Latent Factors of Multi-Omics Data and Clustering

Xia Anbang
Instituto Superior Técnico - Universidade de Lisboa

***Abstract*—Cancer is a very complex disease; often, its types cannot easily be classified simply by its location or manifested characteristics. In these circumstances, it is critical to find the group of patients who have similar underlying biological information and apply similar treatment for the person that belongs to the same group.**

The Cancer Genome Atlas (TCGA) database which focuses on cancer diseases with various types of omic data, such as mutations, RNA expressions and DNA methylation. Often, this data have more variables than samples which faces the problem of curse of dimensionality.

With this multi-omic data, this work aims to discover unknown factors common to the three data types using factor analysis tools such as iCluster and MOFA; this dimension reduction process can select more relevant information for further analysis.

This thesis proposes a methodology that compares MOFA and iCluster by finding the underlying latent factors and perform a clustering of patients who share biological similarities within the group.

After testing both methods, first on the synthetic data and comparing their abilities to recover the underlying factors and clusters, we decide to apply MOFA method for the Ovarian Carcinoma (OV) data extracted from TCGA, to find latent factors and the relevant clustering results.

From synthetic data analysis, we conclude that the MOFA has better performance. For real data, the genes find are cancer related but the cluster results are insignificant.

I. Introduction

The data explosion in recent years is bringing enormous opportunities to machine learning, allowing the development of new algorithms that can achieve surprising results in various areas. In the medical area, TCGA, a project that wants to catalogue genetic mutations responsible for mutations using genome sequence and bioinformatics is selected for this study.

When we use machine learning in medicine, instead like doctor approaches problems and finding solutions through constant learning and progressing during the career, it tries learning rules from data. Starting with patient-level observations, it tries to find algorithm to deal with vast number of variables, looking for combination that can predict the outcomes. Where machine learning shines is in handle with enormous number of features, remarkably in cases where predictor is more than observations which is called "curse of dimensionality" [1], and needs to be dealt by combining in non-linear and highly iterative ways [2].

An important technique to make machine learning algorithms to having feasible results is reducing the number of features. In general there exists two types of dimension reduction technique. First type, which reduce the feature numbers by selecting the most relevant features that have influence in the result. The representative of this type of technique is when we use, for example optimization functions as LASSO [3] and Elastic Net[4] penalties in regression models. Another type consist using mathematical techniques that transform the original features space into new subspace with fewer dimensions. Principal Component Analysis and Factor Analysis are most used technique in this family, and this work is focused on this one. When applying these methods to analyze cancer, we are assuming that the development of cancer on the human body is influenced by some genes.

Finally, with reduced data of gene sequences, we can cluster the patients into different clusters in way that the patients from same clustering are similar and different from inter-clusters. The grouping of patients is useful for medical treatment, because the similar patients may be treated efficiently by same treatment, and it is possible to separate the cancers by their intrinsic characteristics in various ways.

The specific purpose is to use Multi-Omic Factor Analysis (MOFA) [5] and iCluster[6] to reduce the dimension of data then apply clustering algorithm to latent space. First of all, we apply them to the generated synthetic data that has characterization of multi-omic biological data, analyze their results and interpretability for testing the viability and performance of framework. Then we apply the framework to the OV data from TCGA for getting the latent factors. After obtain the latent factors, we use the K-means algorithms to grouping the patients.

This document is organized as follows:

II introduce the theoretical basis of dimension reduction techniques.

III introduces K-means and cluster evaluation index.

IV introduces the methodology used, which explains the data used, how to process the data, data codification, the way to use frameworks and how to compare both integrated frameworks and pipeline of methodology.

V shows the results obtained by using synthetic data, analysing the results and finally apply the methodology to real data.

VI and VII describes the achievements obtained by this work and the possible future works.

II. Dimension Reduction

When we have a set of data with large number of dimensions, it is often desired to reduce the dimension, simplifying the data-set as much as possible while keeping the original information.

A. Principal Component Analysis

Consider a set of observed data matrix $Y \in \mathbb{R}^{n \times d}$, with n samples and d features. The objective is reducing the original

data dimension d into lower dimension k, such that $k < d$. PCA approximates the original matrix Y by component weights W and principal components Z,

$$Y \approx ZW^T. \tag{1}$$

Where $Z \in \mathbb{R}^{n\times k}$ and $W \in \mathbb{R}^{d\times k}$.

B. Factor Analysis

Factor analysis is a model that seeks to relate a d-dimensional observation vector y to corresponding k-dimensional vector of latent variables z, represented by following linear relationship:

$$y = Wz + \mu + \epsilon \tag{2}$$

W is the weight matrix that relates the observed set with hidden set. The parameter μ allows the model to have noon-zero mean, and ϵ represents the residual error. Conventionally the latent variables, $z \approx$ N(0,I) and defined to be independent and Gaussian with unit variance. By defining the error or noise term ϵ to be likewise Gaussian $\epsilon \approx$ N(0,Ψ), induces the corresponding observations y to be Gaussian $y \approx$ N(μ, $WW^T + \Psi$). The parameters are determined by Maximum Likelihood (ML) algorithm, because there is no closed-form analytic solution for finding W and Ψ.

C. Probabilistic Principal Component Analysis

The normal Principal Component Analysis (PCA) formulation presented is based on the linear projection of original data onto lower dimensional subspace data. PCA also can be expressed in the ML solution of a probabilistic latent variable model named as Probabilistic Component Analysis (PPCA) [7] proposed by M. Tipping and C. Bishop at 1999, which is a special case of latent variable model resulting when using the isotropic Gaussian noise model in the ϵ term present in the equation 2 $\epsilon \sim$ N(0, $\sigma^2 I$).

