Bioinformatics, 38(4), 2022, 900–907
https://doi.org/10.1093/bioinformatics/btab786
Advance Access Publication Date: 26 November 2021
Original Paper

Genome analysis

PIntMF: Penalized Integrative Matrix Factorization method for multi-omics data

Morgane Pierre-Jean *, Florence Mauger, Jean-François Deleuze and Edith Le Floch

Centre National de Recherche en Génomique Humaine, CEA, Université de Paris-Saclay, Evry, France

*To whom correspondence should be addressed.

Associate Editor: Tobias Marschall

Received on March 11, 2021; revised on September 30, 2021; editorial decision on November 11, 2021; accepted on November 11, 2021

Abstract

Motivation: It is more and more common to perform multi-omics analyses to explore the genome at diverse levels and not only at a single level. Through integrative statistical methods, multi-omics data have the power to reveal new biological processes, potential biomarkers and subgroups in a cohort. Matrix factorization (MF) is an unsupervised statistical method that allows a clustering of individuals, but also reveals relevant omics variables from the various blocks.

Results: Here, we present PIntMF (Penalized Integrative Matrix Factorization), an MF model with sparsity, positivity and equality constraints. To induce sparsity in the model, we used a classical Lasso penalization on variable and individual matrices. For the matrix of samples, sparsity helps in the clustering, while normalization (matching an equality constraint) of inferred coefficients is added to improve interpretation. Moreover, we added an automatic tuning of the sparsity parameters using the famous glmnet package. We also proposed three criteria to help the user to choose the number of latent variables. PIntMF was compared with other state-of-the-art integrative methods including feature selection techniques in both synthetic and real data. PIntMF succeeds in finding relevant clusters as well as variables in two types of simulated data (correlated and uncorrelated). Next, PIntMF was applied to two real datasets (Diet and cancer), and it revealed interpretable clusters linked to available clinical data. Our method outperforms the existing ones on two criteria (clustering and variable selection). We show that PIntMF is an easy, fast and powerful tool to extract patterns and cluster samples from multi-omics data.

Availability and implementation: An R package is available at https://github.com/mpierrejean/pintmf.

Contact: mpierrejean.pro@gmail.com

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The improvement of high-throughput biological technologies enables the production of various omics data such as genomic, transcriptomic, epigenomic, proteomic and metabolomic data (Ritchie *et al.*, 2015; Yugi *et al.*, 2016). The generation of these data allows investigating biological processes in cancer or complex diseases. For example, The Cancer Genome Atlas [TCGA (Network *et al.*, 2012)] has already produced numerous omics data for a set of 32 cancer types (Vasaikar *et al.*, 2018). Recently, other multi-omics studies on complex diseases and single-cell data have been published (Bock *et al.*, 2016; Rowlands *et al.*, 2014; Yang, 2020).

However, integrating omics data addresses several statistical challenges, such as dealing with a large number of variables, few samples and data heterogeneity (Bersanelli *et al.*, 2016). Indeed, the statistical distributions of omics data are very heterogeneous. For instance, mutations can be modeled by a binary distribution, while RNAseq data can be modeled by a Negative Binomial distribution and metabolomic data by a Gaussian distribution. In addition, the omic block sizes could vary from one hundred to one billion variables. Furthermore, collecting several types of omics data from a single sample could be difficult due to the cost and access to the biological material.

Over the past decade, unsupervised integrative methods have been developed to analyze multi-omics datasets and to identify potential biomarkers and new classifications in complex diseases (Cantini *et al.*, 2020; Chauvel *et al.*, 2020; Huang *et al.*, 2017; Pierre-Jean *et al.*, 2020; Tini *et al.*, 2019). Blocks of omics data can be seen as matrices, and relevant information can be extracted using dimension reduction methods, particularly, matrix factorization (MF) methods (Sastry *et al.*, 2020) and canonical correlation analysis (CCA) (Tenenhaus and Tenenhaus, 2011).

CCA methods are used to integrate multi-omics data and aim to maximize the correlation between omics datasets under certain constraints (Rodosthenous *et al.*, 2020; Tenenhaus *et al.*, 2014; Tenenhaus and Tenenhaus, 2011).

Then, MF techniques infer two matrices when applied to a single omic dataset: the first one describes the structure between variables (e.g. genes, probes, regions) and the second one describes the structure between samples.

One famous MF method is the Non-Negative Matrix Factorization [NMF (Lee and Seung, 1999)]. This method implements non-negativity constraints on the two inferred matrices. NMF provides a way to explain the structure of data by providing variable profiles (dictionary for each dimension). NMF also enables a classification of the samples thanks to the second matrix. The NMF is a commonly applied method used for a single omic block to identify disease subtypes in gene expression data (Burstein *et al.*, 2015) or recently, in DNA methylation data (Reilly *et al.*, 2019).

More recently, extensions of MF have been developed to perform integrative analysis (Chalise *et al.*, 2014; Chen and Zhang, 2018; Mo *et al.*, 2013). MF extensions need to infer more than two matrices: one matrix for each omic block is computed and one matrix for samples.

MF showed that it is a powerful technique to integrate heterogeneous data (Cantini *et al.*, 2020; Chauvel *et al.*, 2020; Pierre-Jean *et al.*, 2020). In our article, we propose a Penalized Integrative Matrix Factorization method called PIntMF, to discover new patterns and a new classification of a cohort. First, to add sparsity on the first inferred matrix (corresponding to the variable blocks), we used a common regularization technique: the Least Absolute Shrinkage and Selection Operator [LASSO (Tibshirani, 1996)]. Moreover, the sparsity on the variable block helps to the interpretation of patterns that drive the clustering of the samples. Then, sparsity, non-negativity and equality constraints are added to the second matrix (corresponding to the samples) to improve the interpretability of the clustering. The originality is the mix of the constraints for the clustering of the samples and the discovery of potential biomarkers.

In addition, we propose criteria to choose the number of latent variables and to properly initialize the algorithm.

The performance of this new unsupervised model was evaluated on both simulated and real data. First, we applied PIntMF on a simulated framework introduced by our group in Pierre-Jean *et al.* (2020) and on a simulated framework from Chung and Kang (2019). We compared our method to several existing unsupervised methods that perform both variable selection and clustering: intNMF (Chalise and Fridley, 2017), SGCCA (Tenenhaus *et al.*, 2014), MoCluster (Meng *et al.*, 2016), CIMLR (Ramazzotti *et al.*, 2018) and iClusterPlus (Mo and Shen, 2018). Then, we applied the model on a murine liver dataset (Williams *et al.*, 2016) and glioblastoma cancer data from TCGA already used in Shen *et al.* (2012).

