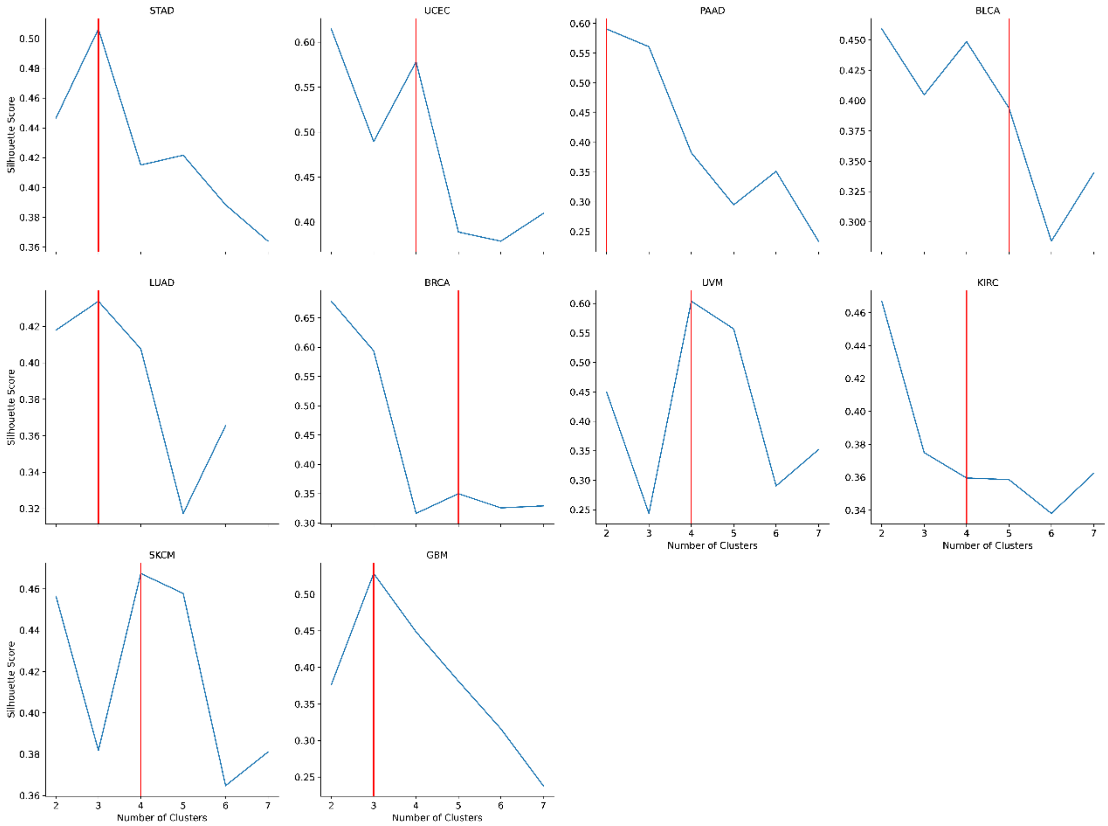

Figure S8. Silhouette scores MCluster-VAEs achieved based on different number of clusters on ten cancer datasets. The red line denotes the number of clusters obtained from previous large-scale studies for each tumor type.

Reference

1. Munkres J. Algorithms for the Assignment and Transportation Problems. Journal of the Society for Industrial and Applied Mathematics. 1957;5(1):32–8.

2. Pfitzner D, Leibbrandt R, Powers D. Characterization and evaluation of similarity measures for pairs of clusterings. Knowl Inf Syst. 2009 Jun 1;19:361–94.

3. Hendrycks D, Gimpel K. Gaussian Error Linear Units (GELUs). arXiv e-prints. 2016;arXiv:1606.08415.

4. Loshchilov I, Hutter F. SGDR: Stochastic Gradient Descent with Warm Restarts. arXiv e-prints. 2016;arXiv:1608.03983.

5. MacQueen J, others. Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Oakland, CA, USA; 1967. p. 281–97.

6. Shi J, Malik J. Normalized cuts and image segmentation. 2000.

7. Witten DM, Tibshirani RJ. Extensions of sparse canonical correlation analysis with applications to genomic data. Stat Appl Genet Mol Biol. 2009;8(1):Article28.

8. Hoadley KA, Yau C, Wolf DM, Cherniack AD, Tamborero D, Ng S, et al. Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin. Cell. 2014 Aug 14;158(4):929–44.

9. Ma T, Zhang A. Integrate multi-omic data using affinity network fusion (ANF) for cancer patient clustering. In: 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). 2017. p. 398–403.

10. Wang B, Jiang J, Wang W, Zhou ZH, Tu Z. Unsupervised metric fusion by cross diffusion. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. 2012. p. 2997–3004.

11. Wang B, Mezlini AM, Demir F, Fiume M, Tu Z, Brudno M, et al. Similarity network fusion for aggregating data types on a genomic scale. Nature Methods. 2014;11(3):333–7.

12. 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 Oct 26;9(1):4453.

13. Rappoport N, Shamir R. NEMO: cancer subtyping by integration of partial multi-omic data. Bioinformatics. 2019 Sep 15;35(18):3348–56.

14. Mo Q, Shen R, Guo C, Vannucci M, Chan KS, Hilsenbeck SG. A fully Bayesian latent variable model for integrative clustering analysis of multi-type omics data. Biostatistics. 2018 Jan 1;19(1):71–86.

15. Ronen J, Hayat S, Akalin A. Evaluation of colorectal cancer subtypes and cell lines using deep learning. Life Sci Alliance. 2019 Dec 1;2(6):e201900517.

16. Hira MT, Razzaque MA, Angione C, Scrivens J, Sawan S, Sarker M. Integrated multi-omics analysis of ovarian cancer using variational autoencoders. Scientific Reports. 2021;11(1):6265.

17. 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.

18. Chai H, Zhou X, Zhang Z, Rao J, Zhao H, Yang Y. Integrating multi-omics data through deep learning for accurate cancer prognosis prediction. Comput Biol Med. 2021 Jul;134:104481.

19. Yang H, Chen R, Li D, Wang Z. Subtype-GAN: a deep learning approach for integrative cancer subtyping of multi-omics data. Bioinformatics. 2021 Feb 18;

20. Rappoport N, Shamir R. Multi-omic and multi-view clustering algorithms: review and cancer benchmark. Nucleic Acids Research. 2018;46(20):10546–62.