Consider a prior distribution over latent variable z given by zero-means with unit variance,

$$p(z) = N(z|0, I). \tag{3}$$

And the conditional distribution for the observed variable y conditioned on the value of the latent variable as:

$$p(y|z) = N(y|Wz + \mu, \sigma^2 I), \tag{4}$$

where the mean of y is a linear combination of matrix W with matrix of samples in latent space added with vector μ, and the variance is giving by $\sigma^2 I$. With this assumption, the observed values y are mapped by 2 with $\epsilon = \sigma^2 I$. Note that, in this framework the variables are mapped from the latent space to the observed space, the reverse mapping is obtained by using Bayes's Theorem to find the latent variables z.

We need to determine the values of the parameters W, μ and σ^2 first by using ML. To write down the likelihood function, we need the marginal distribution of the observed data y which is obtained by integrating out the latent variables:

$$p(y) = \int_z p(y|z)p(z)dz, \tag{5}$$

which is also Gaussian like, and it's given by,

$$p(y) = N(\mu, C). \tag{6}$$

where the observation co-variance model is specified by $C = WW^T + \sigma^2 I$.

And the posterior distribution $p(z|y)$ of the latent variables giving the observation y. Which can be calculated using the Bayes's rule, IT is given by:

$$p(z|y) = \mathcal{N}(z|M^-1W^T(y - \mu), \sigma^{-2}M^{-1}). \tag{7}$$

The Parameters can be find through ML or by Estimation Maximization (EM).

III. Cluster

Clustering tries to separates the objects from data-set to various non overlapping subsets, each subset is denominated by cluster.

A. Clustering Algorithm

This section will introduce the most used clustering algorithm in this work, namely K-means.

1) K-means: Giving a data-set $D = \{x_1, x_2,...,x_m\}$, the K-means algorithm will cluster the samples into k clusters $C = \{C_1, C_2,...,C_k\}$, the objective is minimizing the within-cluster sum of squares. Formally is to find:

$$\arg\min_c \sum_{i=1}^{k} \sum_{x\in C_i} ||x - \mu_i||_2^2, \tag{8}$$

which $\mu_i = \frac{1}{|C_i|}\sum_{x\in C_i} x$ is mean of points in C_i. The formula 8 are minimizing the sum square distance of all points to each mean point of each cluster, the lowest the sum the more similar are samples in cluster C. To find the global minimum in this problem is difficult, it need take consideration to all samples in the cluster C, this is a NP hard problem [8]. However there are efficient heuristic, greedy, dynamic programming algorithms to find the local minimum quickly.

B. Cluster evaluation

Considering $D = \{x_1, x_2,...,x_m\}$, suppose from clustering we got $C = \{C_1, C_2,...,C_k\}$, and the comparison model with $C^* = \{C_1^*, C_2^*,...,C_k^*\}$, and corresponding obtained label λ and λ^*. We can pair-wise all the configuration, and defining

$$a = |SS|, SS = \{(x_i, x_j)|\lambda_i = \lambda_j, \lambda_i^* = \lambda_j*, i < j\}, \tag{9}$$

$$b = |SD|, SD = \{(x_i, x_j)|\lambda_i = \lambda_j, \lambda_i^* \neq \lambda_j*, i < j\}, \tag{10}$$

$$c = |DS|, DS = \{(x_i, x_j)|\lambda_i \neq \lambda_j, \lambda_i^* = \lambda_j*, i < j\}, \tag{11}$$

$$a = |DD|, DD = \{(x_i, x_j)|\lambda_i \neq \lambda_j, \lambda_i^* \neq \lambda_j*, i < j\}. \tag{12}$$

In the set SS containing the samples that are classified to same cluster in C and same also belongs to cluster in C^*, in

the set SD containing the samples that are classified to same cluster in C and different cluster in C^*, DS containing the samples that are classified to different cluster in C and same cluster in C^* and DD containing the samples that are not classified to same cluster in C neither in C^*.

From equations 9, 10, 11 and 12 can infer the following index.

1) Precision and Recall:

$$P = \frac{a}{a+c} \tag{13}$$

$$R = \frac{a}{a+b} \tag{14}$$

Once we have precision and recall, we can infer F-measure coefficent, which is the harmonic mean of the precision and recall, defined as:

$$F = \frac{2}{\frac{1}{P}+\frac{1}{r}} = \frac{2PR}{P+R} = \frac{2a}{2a+b+c} \tag{15}$$

2) Jaccard Coefficient(JC):

$$JC = \frac{a}{a+b+c}, \tag{16}$$

3) Rand index:

$$RI = \frac{2(a+d)}{m(m-1)} = \frac{a+d}{a+b+c+d}. \tag{17}$$

4) Fowlkes and Mallows Index:

$$FMI = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}}. \tag{18}$$

All above index are valued between 0 to 1. The greater the better are the result.

IV. Proposed Methodology

The methodology used for this project is present in this chapter. First we will introduce the integrated frameworks used. Then the data used.

A. Integrated frameworks

The integrated frameworks used was factor analysis based and able to deal with multiple heterogeneous data, able to infer a set of hidden factors that capture source of variation across multiple data-types and having extra functionalities as clustering and factor analysis.

1) iCluster: Ronglai Sheng [6] has proposed in his original paper in 2009 a framework that includes latent variable model with clustering, the result method is called iCluster. iCluster incorporates flexible modeling of the associations between different data types and the variance-covariance with data types, while simultaneously reducing the dimension of data-sets and clustering in a single model.