2 Materials and methods

2.1 Model description

In the following, $\mathbf{A}$ denotes a matrix, $\mathbf{a}$ is a vector and a is a scalar. We consider K matrices $\mathbf{X}_1, \dots \mathbf{X}_K$ as the input of each method. Each matrix $\mathbf{X}_k$ is of size $n \times J_k$ (n is the number of samples and J_k the number of variables for the block k). In this article, we propose a model based on the matrix factorization method, i.e.:

$$\mathbf{X}^k \approx \mathbf{W}\mathbf{H}^k \tag{1}$$

where $\mathbf{W}$ denotes a common basis matrix and $\mathbf{H}^k$ a specific coefficient matrix associated with the block k. $\mathbf{W}$ is of size $n \times P$ and $\mathbf{H}^k$ is of size $P \times J_k$. Therefore, the variable P is the number of latent variables in the model.

To ensure identifiability and improve interpretation of the model, non-negativity and sparsity constraints are imposed on $\mathbf{W}$ [as in intNMF model described in Chalise and Fridley (2017)]. $\mathbf{W}$ will be used to cluster samples simultaneously across the K omics blocks. On $\mathbf{H}^k$, a sparsity constraint is imposed to perform variable selection simultaneously to the clustering of samples. Sparsity ensures a better interpretation of the variables that drive the clustering of samples. The model 1 can be extended to the following optimization problem:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{H}_1,\dots,\mathbf{H}^k} \sum_{k=1}^{K} ||\mathbf{X}^k - \mathbf{W}\mathbf{H}^k||_F^2 + \lambda_k ||\mathbf{H}^k||_1 + \\ \sum_{i=1}^{n} \mu_i ||\mathbf{w}_{i\bullet}||_1 \\ \text{s.t.} \quad \mathbf{W} \geq 0 \end{aligned} \tag{2}$$

where $||\mathbf{H}^k||_1 = \sum_{p=1}^{P} \sum_{j=1}^{J_k} |h_{pj}^k|$.

2.2 Solving equation

The optimization problem 2 is not convex on $\mathbf{W}, \mathbf{H}_1, \dots, \mathbf{H}^K$, but is convex separately on each matrix. Consequently, it can be solved alternatively on $\mathbf{W}, \mathbf{H}_1, \dots, \mathbf{H}^K$ until convergence.

Solve on $\mathbf{W}$: In this step, each $\mathbf{H}^k$ is fixed and the problem 3 is solved on $\mathbf{W}$.

$$\min_{\mathbf{W}} \sum_{k=1}^{K} ||\mathbf{X}^k - \mathbf{W}\mathbf{H}^k||_F^2 + \sum_{i=1}^{n} \mu_i ||\mathbf{w}_{i\bullet}||_1 \quad \text{st.} \quad \mathbf{W} \geq 0 \tag{3}$$

All individuals are independent for the weights $\mathbf{W}$ when $\mathbf{H}^k$ are fixed. The problem for an individual i can be written as follows:

$$\min_{\mathbf{w}_{i\bullet}} \sum_{k=1}^{K} ||\mathbf{x}_{i\bullet}^k - \mathbf{w}_{i\bullet}\mathbf{H}^k||^2 + \mu_i ||\mathbf{w}_{i\bullet}||_1 \quad \text{st.} \quad \mathbf{w}_{i\bullet} \geq 0 \tag{4}$$

Equation 4 is equivalent to

$$\min_{\mathbf{w}_{i\bullet}} \sum_{k=1}^{K} \sum_{j=1}^{J_k} (x_{ij}^k - \mathbf{w}_{i\bullet}\mathbf{h}_{\bullet j}^k)^2 + \mu_i ||\mathbf{w}_{i\bullet}||_1 \quad \text{st.} \quad \mathbf{w}_{i\bullet} \geq 0 \tag{5}$$

The optimization problem described by 5 is a classical lasso problem with a positivity constraint. It can be easily and fastly solved by glmnet R package (Jerome *et al.*, 2010).

Solve on $\mathbf{H}^k$: When $\mathbf{W}$ is fixed, each $\mathbf{H}^k$ can be solved independently. In this section, to be more readable, the index k is removed from the equations.

$$\min_{\mathbf{H}} Q(\mathbf{H}) = \min_{\mathbf{H}} ||\mathbf{X} - \mathbf{W}\mathbf{H}||_F^2 + \lambda \sum_{p=1}^{P} \sum_{j=1}^{J} |h_{pj}| \tag{6}$$

$$\begin{aligned} Q(\mathbf{H}) &= \text{Trace}\{(\mathbf{X} - \mathbf{W}\mathbf{H})(\mathbf{X} - \mathbf{W}\mathbf{H})^T\} + \\ &\quad \lambda \sum_{p=1}^{P} \sum_{j=1}^{J} |h_{pj}| \\ &= vec(\mathbf{X} - \mathbf{W}\mathbf{H})^T vec(\mathbf{X} - \mathbf{W}\mathbf{H}) + \\ &\quad \lambda \sum_{p=1}^{P} \sum_{j=1}^{J} |h_{pj}| \end{aligned}$$

We denote

$$\mathbf{h} = vec\left(\mathbf{H}\right) = \begin{pmatrix} \mathbf{H}_{11} \\ \vdots \\ \mathbf{H}_{P1} \\ \vdots \\ \mathbf{H}_{1J} \\ \vdots \\ \mathbf{H}_{PJ} \end{pmatrix} \text{ and } \mathbf{x} = vec\left(\mathbf{X}\right) = \begin{pmatrix} \mathbf{X}_{11} \\ \vdots \\ \mathbf{X}_{n1} \\ \vdots \\ \mathbf{X}_{1J} \\ \vdots \\ \mathbf{X}_{nJ} \end{pmatrix}.$$

$$\begin{aligned} Q(\mathbf{H}) &= (\mathbf{x} - vec(\mathbf{WH}))^T(\mathbf{x} - vec(\mathbf{WH})) + \lambda||\mathbf{h}||_1 \\ &= (\mathbf{x} - (\mathbb{I}_J \otimes \mathbf{W})vec(\mathbf{H}))^T(\mathbf{x} - (\mathbb{I}_J \otimes \mathbf{W})vec(\mathbf{H})) \\ &\quad + \lambda||\mathbf{h}||_1 \\ &= (\mathbf{x} - \tilde{\mathbf{W}}\mathbf{h})^T(\mathbf{x} - \tilde{\mathbf{W}}\mathbf{h}) + \lambda||\mathbf{h}||_1 \end{aligned}$$

where $\mathbb{I}_J$ is the identity matrix of size J and $\tilde{\mathbf{W}} = \mathbb{I}_J \otimes \mathbf{W}$

We can reformulate the problem as LASSO:

$$Q(\mathbf{H}) = ||\mathbf{x} - \tilde{\mathbf{W}}\mathbf{h}||^2 + \lambda||\mathbf{h}||_1$$

$\boldsymbol{\lambda}$ will be optimized for each block $k = 1, \ldots, K$.