a) Clustering method - Eigengene K-means algorithm: The standard K-means algorithm is introduced in chapter III with expression described in 19, which is sensitive to the choice of starting point and might iterate to the local minima rather than to the global minimum. A better optimization scheme for K-means arises from PCA, proposed by Zha et al [9]. Let $Z = (z_1, ..., z_k)^T$ with k-th row being indicator vector of cluster k and normalized to have unit length:

$$\arg\min_{c} \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||_2^2, \tag{19}$$

$$z_k^T = (0, ...0, \frac{1}{\sqrt{n_k}}, ..., \frac{1}{\sqrt{n_k}}, 0, ..., 0), \tag{20}$$

where n_k is the number of samples in cluster k and $\sum_{k=1}^{K} n_k = n$. With cluster assign matrix Z, the optimal of clustering solution can achieved by minimizing the within-cluster variance. Let XX^T be the Gram matrix of the sample, then the K-means loss function in 19 can be expressed as:

$$\text{trace}(XX^T) - \text{trace}(ZX^TXZ^T), \tag{21}$$

which is the total variance minus the between-cluster variance. The total variance is constant given the data, the problem in 19 is equivalent to maximizing the between-cluster variance:

$$\max_{ZZ^T = I_k} \text{trace}(ZX^TXZ^T). \tag{22}$$

Considering a continuous $Z^\star$ that satisfies all the conditions of Z except can take values different of zero or one. Then the problem is equivalent to the eigenvalue decomposition of S. Then the closed-form solution for 22 is $\widehat{Z}^\star = E$, where $E = (e_1, ..., e_k)^T$ are the eigenvectors corresponding to the K largest eigenvalues from the eigendecomposition of S. Ding and He also published in [10] that the redundancy in Z such that the K-means solution can be defined by the first $K-1$ eigenvectors.

Although defining the Z as continuous makes some problem in interpretability of the cluster indicator matrix, but it is necessary to achieve for find closed-form solution for K-means. Since we are in genomic data context, the algorithm described are named as eigengene K-means[6].

b) Gaussian latent variable model: Consider again a Gaussian latent variable model representation of the eigenvalue K-means clustering:

$$Y = WZ + \epsilon. \tag{23}$$

The formulation is similar than 2, where Y is mean-centered in this case. Differently from normal formulation, $Z = (z_1, ..., z_k)^T$ is the cluster indicator matrix of dimension $(K-1) \times n$ as previous defined. W is the coefficient matrix of dimension $p \times (K-1)$, and ϵ is a error term with zero mean and diagonal covariance matrix Ψ = diag(Ψ_1,...,Ψ_p). Considering a continuous parameterization $Z^\star$ of Z and additional assumption that $Z^\star \sim$ N(0,I) and $\epsilon \sim$ N(0,Ψ). Then the K-means problem with likelihood-based solution is available through model (23). The parameters inference will be based on the posterior mean of $Z^\star$ given the data and the inference method are already presented in the section when introduce PPCA.

c) Joint latent variable model: Since the objective is study biological genomic data, there exist more than one type

of data to explain the under-layer disease and sub-types of patient. So we need generalize the formulation present in 23 to multi-omic formulation. This means estimating matrix of indicator $Z = (z_1, ..., z_{K-1})$ by,

$$\begin{aligned} Y_1 &= W_1 Z + \epsilon_1 \\ &\cdot \\ &\cdot \\ Y_m &= W_m Z + \epsilon_m, \end{aligned}$$

where m is the number of genomic data type available. $Z \sim$ N(0,I) is common for all data types, $\epsilon_i \sim$ N(0,Ψ) is error term that is independent from each other and W_i is a coefficient matrix. Then the marginal distribution data matrix $Y_1, ..., Y_m$ are multivariate normal with mean zero and covariance matrix $C = WW^T + \Psi$, the corresponding log-likelihood function is defined by,

$$\mathcal{L} = -\frac{N}{2} D \log(2\pi) + \log|C| + Tr(C^{-1}S). \tag{24}$$

The parameter inference is obtained by EM algorithm to obtain the maximum likelihood estimates of W and Ψ, dealing with complete-data log-likelihood, which is more efficient than maximizing directly the marginal data likelihood.

d) A sparse solution: The framework is applied to the biological data sets, which normally the number of features p are much bigger than number of samples n. In this cases, the regularization method is important. The sparse solution is applied to W, and the complete data-likelihood with sparse solution is,

$$l_{c,p}(W, \Psi) = l_c(W, \Psi) - J_\lambda(W), \tag{25}$$

where J_λ is a penalty term on W with non negative regularization parameter λ. Authors of iCluster adopted a lasso penalty, which takes the form,

$$J_\lambda(W) = \lambda \sum_{i=1}^{m} \sum_{k=1}^{K-1} \sum_{j=1}^{p_i} |w_{ikj}|. \tag{26}$$

2) MOFA: MOFA [5] was proposed by R. Argelaguet et al. at 2018, a computational method for discovering the principal sources of variation in multi-omics data-set. This method can infer a set of hidden factors across various types of measures. The result learnt factors can be used for downstream analysis.

a) Method: Considering we have M data matrices Y^1,..,Y^M of dimensions $N \times D_m$, where N is the number of samples and D_m the number of features in data matrix m. The objective of MOFA is try find hidden factor matrix Z such that,

$$Y^m = ZW^{mT} + \epsilon^m \tag{27}$$

where m = 1,..,M, W^m denotes weight matrix for each data matrix m and ϵ^m denotes error term for each data matrix, it's depend on data type. MOFA is formulated in the probabilistic Bayesian framework, proposing prior distributions on all unobserved variables.

Starting by assuming ϵ^m to be Gaussian like, while allowing heteroscedasticity across features, we get:

$$p(\epsilon_d^m) = N(\epsilon_d^m | 0, 1/\tau_d^m). \tag{28}$$

With previous assumptions, we get:

$$p(y_{nd}^m) = N(y_{nd}^m | z_{n,:} w_{d,:}^{mT}, 1/r_d^m) \tag{29}$$

where $w_{d,:}^m$ denotes the d-th row of the loading matrix W^m and $z_{n,:}$ the n-th row of the latent factor matrix Z. For probabilistic treatments, MOFA assumes prior distributions for the weights W^m, the latent factors Z and error term τ^m. It assumes standard Gaussian prior for latent factors and conjugate Gamma for error:

$$p(z_{n,k}) = N(z_{n,k} | 0, 1) \tag{30}$$

$$p(\tau_d^m) = \Gamma(\tau_d^m | a_0^\tau, b_0^\tau) \tag{31}$$

b) Model Regularization: MOFA having two types of regularization on weights W^m: a view- and factor-wise sparsity and a feature-wise sparsity. The view- and factor-wise allows to directly identify which factor is active in which view, and feature-wise sparsity puts zero weights on individual features from active factors.

This regularization is achieved by putting appropriate priors on the weight matrices. MOFA uses Automatic Relevance Determination (ARD) prior for view- and factor-wise sparsity and spike-and-slab prior for feature-wise sparsity.

The spike-and-slab prior contains Dirac delta functions,

$$p(\omega) = (1 - \theta)\delta(\omega) + \theta N(\omega | 0, 1/\alpha) \tag{32}$$

which is problematic for inference. For solving this problem is required to re-parameterize weights w as a product of a Gaussian random variable $\hat{w}$ with Bernoulli random variable s, resulting following prior:

$$p(\hat{w}_{d,k}^m, s_{d,k}^m) = N(\hat{w}_{d,k}^m | 0, 1/\alpha_k^m) Ber(s_{d,k}^m | \theta_k^m) \tag{33}$$

which α_k^m controls strength of factor k in view m and θ_k^m controls contribution of spike term affecting feature-wise sparsity of factor k in view m. For automatically learning these parameters, it assumes following priors:

$$p(\theta_k^m) = Beta(\theta_k^m | a_0^\theta, b_o^\theta) \tag{34}$$

$$p(\alpha_k^m) = \Gamma(\alpha_k^m | a_0^\alpha, b_0^\alpha) \tag{35}$$

with hyper-parameters a_0^θ and b_0^θ = 1 and a_0^α, b_0^α = $1e^{-14}$ to get uninformative priors. A value of θ_k^m close to 0 implies that most of the weights of factor k in view m is shrunk to 0.

c) Parameter Inference: For inference of parameters, MOFA use a variational Bayesian framework, which is

essentially a mean field approximation for approximate inference[11]. The key idea is to approximate the intractable posterior distribution using a simpler class of distributions by minimizing the Kullback–Leibler divergence to the exact posterior, or equivalently maximizing the Evidence Lower Bound (ELBO). Convergence of the algorithm can be monitored based on the ELBO.

More details can be found on [12].

B. Data

Before use introduced frameworks on real data, was generated synthetic data for validating and testing the viability of methods.

1) Synthetic data: Once we want to group patients into subcategories based on the reduced features, the validation data-set should also have this characteristic. The generation of synthetic data are proceeded by the following steps. First of all, is generated a latent variable matrix Z using python sklearn package[13], with predefined number of clusters, samples and features. Then is transformed into multi-omic and higher dimensional space according to the number of views and features selected. Then is added the error term.

a) Toy example: We use the following toy example to explain in detail the synthetic data generation. Using sklearn package to generate 50 samples with 2 latent factors distributed along 4 centers with standard deviation equals to 1.

Then is applied a transformation to Z such that the original data is transformed into a Gaussian 3-view data set, each view with 20 features as illustrated in Figure 1.

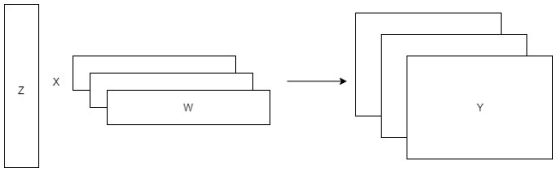


Figure 1: Transformation of synthetic data in latent space to multi-omics space.

The transformation is done by generating 3 matrix of W with random normal distribution $N \sim \text{N}(0, \frac{1}{\sqrt{\alpha}})$, with α as a random kxn matrix with value 1 or 1000. For simulating a sparse solution the weight matrix W is multiplied by a $S \sim B(p = 0, 5)$ matrix to simulates that a latent factor are affect only by some features. Depending which type of distribution is needed, is applying a transformation to ZW and adding the error term.

b) Set of synthetic data: For more general view, will create a big set of synthetic data with different values of parameters for testing the performance of frameworks according to the dimension size of parameters. The synthetic data for this purpose will have following parameters depending on Z or Y. For the Z, we need to consider the number of test samples, number of latent factors, number of cluster and number of cluster's standard derivation values. And for simulate Y, it need to take account of number of samples, number of visible features, number of latent variables, number of clusters and number of data types. The works turns too complex if we test the performance changing all the mentioned parameters, so for simplicity we only choose some of them that was more important for testing, and let the others to be constant.

There was generated 1280 test samples for evaluating the performance, the parameters values and combination of parameters is illustrated in Figure 2.

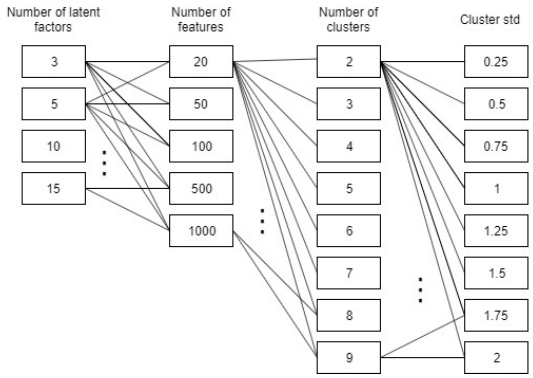


Figure 2: Values and 1280 combinations of testing sample parameters.

2) Real biological data: For this study, are only selected DNA methylation, RNA sequence and mutation types of OV cancer data from TCGA.

a) Data pre-processing: In the RNA-seq data-set, there exist two types of cancer, primary with 374 patients and recurrent with 5 patients, for simplicity this 5 patients samples are ignored. In the set of 56830 genes, only are selected subset of 19941 variables corresponding to the protein-coding genes reported from Ensemble genome browser [14] and the Consensus CDS Project [15]. For simplicity is only selected the intersection information across all data, resulting a set with 269 patients.

Then is used a variance filter to select the most variate features. Resulting 1500 features for RNA-seq, 1000 for DNA methylation and 500 for mutations.