As for $\mathbf{W}$, we used the glmnet package to solve this problem.

Normalization We would like to consider $\mathbf{W}$ as a weight matrix. To avoid problems of convergence or non-identifiability, the normalization by the sum of weights for each row of $\mathbf{W}$ is added after computing the matrix, i.e. each row is divided by its sum after each step:

$$\mathbf{w}_{i\bullet} = \frac{\mathbf{w}_{i\bullet}}{\sum_{p=1}^{P} \mathbf{w}_{ip}} \tag{7}$$

Therefore, the normalization corresponds to an equality constraint.

2.3 Stopping criteria

The stopping criterion of the model is determined by the convergence of the matrix $\mathbf{W}$. The stability of the similarity of matrix $\mathbf{W}$ between two iterations means that the model has converged therefore we stop the algorithm. The similarity between $\mathbf{W}^{t-1}$ and $\mathbf{W}^t$ is measured with the Adjusted Rand Index (ARI). The users have also the possibility to define a maximum number of iterations to limit the computing time of the algorithm.

2.4 Automatic tuning of sparsity parameters

For each block $\mathbf{H}^k$ and $\mathbf{W}$, we need to calibrate the sparsity parameter λ_k and μ_i. The main advantage of glmnet package is the speed (see Supplementary Fig. S9), and, it implements a cross validation technique to choose the best λ or μ. PIntMF takes advantage of glmnet to calibrate the penalty on each block. We use CV at each step to find the optimal values for λ and μ. However, all μ_i have been set to 1 in the following experiments (simulations and applications) to save computational time since the results were very similar.

Therefore, the only parameter that the user needs to tune is the number of latent variables P.

2.5 Optimization of the algorithm

Initialization: Often in NMF algorithms (Lee and Seung, 1999), the matrices are initialized by non-negative random values. We assess four kinds of initialization for PIntMF (hierarchical clustering, random, Similarity Network Fusion and Singular Values Decomposition).

The best initialization is based on the SNF algorithm (Wang *et al.*, 2014) (Supplementary Fig. S1). This initialization has the advantage to take into account simultaneously the K blocks of the analysis. Therefore, for all the following analyses, SNF initialization was used.

Computing optimization of H: Several algorithms to solve the Lasso problem on $\mathbf{H}^k$ were tested. glmnet is the fastest package among them (Supplementary Fig. S9).

2.6 Clustering

In this article, all clusterings are obtained by applying a hierarchical clustering with the ward distance (Ward Jr, 1963) on matrix $\mathbf{W}$. For the optimal number of clusters, P is chosen.

2.7 Criteria to choose the best model

In this section, we present three different criteria to choose the appropriate number of latent variables (P).

Mean square error: The number of latent variables can be optimized by looking at the curve of the Mean Square Error (MSE). In this context, the mean square error (MSE) for each dataset k is defined by:

$$MSE_P^k = \frac{||\mathbf{X}^k - \mathbf{WH}^k||_F^2}{n \times J_k} \tag{8}$$

Then, the total MSE is then defined by averaging the different MSE_P^k:

$$MSE_P = \sum_k MSE_P^k / K \tag{9}$$

Percentage of variation explained (PVE): To measure the performance of the method, we computed the Percentage of Variation Explained (Nowak *et al.*, 2011) defined by the following formula:

$$PVE(\mathbf{W}, \mathbf{H}^k) = 1 - \frac{||\mathbf{X}^k - \mathbf{WH}^k||_F^2}{||\mathbf{X}^k - \overline{\mathbf{X}}^k \mathbf{1}_{Jk}||_F^2} \tag{10}$$

where $\overline{\mathbf{X}}^k$ is a vector containing the average profile of each individual:

$\overline{\mathbf{X}}_i^k = \frac{\sum_j x_{ij}}{J_k}$, and $1_{Jk} = (1, \ldots, 1)$ is a row-vector of size J_k.

Then, we computed the global PVE as the mean of the PVE on the K blocks, i.e.:

$$PVE = \frac{1}{K}\sum_{k=1}^{K} PVE(\mathbf{W}, \mathbf{H}^k) \tag{11}$$

Cophenetic distance: We were inspired by Gaujoux and Seoighe (2010) for the last criterion.

We wanted to assess if the distances in the tree (after hierarchical clustering on $\mathbf{W}$) accurately reflect the original distances.

One way is to compute the correlation between the cophenetic distances and the original distance data generated by the dist() function on $\mathbf{W}$ (Sokal and Rohlf, 1962). The clustering is valid, if the correlation between the two quantities is high. Note that, we use the cophenetic function defined by Sneath *et al.* (1973).

The cophenetic correlation usually decreases with the increase of P values. Brunet *et al.* (2004) suggested choosing the smallest value of P for which this coefficient starts decreasing.

3 Performance criteria

Two criteria are used to assess the performance of our method and to compare it with others.

3.1 Adjusted Rand Index

On a simulated dataset and on well-known real datasets, it is possible to compute the similarity between the true and the inferred classifications. We use the ARI as a criterion to evaluate the performance of our method. The ARI (Rand, 1971) is equal to one when the two classifications that are compared are totally similar and zero or even negative if the classifications are completely different.

3.2 Area under the ROC curve

On a simulated dataset, the variables that drive the subgroups are known, and it is easy to compute false-positive and true-positive rates. First, variables are ordered by their standard deviation (from the highest to the lowest) computed on the $\mathbf{H}$ matrix to highlight the largest differences between the P components and therefore the most contributory to the clusters. To summarize the information of these two quantities, we compute the area under the TPR-FPR curve [area under the roc curve (AUROC)]. An AUROC equal to one means that the method selects the variables with no error. An AUROC under 0.50 means that false-positive variables are selected before the true positive ones.