C. Proposed pipeline

First was used the toy example for explain in detail the way was generated the synthetic data. Then the generated set of 1280 test samples will be used for analysing the MOFA and 768 test samples for iCluster, the samples with feature dimension 500 and 1000 wasn't tested in the iCluster due to training time problem.

After analysing the results of with synthetic data to get performance of frameworks, the best one will be used for OV data analysis. iCluster was actually used too, but due to high dimensional and software problem, the tests was failed. The result pipeline is illustrated in Figure 3.

V. Results

This part will present the results obtained. Chapter contains two sections, the first one is result from applying methodology

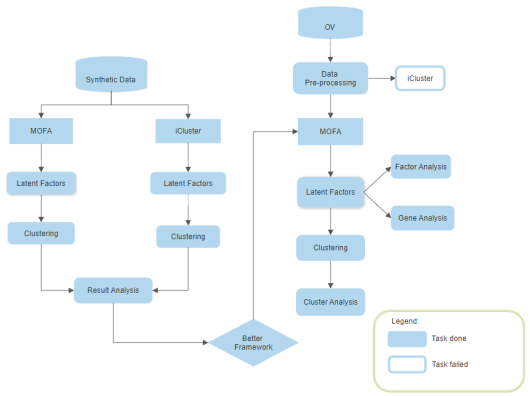


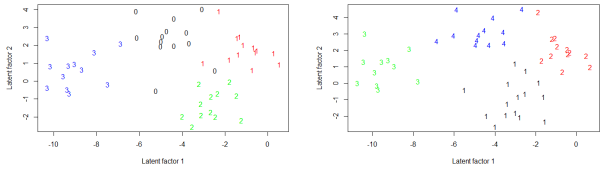
Figure 3: Flowchart of methodology.

to the synthetic data and the second one was used for real data-set.

A. *Synthetic data results*

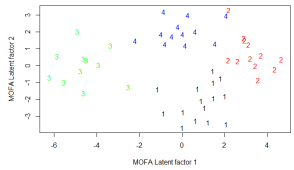
The synthetic data was used for validating the availability and performance of frameworks. This section presents the result obtained from Synthetic data.

1) Toy example results: The original cluster of toy example is presented in Figure 4a. Figure 4b shows the clusters resulted using K-means on original data, is possible to visualize that without any transformation the clusters result have wrong classification on some points. This was created on purpose for having average value of standard deviation of points to induce misclassification. Figure 4 shows the cluster points plotted in the latent space found by MOFA. It is possible to visualize that the recovered space is very similar that then original one, the cluster's position are also similar, only the points have some minor movements.



(a) Original cluster's label of data set.

(b) K-means cluster of original data set.



(c) K-means cluster of latent factors obtained by MOFA.

Figure 4: Cluster comparison with K-means algorithm in original data points and in latent space found by MOFA.

a) Using MOFA: For MOFA, was chosen default parameters to train the data set except the DropFactorThreshold, a parameter for drop out the factors that have low variance explained. Then run the framework 25 times for choosing the best model based on the ELBO value. For this example, was obtained 2 latent factors as illustrated in Figure 5, which $R2$ represents the variance captured by each factor and view represent data types. From analysing the variance explained per factor graphic, we can observe that majority of information present across 3 data types was captured only by 2 factors. From graphic, the view_3, which represents the data type 3 is almost explained only by the latent factor number 2(LF2), and view_1 and view_3 was captured by LF1. And from total variance explained graphic shows that the information of view_3 and view_1 was mostly recovered and the view_2 only retains around 25% of variance.

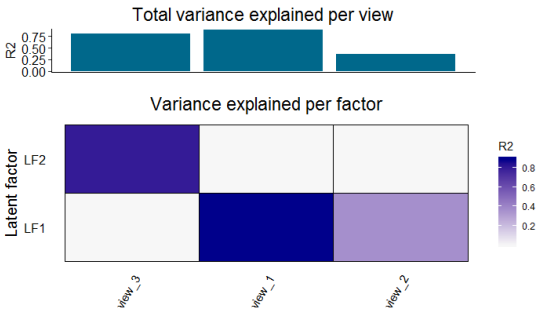


Figure 5: Latent factors obtained from MOFA.

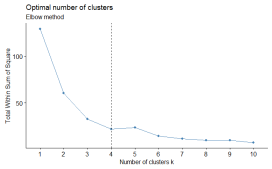
Then is applied K-means clustering to the obtained latent factors. For determining the best k for this data set, was used 3 different way as showed in Figure 6. In Figure 6a is illustrated the k selection using elbow method, that suggests 4 clusters. In Figure 6b is though Silhouette, that suggests 4 clusters and in Figure 6c is using Gap statistic that suggest 3 clusters.

In this case, 2 of 3 methodologies suggests $k = 4$, then for the cluster analysis was assumed to have 4 clusters. The plot of cluster samples are present in Figure 4c.

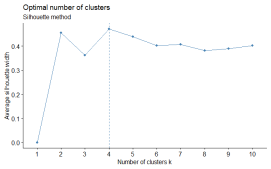
b) Using iCluster: iCluster allows to set number of latent factors and number of clusters k, because the framework is subject to have one less dimension in latent space (latent space with dimension k-1). And value of parameter set lambda(a parameter set to control how many genomics features have non-zero weights on the latent factor). We have to optimize these two parameters. The number of lambda points to sample depends on the number of data types, Toy example have 3 data types, was chose 185 lambda values for training the data.

For each k, the lambda selection criteria is based on Bayesian Information Criteria (BIC). To choose the best k, iCluster uses % explained variation. The result is illustrated in Figure 7, then is chosen $k = 3$ as the best value for the posterior analysis. By selecting 3 clusters, the number of latent factors are automatically fixed as 2.

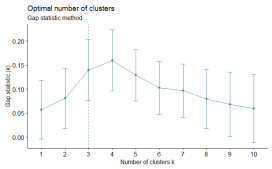
c) Cluster result evaluation: After getting cluster results of samples in MOFA's latent space and iCluster's latent space,



(a) k determination by Elbow method.



(b) k determination by Silhouette method.



(c) k determination by Gap statistic method.

Figure 6: Different methods for determining number of cluster k on latent factors obtained by using MOFA.

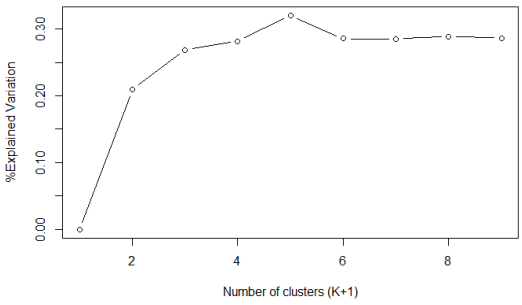


Figure 7: Number of cluster vs percent of variance explained.

it allows to evaluate the performance of both frameworks by comparing the real label and labels obtained. In Table I shows the values of index obtained.

Source of LF	*Precision*	*Recall*	*Jaccard*	*Rand*	*Folkes M.*
Original	0.84375	0.8321918	0.7210683	0.9232653	0.8379509
MOFA	0.8402778	0.8373702	0.7223881	0.9240816	0.8388228
iCluster	0.9583333	0.6216216	0.6052632	0.8530612	0.7718295

Table I: Cluster index of toy example.

The first row of table presents the index values obtained by comparing the real label with labels obtained by applying K-means in the original latent space.

The second entry refers index values obtained by comparing real labels with labels provided by K-means in MOFA's latent space.

And the last entry refers index values come from iCluster's latent space.

From visualizing the results on Figure 4 and Table I, MOFA present better results. Is able to recover majority of cluster labels, the samples localization on latent space and the evaluation index values are very close to values obtained by applying K-means directly to the original space. For the iCluster, the restriction of relation between number of cluster and latent space dimension makes the framework to have worse performance almost in all indexes except in precision.

2) Set of data with various parameters: Next are present results obtained from testing the 1280 synthetic samples.

a) Cluster's STD effect: First of all, we want to study how the cluster's Standard Deviation (STD) affects the model performance. And the result is showed in Figure 8. The figures shows how the 5 clustering index vary when the clustering standard deviation changes.

In Figure 8a shows the index values resulted by comparing the real labels with label obtained by applying K-means on original latent factors. For standard derivation less than 1.0, the problem is trivial on original space. Almost all index have high values, K-means can group the correct clusters. When STD increases, the misclassification error also increases due to the points of different groups are intercepting.

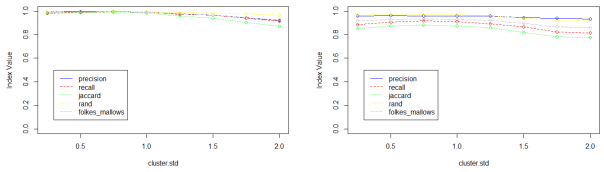
In second Figure 8b shows the index values come from the comparison between real labels of samples and labels resulted by K-means applied to the MOFA's latent factors with Silhouette as selection criteria for find k. The index values have best performance on STD values between 0.5-1.0. Theoretically, small STD values of clusters will get better index values, and this didn't happen may due to space transformations on data-set.

In Figure 8c is similar than previous one, but the k selection criteria is elbow method. From comparing this graphic with Figure 8b, the results are worse in almost all index except in precision. The index values doesn't change too much along different STD values. Beyond analysing the STD effect, We also can conclude that, for this type of data, the number of cluster obtained by Silhouette method is better than elbow method.

In Figure 8d are presented clustering index obtained by applying the iCluster framework. For this case, the samples used to obtain the result are less than others cases due to time cost problem for training medium and big size data. The index values decreases when the original data points dispersion increases.

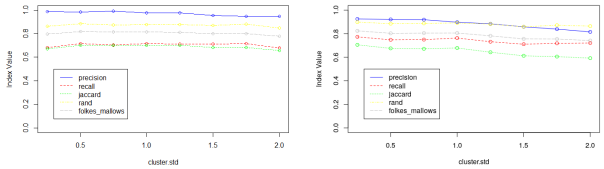
b) Original Latent Factor (LF) recovering: It was also tested the relation between the number of original latent factors with the number of features.

1) MOFA
   The result is illustrated in Figure 9 for MOFA, the x axis represents a group of test with 64 samples with same latent space dimension and same feature dimension. Graphics represents the distribution along recovered space of LF, which can be visualized that the tests with original latent space with dimension 3 and 5 have high density concentrated in 3 and 5 respectively.
2) iCluster
   The same analysis is applied to iCluster, the results are present in Figure 10. For the iCluster case, the sample's feature dimension only used are 20, 50 and 100.
   Differently than the MOFA's results, iCluster also have worse performance when the original latent space is reduced. For number of original LF equals to 3, the



(a) Index values using original factors.

(b) Index values using MOFA's factors by Silhouette.



(c) Index values using MOFA's factors by elbow.

(d) Index values using iCluster's factors by elbow.

Figure 8: The cluster's STD effect on index values.

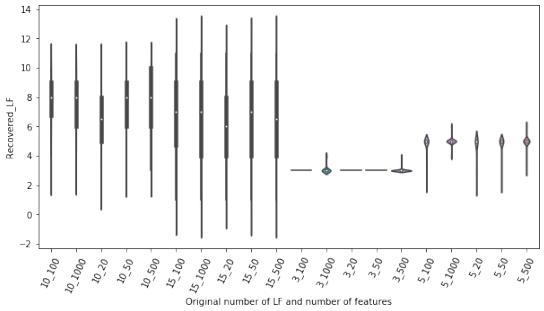


Figure 9: Effects of feature number on latent numbers with MOFA in graphic representation. In the x axis, the first element represents original #LF, and second element is dimension of samples in feature space.

results are distributed along various values, more than half of cases don't hit the correct solution, but general view, the solutions have precision. For other cases, the solution is very dispersed, the same conclusion can be visualized on the Figure 10, the density was concentrated on the values between 2-4.