4 Results

4.1 Performance on simulated datasets

We assess the performance of PIntMF in two simulated frameworks described below.

4.1.1 Simulations on independent datasets (non-correlated blocks)

The performance of PIntMF to cluster samples and to select relevant variables was evaluated on simulated data described by Pierre-Jean *et al.* (2020). The framework of these simulations is composed of three blocks with three different types of distribution (Binary, Beta-like and Gaussian) to simulate the heterogeneity of the integrative omics data studies. Indeed, a binary distribution could match a mutation (equal to 1 if the gene is mutated and 0 otherwise); a Beta-like distribution could match DNA methylation data, and a Gaussian distribution could match gene expression values.

Four unbalanced clusters (composed of 25, 20, 5 and 10 individuals) have been simulated (Benchmarks 1–5). Datasets with 2, 3 and 4 balanced clusters have also been simulated (Benchmarks 6–8). Each benchmark is simulated 50 times.

PIntMF was compared with several integrative unsupervised methods (Pierre-Jean *et al.*, 2020) that perform both clustering and variable selection namely: intNMF (Chalise *et al.*, 2014), SGCCA (Tenenhaus *et al.*, 2014), MoCluster (Meng *et al.*, 2016), iClusterPlus (Mo *et al.*, 2013) and CIMLR (Ramazzotti *et al.*, 2018).

Clustering performance was evaluated using the ARI on simulated data (see Section 3.1).

On the eight simulated benchmarks with various levels of signal-to-noise ratio, PIntMF and MoCluster outperform the other methods with an ARI equal to 1 in most cases (Fig. 1).

The performance of variable selection is assessed using the AUROCs after computing False Positive Rates (FPR) and True Positive Rates (TPR) (see Section 3.2). The computation of the AUROC shows that PIntMF performs as well as MoCluster on the three types of data (Supplementary Table S2). Indeed, PIntMF reaches either the first or the second-best AUROC for these simulations. Moreover, the lowest AUROC is equal to 0.88, which means that the method is both sensitive and specific.

4.1.2 Simulation based on real data (correlated blocks)

We evaluate the performance of PIntMF on a simulated framework based on real cancer data developed by Chung and Kang (2019). Indeed, the previous simulated framework does not simulate any correlation between omics blocks.

OmicsSIMLA is a simulation tool for generating multi-omics data with disease status. The tool simulates CpGs with methylation proportions, RNA-seq read counts and normalized protein expression levels. Here, we simulated 50 datasets containing 50 cases (i.e. short-term survival) and 50 controls (i.e. long-term survival), and three omics blocks (RNAseq, DNA methylation and proteins). We aimed to recover the two groups but also the different features that drive overall survival using the simulated DNA methylation, expression and protein data. For two of the three blocks (expression and DNA methylation), the variables simulated with a differential expression or methylation between the two groups are known. The simulated data are described in Supplementary Materials (Supplementary Section S6).

In these simulations, we also compared the performance of PIntMF to other methods in terms of clustering and variable selection. First, CIMLR does not give any results on these simulations (the algorithm does not converge). For all the other methods, the ARI is equal to 1 (maximum value) for all 50 datasets.

Then, we compared the variable selection performance of PIntMF, intNMF, iClusterPlus, MoCluster and SGCCA by computing the AUROC on expression and DNA methylation blocks only (the protein block does not contain any variable simulated with differential abundance, more details are given in Supplementary Section S6).

DNA Methylation dataset: PIntMF and iclusterPlus outperform the others with similar performances but the AUROC of iclusterPlus is significantly higher. However, the AUROC of PIntMF is significantly higher than for MoCluster, SGCCA and intNMF (Fig. 2).

Expression dataset: PIntMF is the best method with an AUROC significantly higher than the other methods. However, all methods achieve an AUROC higher than 0.92 (Fig. 2).

On these simulations, PIntMF gives similar results to iClusterPlus, but with automatic tuning of parameters. Moreover, the algorithm of PIntMF is faster than iClusterPlus.

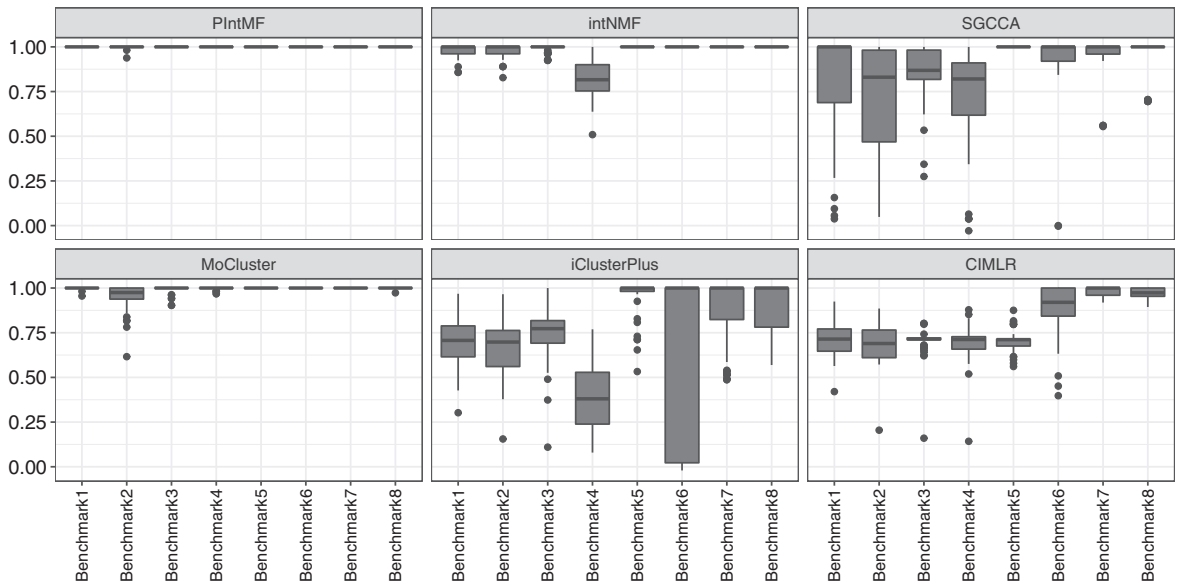


Fig. 1. ARI of PIntMF, intNMF, SGCCA, MoCluster, iClusterPlus and CIMLR methods on simulated datasets. B1: Reference, B2: More Gaussian noise, B3: More Gaussian noise and more Binary noise, B4: More Beta noise and more Binary noise, B5: More Relevant variables, B6: 2 balanced clusters, B7: 3 balanced clusters, B8: 4 balanced clusters