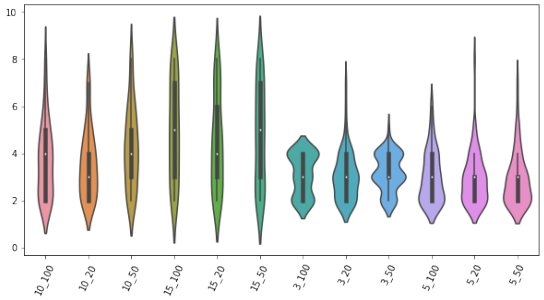


Figure 10: Effects of feature number for finding latent space dimension by iCluster in graphic representation.

c) Recovering number of cluster: Figure 11 presents the influence of number of factors in terms of recovering the number of clusters. For samples with original k equals 2, 3 and 4, MOFA can find exact number of cluster more than half of cases. From Figure 11 also can be visualized that the first elements have density concentrated in the correct number of k, after original number of k are greater or equal to 5, starts the dispersion of k found across various values.

The results point out that the MOFA have high accuracy and average precision for finding the exact number of clusters. The results also shows that the number of latent factors doesn't affects too much in the task for finding a correct number of clusters k. In the each entry was incorporated tests with different number of features, this means that this result ignores the influence of feature dimension in this task.

For the iCluster's case, the framework have the constrain in the problem formulation, that imposes the relation number of cluster k equal number of latent variable plus one.

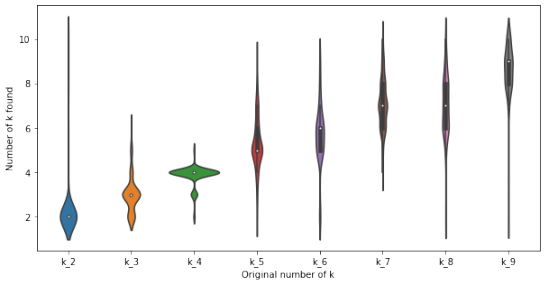


Figure 11: Graphic of experiment for testing the effect of number of latent factors in number of clusters using MOFA.

d) Time consumption: Table II presents approximate time interval used for training each dimension of data. For MOFA, it runs 10 random initialization and iCluster uses 135 lambda values with 10 initialization. For iCluster, the time consumption is large because the framework trains one model per each value of lambda and for each value of k, number of clusters.

Feature dimension	*iCluster*	*MOFA*
20	5-7min	$<$ 1min
50	15-45min	1-2min
100	1-3h	1-3min
500	5-20h	2-5min
1000	22-40h	15-25min

Table II: Training time consumption.

e) Result analysis: From the previous results, it shows that generally MOFA presents better results than iCluster. MOFA is able to apply higher dimensional data within acceptable time thanks to fewer parameters to test , to choose and better optimization algorithm, can get the latent factors and clusters separately.

B. Real biological data results

After testing these two frameworks with synthetic data, this part will use OV data from TCGA database for analysis. The data processing was already done, now we run the data with MOFA framework. For train the model, most of parameters are default one. There was used 25 latent factors to start, DropFactorThreshold as 0.02 for drop out the factors that explain less 0.02 variance in all data types. The maximum iterations for train the model are 4000 and the stop parameter ELBO difference value was set to 0.2. The best model of 25 random initialization was used for later analysis based on the ELBO value.

Latent factor analysis: With this data-set the best model have only 3 latent variables, the variance captured and explained by each factor are present in Figure 12.

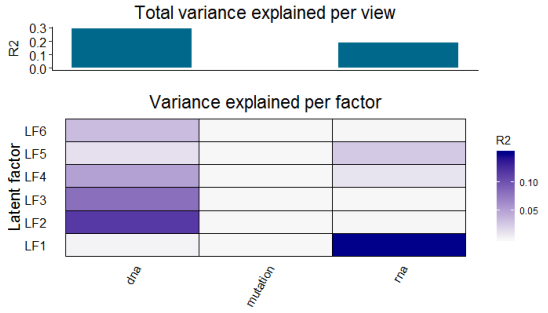
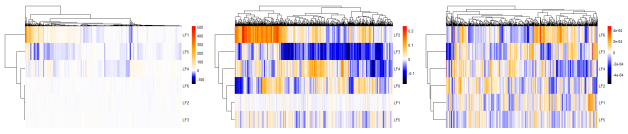


Figure 12: Variance explained by each factor.

From analysing the figure, we can see that the DNA methylation information is the most captured one, was retained about 0.30 variance of total variance, then the RNA-seq with approximately 0.20 and for mutation data is show that the variance captured is almost null. This is happens because the mutation data is a very sparse binomial matrix, in this case the way to calculate the variance captured is not very effective.

In detail, we can see that the DNA methylation information is mostly present in LF2, LF3, LF4 and LF5. RNA-seq information in LF1 and LF5.

After obtain the factors, we can do further analysis with them. Each factor are composed by linear combination of feature weights in sparse space which are plotted in Figure 13a.



(a) Heat-map plot of weights in mRNA. (b) Heat-map plot of weights in DNA. (c) Heat-map plot of weights in mutation.

Figure 13: Heat-map plot of weights in each type of data.

We can see from figure, that RNA-seq information is highly concentrated in few features, that makes the weight of each feature having big values comparing with other data. In DNA methylation, LF2 is positive correlated with a group of features and LF3 is composed by features with negative correlation. For mutation data, the weight matrix have very small weights and the features don't form a clear group.

Gene analysis: In order to understand the functionalities of each factor, there was selected top 10 features of LF1, LF2 and LF3 that are most important in each type of data in Table III. The RNA-seq data and mutation was converted from Ensemble ID to standard symbol for better interpretation.

Clustering: We also want to group the patient into clusters based on K-means in the latent space. The choice of number of cluster is based on the Silhouette method, and it suggest 5 clusters. This data don't have well defined labels for comparison, the survival analysis on the samples of each cluster was made for analysis, and is present in Figure 14. From observation, the figure shows that the cluster are not well separate between clusters.

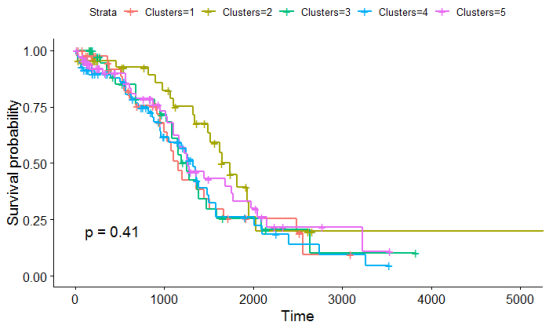


Figure 14: Survival plot with different clusters.

VI. Achievements

The goal of this work is to analyze the high dimensional bio-metrics data and to group the patients in various clusters that maybe share similar underlying biological information for later clinical treatments.

The proposed methodology uses iCluster and MOFA for finding the latent factors. The results of synthetic data demonstrate that the MOFA is more advantageous than iCluster in many ways. Also, MOFA has the ability for recovering the correct number of latent factors for small cases and it can recover approximately number of original clusters. And iCluster shows bad results in the parameters recovering in high dimensional data. From the time cost point of view, iCluster requires much more time for training the samples due to parameters numbers.

With OV data-set of TCGA, MOFA can infer few number of hidden factors, but due to high dimensional and complexity of data, the factors cannot capture too much variance from original data. Nevertheless the obtained factors can provide the most important genes that are composed by them. Moreover this work has selected some genes that possibly are important in the cancer disease, able to provide useful information for the specialists to analyze.

LF1			LF2			LF3		
mRNA	DNA m.	Mutation	mRNA	DNA m.	Mutation	mRNA	DNA m.	Mutation
TMSB10	cg03668539	TENM1	TIMP2	cg21312148	VCAN	WNT11	cg13603171	LRRC7
RPL18A	cg05406101	LAMA3	NDUFB4	cg20676475	CDH10	B4GALT1	cg25509184	TRIO
RPS12	cg00448720	WNK1	GNAS	cg12348970	KMT2C	KHSRP	cg22396755	DNAH5
RPL35	cg27462398	PTPRH	CCNE1	cg25856811	FAT1	PTN	cg18952647	SYNE2
RPLP1	cg15821095	MYH7	URI1	cg00152644	MYH1	MRPL37	cg07027513	KMT2C
RPS11	cg21591452	AHNAK	ATP6V0E1	cg05440289	TSHZ3	DPEP3	cg22881914	MDN1
RPL13A	cg15679651	HMCN1	CANX	cg21754343	FLG	MXRA8	cg09107315	FMN2
RPS8	cg16979445	OBSCN	MRPS12	cg07014174	ADAMTS19	HNRNPK	cg08575537	CSMD3
RPL8	cg01281904	TLN2	POMP	cg11750883	FAM135B	STOML2	cg09492887	PKHD1
RPL29	cg16773028	DNAH2	NHP2	cg24423088	FMN2	ZNF503	cg19308222	TLN2

Table III: Top-10 features on LF1, LF2 and LF3.

Furthermore, can be analyzed patients from same cluster, special in patients that belong to the cluster number 2 with others.

VII. Future Work

Normally, the areas of latent factors and clusters don't have well defined labels for comparing the results. It is interesting to apply this methodology to other types of data which is easier to verify the performance. In this work, the genes are filtered by variance filter, while more complex and variate types of filter can be applied to test the results. For example, classic feature selection techniques as LASSO or Elastic-net can be applied further.

In this work, the synthetic data test and real data analysis only use K-means clustering algorithm, so another point to work on is to test more types of algorithms and other distance metrics for clustering.

References

[1] R. Bellman, *Dynamic Programming*. Dover Publications, 1957.

[2] S. Mullainathan and J. Spiess, "Machine learning: An applied econometric approach," *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 87–106, 2017.

[3] R. Tibshirani, "Regression Shrinkage and Selection Via the Lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[4] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, vol. 67, no. 2, pp. 301–320, 2005.

[5] R. Argelaguet, B. Velten, D. Arnol, S. Dietrich, T. Zenz, J. C. Marioni, F. Buettner, W. Huber, and O. Stegle, "Multi-Omics Factor Analysis—a framework for unsupervised integration of multi-omics data sets," *Molecular Systems Biology*, vol. 14, no. 6, p. e8124, 2018.

[6] R. Shen, A. B. Olshen, and M. Ladanyi, "Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis," vol. 25, no. 22, pp. 2906–2912, 2009.

[7] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, vol. 61, no. 3, pp. 611–622, 1999.

[8] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "NP-hardness of Euclidean sum-of-squares clustering," *Machine Learning*, vol. 75, no. 2, pp. 245–248, 2009.

[9] H. Zha, X. He, C. Ding, H. Simon, and M. Gu, "Spectral relaxation for k-rneans clustering," *Advances in Neural Information Processing Systems*, 2002.

[10] C. Ding and X. He, "Cluster structure of K-means clustering via principal component analysis," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3056, pp. 414–418, 2004.

[11] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational Inference: A Review for Statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[12] R. Argelaguet, B. Velten, D. Arnol, S. Dietrich, T. Zenz, J. C. Marioni, F. Buettner, W. Huber, and O. Stegle, "Multi-Omics Factor Analysis—a framework for unsupervised integration of multi-omics data sets," *Molecular Systems Biology*, vol. 14, no. 6, 2018.

[13] "Python scikit package." https://scikit-learn.org/stable/. Accessed: 2019-10-23.

[14] "Ensemble genome browser." https://www.ensembl.org/info/about/species.html. Accessed: 2019-10-23.

[15] "Consensus cds project." https://www.ncbi.nlm.nih.gov/projects/CCDS/CcdsBrowse.cgi. Accessed: 2019-10-23.