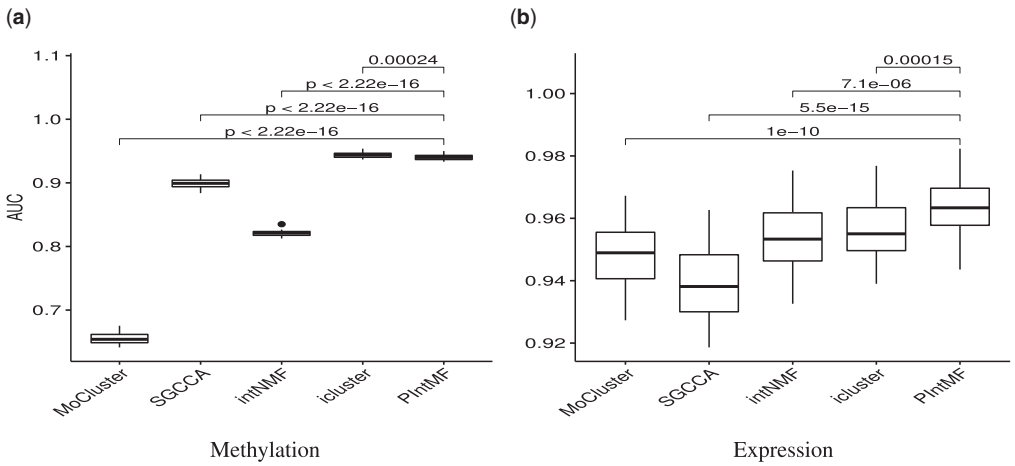


Fig. 2. AUROC of PIntMF, MoCluster, SGCCA, iClusterPlus and intNMF for OmicsSIMLA simulations on (**a**) DNA methylation and (**b**) Gene expression blocks

4.1.3 Stability selection

Jackknife was performed to evaluate the stability of variable selection. To perform this technique, we run the model PIntMF on the data without one sample at each step. Therefore, we obtain n datasets containing $n-1$ individuals on which we apply the method.

The stability of the selected variables for Binary, Gaussian, methylation and expression datasets seems to be strong (Supplementary Fig. S10). For proteins and for beta-like data, the bootstrap reveals that some selected variables are not stable, even though it was expected for proteins since they were not simulated with a differential expression between clusters. In the case where the selection is not stable, the solution could be using jackknife to remove potential false-positive variables.

4.1.4 Summary

Our method PIntMF provides satisfying clustering and variable selection both on correlated blocks (Simulation Framework 2) and on non-correlated blocks (Simulation Framework 1). PIntMF is the only method that performs well on all simulated settings. We conclude on these two frameworks of simulated data that PIntMF is a fast and flexible tool.

4.2 Applications

In this section, we assess the performance of the PIntMF method on real data by considering two applications. The first one is a dataset from murine liver (Williams *et al.*, 2016) under two different diets already used in two previous comparison articles (Pierre-Jean *et al.*, 2020; Tini *et al.*, 2019), and the objective is to recover the diets of the mice (fat diet or chow diet). The second one is a glioblastoma dataset from TCGA used by Shen *et al.* (2012) and the goal is to find the tumor subtypes.

4.2.1 PIntMF highlights variables linked to phenotypes of samples

We analyzed the BXD cohort (composed of 64 samples) (Williams *et al.*, 2016); the mice were divided into two different environmental conditions of diet: chow diet (CD) (6% kcal of fat) or high-fat diet (HFD) (60% kcal of fat). Measurements have been made in the livers of the entire population at the transcriptome (35 554 variables), the proteome (21 547 variables) and the metabolome (956 variables) levels.

Therefore, we applied PIntMF to this dataset as well as intNMF, MoCluster, SGCCA, iClusterPlus and CIMLR (Supplementary Table S4).

PIntMF produces a perfect classification of the individuals for this real dataset.

For this dataset, all criteria for the model selection were computed (Supplementary Fig. S6), and two groups were selected for further analysis.

PIntMF highlights interesting variables that seem to have different abundance between the two groups CD and HFD (Fig. 3): Vitamin E ($C_{29}H_{50}O_2$), Cholesteryl ($C_{36}H_{62}O_5$), Mustard Oil (C_4H_5NS). Saa2 gene that codes for a protein involved in the HDL complex seems to be differentially expressed between the two groups. Then, the Cidea gene that is involved in the metabolism of lipids and lipoproteins has a slightly different level of expression between the two groups. Finally, Cyp2b9 oxides steroids, fatty acids and xenobiotics are less expressed in the high-fat diet group. To conclude, PIntMF succeeds well into recovering the correct classification and relevant markers in all datasets.

4.2. 2 PIntMF reveals a new classification of non-annotated samples on TCGA dataset

In addition, we analyzed a subset of the glioblastoma dataset from the cancer genome atlas (TCGA): the Glioblastoma study (2009) used in (Shen *et al.*, 2012). The dataset contains three matrices: copy number variation (1599 regions), DNA methylation (1515 CpGs) and mRNA expression (1740 genes) in 55 samples. GBM samples were classified into four subtypes (Classical: CL, Mesenchymal: MES, Neural: NL and Proneural: PN). In addition, there are samples with no subtype (NA). Using the PIntMF method, we highlight samples with no classification close to labeled samples. Looking at the three criteria, the best number of latent variables seems to be five (Supplementary Fig. S7). For example, the green cluster from PIntMF matches a part of the CL subtype, and one sample labeled as NA is in this green cluster. Then, the purple cluster from PIntMF matches the PN subtype, and one sample labeled as NA can be classified within the PN subtype (Fig. 4a). Clusters 1 (red) and 2 (blue) are more heterogenous. However, the red one is composed of NL and NA labeled samples. The blue one is close to samples labeled as PN.

We performed a survival analysis to identify a relation between groups found by PIntMF and the survival rate (Fig. 4b). The survival test gives a significant P-value at 5% (P-value = 0.00013 with log-rank test). The prognosis for the purple (4) group is better than those of the red and green (1 and 3) groups and even better than the orange and blue (2 and 5) groups. Note that, the PN subtype is split into two groups (purple and blue) that have two very different survival curves.

The previous study (Shen *et al.*, 2012) performed with the iCluster method (Shen *et al.*, 2009) identified 3 subgroups with a less significant P-value (0.01) than PIntMF for the survival differences between subgroups. Their Cluster 1 matches the PN group, Cluster 2 matches the CL group and Cluster 3 is mostly composed of the MES subtype. Authors do not give any information about the samples with no subtypes.

H matrices exhibit various types of genomic profiles according to the clusters (Fig. 4). For instance, the orange cluster (5) shows

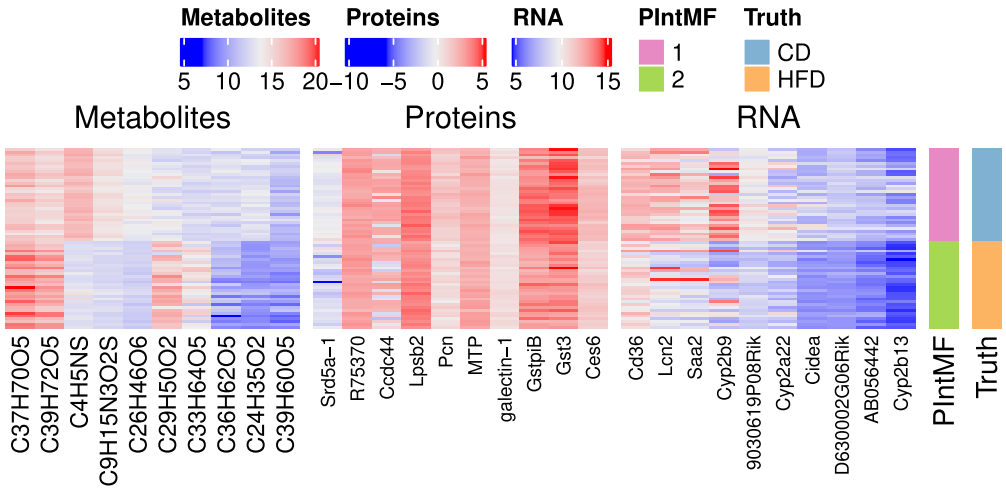
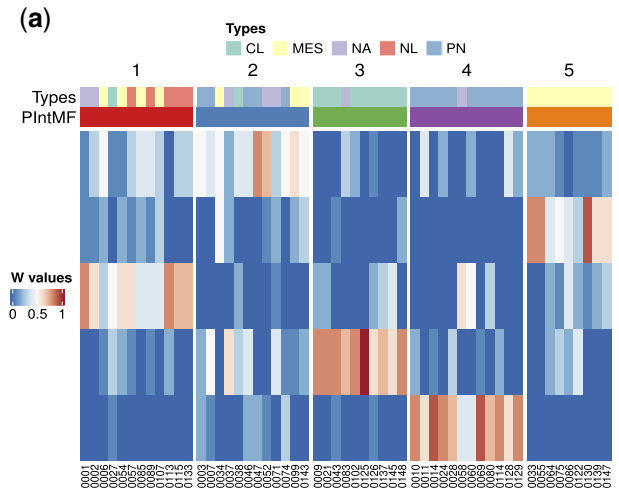
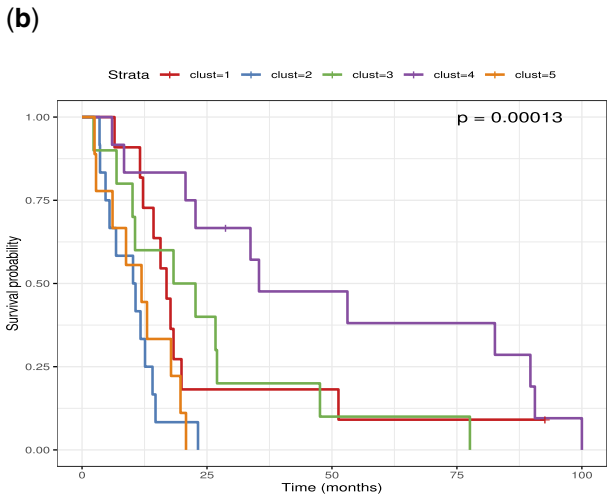


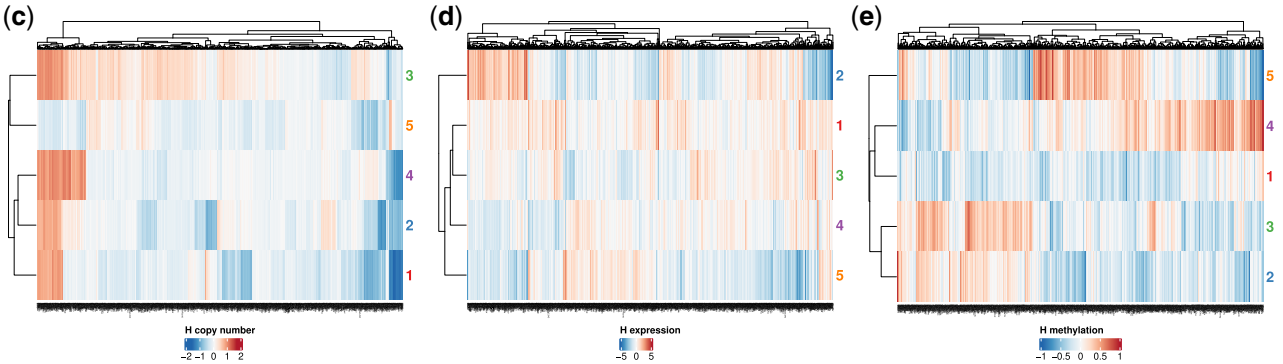
Fig. 3. BXD cohort results: Top 10 selected variables with PIntMF of each dataset (Metabolites, Proteins and RNA), the clustering given by PIntMF and the true clustering are on the right



Heatmap plot of **W**: Homogeneity between subtypes and subgroups identified by PIntMF



Kaplan-Meier plot: The subgroups identified by PIntMF show survival differences



Copy number variation | Gene Expression | DNA Methylation

Fig. 4. (**a**) Heatmap of **W**. The clustering of PIntMF was compared to glioblastoma subtypes. (**b**) Survival curves with *P*-value of log-rank test. (**c–e**) **H** matrix for the three considered omics blocks on glioblastoma dataset

few alterations at the copy number variation level (Fig. 4c) but a particular profile for DNA methylation and gene expression data (Fig. 4e). The blue cluster (2) has a distinct pattern of expression (Fig. 4d).

5 Discussion

Here, we present the PIntMF model that is capable of discovering new subgroups from a cohort and potential new biomarkers from several types of omics data. PIntMF is a matrix factorization model with positivity and sparsity constraints (Lasso) on inferred matrices. The method and all the scripts of this article are available in an R package entitled PIntMF.

The main advantage of this method is the automatic tuning of the lasso penalties for both variable and sample matrices. To optimize the computational time of the algorithm (Supplementary Fig. S9), we tried several algorithms to infer matrices $\mathbf{H}^k$. `glmnet` is very fast compared with the other widely used algorithms (ncvreg, quadrupen and biglasso), therefore it was retained for all the analyses. We also optimized the algorithm initialization using the SNF algorithm (Wang *et al.*, 2014). This initialization enables the algorithm to provide the best clustering and the best percentage of explained variation (Supplementary Fig. S1). Moreover, this initialization is performed at the integrative level rather than on each individual data block.

PIntMF automatically tunes the penalties on matrices $\mathbf{H}^k$ and $\mathbf{W}$, without any user intervention, and we noticed that all the matrices are quite sparse on real datasets (Fig. 4). The user needs to choose only one parameter that is the number of latent variables. The last parameter can be chosen by looking at the MSE, cophenetic coefficient and the PVE (Supplementary Figs S2–S6). All these criteria are implemented in the R package. For non-correlated data simulations, only the cophenetic coefficient and the PVE allow selecting the correct number of latent variables.

It is still difficult to evaluate the performance of an integrative method on simulations (Cantini *et al.*, 2020). The relationships between blocks of omics data are complex, often not well-known, and the modeling of these links is complicated. To our knowledge, there does not exist any reference dataset to assess performances in terms of clustering and variable selection. Therefore, we evaluated the algorithm on two different simulation frameworks (completely simulated and based on real-data) and two real datasets. Furthermore, we compared it with several other state-of-the-art integrative methods. We demonstrated, on the first simulated dataset (non-correlated blocks), that PIntMF outperforms the other methods on both clustering and variable selection. Indeed, on simulated data, the clustering from PIntMF makes few classification errors. PIntMF is also more robust to heterogeneous data compared with the others: the method performs as well on gaussian distributions as on binary or beta distributions for the variable selection. On another simulated framework based on real data (correlated blocks), we observed good clustering performances (perfect classification) and variable selection levels (AUROC upper than 90%). When applying the algorithm on two real datasets (BXD and TCGA data, Section 4.2), we demonstrated that the method could deal with real datasets. In particular, we found relevant subgroups but also interesting variables linked to the clinical phenotypes (diet and overall survival).

A weakness of the model is that the convergence of the algorithm to an optimal solution is not mathematically justified. Furthermore, a significance test for each selected variable is not given due to the use of the LASSO regression (Jain and Xu, 2021). Jackknife could provide an idea of the confidence in the selected variables (Supplementary Fig. S10). However, this approach is very time-consuming when applied on large datasets.

The method could be further improved by dealing with missing values. Missing values could be inside a block for a few variables. These missing values could be imputed using the average of the other correlated variables, by the values of the nearest neighbor or by more complex methods as proposed by Voillet *et al.* (2016), González *et al.* (2009), Husson and Josse (2013) and Song *et al.* (2020). Commonly, a whole block can also be missing for an individual. In this case, the matrix $\mathbf{W}$ could be computed only on the blocks that are present for this individual. Thanks to the $\mathbf{W}$ matrix, we could deduce a new profile for this patient from the $\mathbf{H}^k$ matrix inferred with the other individuals.

We could also extend the scope of PIntMF by including prior information such as the genome structure. For instance, we could force the algorithm to select the same genes in the DNA methylation block and the expression block. A group Lasso penalty (Simon *et al.*, 2013) could be added to the proposed model to include such a prior.

To conclude, PIntMF is an easy and flexible method to integrate omics data. It implements an original mixture of constraints on matrices to cluster samples and discover new biomarkers. PIntMF exhibits good performance in terms of classification or variable selection in both simulation cases (correlated blocks or non-correlated blocks). It outperforms the other tested methods since it is the only one that works well in all our simulated frameworks. PIntMF is fast and automatically tunes the penalty for each block to select an appropriate number of variables (sparse matrices). Moreover, it provides a sparse matrix $\mathbf{W}$ to facilitate the clustering of samples. Finally, we also provide three criteria namely MSE, PVE and cophenetic coefficient to choose the best number of latent variables.

The integration of several types of omics data using our method could help in discovering potential new biomarkers even with a small number of patients. Finally, it could also help to classify patients with unknown phenotypes.

Data availability

An R package named PIntMF can be used to reproduce all simulations and figures and is available online at https://github.com/mpierrejean/pintmf.

Acknowledgement

The authors thank Steven McGinn for English language editing.

Financial Support: none declared.

Conflict of Interest: none declared.

References

Bersanelli,M. *et al.* (2016) Methods for the integration of multi-omics data: mathematical aspects. *BMC Bioinformatics*, **17**, 15.

Bock,C. *et al.* (2016) Multi-omics of single cells: strategies and applications. *Trends Biotechnol.*, **34**, 605–608.

Brunet,J.-P. *et al.* (2004) Metagenes and molecular pattern discovery using matrix factorization. *Proc. Nat. Acad. Sci. USA*, **101**, 4164–4169.

Burstein,M.D. *et al.* (2015) Comprehensive genomic analysis identifies novel subtypes and targets of triple-negative breast cancer. *Clin. Cancer Res.*, **21**, 1688–1698.

Cantini,L. *et al.* (2020) Benchmarking joint multi-omics dimensionality reduction approaches for cancer study. *Nat. Commun.*, **2**, 124.

Chalise,P., and Fridley,B.L. (2017) Integrative clustering of multi-level omic data based on non-negative matrix factorization algorithm. *PLoS One*, **12**, e0176278.

Chalise,P. *et al.* (2014) Integrative clustering methods for high-dimensional molecular data. *Transl. Cancer Res.*, **3**, 202–216.

Chauvel,C. *et al.* (2020) Evaluation of integrative clustering methods for the analysis of multi-omics data. *Brief. Bioinf.*, **21**, 541–552. [CrossRef][*10.1093/bib/bbz015*]

Chen,J., and Zhang,S. (2018) Discovery of two-level modular organization from matched genomic data via joint matrix tri-factorization. *Nucleic Acids Res.*, **46**, 5967–5976.

Chung,R.-H., and Kang,C.-Y. (2019) A multi-omics data simulator for complex disease studies and its application to evaluate multi-omics data analysis methods for disease classification. *GigaScience*, **8**, giz045.

Gaujoux,R., and Seoighe,C. (2010) A flexible r package for nonnegative matrix factorization. *BMC Bioinformatics*, **11**, 367.

González,I. *et al.* (2009) Highlighting relationships between heterogeneous biological data through graphical displays based on regularized canonical correlation analysis. *J. Biol. Syst.*, **17**, 173–199.

Huang,S. *et al.* (2017) More is better: recent progress in multi-omics data integration methods. *Front. Genet.*, **8**, 84.

Husson,F., and Josse,J. (2013) Handling missing values in multiple factor analysis. *Food Qual. Preference*, **30**, 77–85.

Jain,R., and Xu,W. (2021) Hdsi: high dimensional selection with interactions algorithm on feature selection and testing. *PLoS One*, **16**, e0246159.

Jerome,F. *et al.* (2010) Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.*, **33**, 1–22.

Lee,D.D., and Seung,H.S. (1999) Learning the parts of objects by non-negative matrix factorization. *Nature*, **401**, 788–791.

Meng,C. *et al.* (2016) mocluster: identifying joint patterns across multiple omics data sets. *J. Proteome Res.*, **15**, 755–765.

Mo,Q., and Shen,R. (2018) *iClusterPlus: Integrative Clustering of Multi-Type Genomic Data. Bioconductor R package version 1.18.0*.

Mo,Q. *et al.* (2013) Pattern discovery and cancer gene identification in integrated cancer genomic data. *Proc. Natl. Acad. Sci. USA*, **110**, 4245–4250.

Network,C.G.A. *et al.* (2012) Comprehensive molecular portraits of human breast tumours. *Nature*, **490**, 61.

Nowak,G. *et al.* (2011) A fused lasso latent feature model for analyzing multi-sample ACGH data. *Biostatistics*, **12**, 776–791.

Pierre-Jean,M. *et al.* (2020) Clustering and variable selection evaluation of 13 unsupervised methods for multi-omics data integration. *Brief. Bioinf.*, **21**, 2011–2030. [CrossRef/*[10.1093/bib/bbz138*]

Ramazzotti,D. *et al.* (2018) Multi-omic tumor data reveal diversity of molecular mechanisms that correlate with survival. *Nat. Commun.*, **9**, 4453.

Rand,W.M. (1971) Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.*, **66**, 846–850.

Reilly,B. *et al.* (2019) DNA methylation identifies genetically and prognostically distinct subtypes of myelodysplastic syndromes. *Blood Adv.*, **3**, 2845–2858.

Ritchie,M.D. *et al.* (2015) Methods of integrating data to uncover genotype–-phenotype interactions. *Nat. Rev. Genet.*, **16**, 85–97.

Rodosthenous,T. *et al.* (2020) Integrating multi-omics data through sparse canonical correlation analysis for the prediction of complex traits: a comparison study. *Bioinformatics*, **36**, 4616–4625. [CrossRef/*[10.1093/bioinformatics/btaa530]*

Rowlands,D.S. *et al.* (2014) Multi-omic integrated networks connect DNA methylation and miRNA with skeletal muscle plasticity to chronic exercise in type 2 diabetic obesity. *Physiol. Genomics*, **46**, 747–765.

Sastry,A.V. *et al.* (2020) Matrix factorization recovers consistent regulatory signals from disparate datasets. *BioRxiv*.

Shen,R. *et al.* (2009) Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics*, **25**, 2906–2912.

Shen,R. *et al.* (2012) Integrative subtype discovery in glioblastoma using icluster. *PLoS One*, **7**, e35236.

Simon,N. *et al.* (2013) A sparse-group lasso. *J. Comput. Graph. Stat.*, **22**, 231–245.

Sneath,P.H. *et al.* (1973) *Numerical Taxonomy. The Principles and Practice of Numerical Classification*.W. H. Freeman, San Francisco.

Sokal,R.R., and Rohlf,F.J. (1962) The comparison of dendrograms by objective methods. *Taxon*, **11**, 33–40.

Song,M. *et al.* (2020) A review of integrative imputation for multi-omics datasets. *Front. Genet.*, **11**, 570255.

Tenenhaus,A., and Tenenhaus,M. (2011) Regularized generalized canonical correlation analysis. *Psychometrika*, **76**, 257–284.

Tenenhaus,A. *et al.* (2014) Variable selection for generalized canonical correlation analysis. *Biostatistics*, **15**, 569–583.

Tibshirani,R. (1996) Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B (Methodological)*, **58**, pages 267–288.

Tini,G. *et al.* (2019) Multi-omics integration - a comparison of unsupervised clustering methodologies. *Brief. Bioinf.*, **20**, 1269–1279. [CrossRef/*[10.1093/bib/bbx167*]

Vasaikar,S.V. *et al.* (2018) Linkedomics: analyzing multi-omics data within and across 32 cancer types. *Nucleic Acids Res.*, **46**, D956–D963.

Voillet,V. *et al.* (2016) Handling missing rows in multi-omics data integration: multiple imputation in multiple factor analysis framework. *BMC Bioinformatics*, **17**, 1–16.

Wang,B. *et al.* (2014) Similarity network fusion for aggregating data types on a genomic scale. *Nat. Methods*, **11**, 333–337.

Ward Jr,J.H. (1963) Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.*, **58**, 236–244.

Williams,E.G. *et al.* (2016) Systems proteomics of liver mitochondria function. *Science*, **352**, aad0189.

Yang,X. (2020) Multitissue multiomics systems biology to dissect complex diseases. *Trends Mol. Med.*, **26**, 718–728. [CrossRef/*[10.1016/j.molmed.2020.04.006]*

Yugi,K. *et al.* (2016) Trans-omics: how to reconstruct biochemical networks across multiple omic layers. *Trends Biotechnol.*, **34**, 276–290.