UCLouvain

École polytechnique de Louvain

# Integration of single cell multi-omics data

Author: **Charlotte LEONARD**
Supervisors: **Michel VERLEYSEN, Laurent GATTO**
Readers: **Christophe VANDERAA, Philippe HAUCHAMPS**
Academic year 2021–2022
Master [120] in Mathematical Engineering

# Abstract

Recent advances in data extraction technologies enable researchers to collect new types of data. Analyzing these data can provide a new level of understanding of cells and the mechanisms that underlies them. Multiple methods exist to analyze these data. Including factorization methods. They are intended to reduce the dimension and enlighten patterns in the data matrices.

The integrative nonnegative matrix factorization with shared and unshared features (UINMF) method is a new matrix integration method that reduces the dimension and enlightens clusters in the data. It is derived from the workhouse method, in the field of matrix factorization, nonnegative matrix factorization (NMF). It is extended to capture different signals distinctly in cells and account for all features of the datasets, the features shared across datasets as well as the features unshared.

The purpose of this work is to evaluate if the algorithm is reliable and can be used on real biological data. We will therefore test if the algorithm can recognize patterns and clusters among test datasets. The test datasets will be constructed such that they will have proprieties similar to real biological data. For instance a high number of samples (cells) and features (genes).

# Acknowledgements

Firstly, I would like to thank my supervisors Prof. Michel Verleyzen and Prof. Laurent Gatto their support, evaluations, and explanations all trough throughout the year.

I also would like to express my gratitude to Christoph Vanderaa for his support all year long, as well as for his feedback and advice.

Furthermore, I would like to thank the readers and members of the jury, Philippe Hauchamps and Christophe Vanderaa for giving the time to review this work.

Finally, I would like to say that I am very grateful for my friends and family for their presence in the past few months.

# Contents

# Chapter 1

# Introduction

The mechanisms that happen in cells are very complex and rely on a lot of different molecular actors that can be quantified. Recent advances in data collection allowed us to extract different types of information about cells. Focusing on one cell can give more precise information and can improve the understanding of some mechanisms much better than analyzing multiple cells averaged together into an ensemble. These new technologies, collectively named single-cell technologies, can help us collect different types of information about cells. Those types of information, the modalities, are called omics data. Omics refers to a field of study in biological sciences that ends with -omics, such as genomics, transcriptomics, proteomics, or metabolomics, i.e. focusing on the whole genomes, transcriptomes, proteomes, or metabolomes.

Those modalities bring some quantitative information about the cells that can be used to extract a comprehensive view of cellular mechanisms. Combining the different types of information and analyzing different types of modalities jointly or analyzing the same modalities from different cells, from different species of some cancerous and non-cancerous cells can bring a better understanding of the cross-modalities interactions that define the cells[1, 4, 13, 32].

All this information collected in the cells about all the omics modalities can be analyzed and treated to extract new knowledge about cells. The new technologies that extract information create a need for analysis techniques to exploit these data.

In this work, we will discuss some of the analysis challenges that come up with the emerging data collection possibilities. We will then focus on the integration of different omics modalities. We will discuss the factorization algorithm that can enlighten patterns and clusters among the multi-omics datasets.

We will discuss a very common matrix factorization algorithm, the nonnegative matrix factorization (NMF). This method is widely used to discover structures in

physical or biological data. The purpose of NMF is to reduce the dimension of the datasets and extract hidden structures in the data. It is done by factorizing the data matrix into lower dimensional matrices. NMF both compresses the data and enables a better interpretation. The particularity of NMF is that the components are nonnegative. For most physical or biological applications the components only have a meaning if they are nonnegative. NMF is therefore a very natural choice for the physical or biological applications.

One difficulty of the nonnegative matrix factorization is that it is a non-convex optimization problem. It is therefore a non-trivial tool. Because of the non-convexity, the best algorithm can provide local minima. We will overview different algorithms that were designed to solve the NMF problem. The multiplicative update rule was one of the first methods created. This method is very popular for its natural and flexible foundation. The Alternating Least Squares (ALS) method that is one of the basic workhouse methods. And finally, the alternating Nonnegative Least Squares (ANLS) methods that is a methods that provides a very good computational efficiency.

Some different aspects of the problem must be modified to provide a good solution. Indeed the quality of the solution can be influenced by the initialization. The initialization is highly dependent on the problem and defining a good initialization is important. The uniqueness of the solution is also a point of great importance. As the problem has a high dimension, from the high number of features characteristic to omics data, the uniqueness of the solution is a concern. Finally, the regularization is also an important point. Indeed regularization can both enhance the uniqueness and add prior knowledge. The prior knowledge can influence the result toward a more realistic solution.

To have a better understanding of biological phenomenon, analysing the cross-modalities interaction between modalities can be highly valuable. To resolve those challenges, the NMF framework was extended to allow the integration of multiple datasets and multi-omics datasets. Joint NMF (jNMF) was developed as an extension to NMF to integrate multiple datasets that share a set of observations. jNMF contains a matrix that captures homogeneous signals shared across the datasets. One limitation of jNMf is that it makes the assumption that the data is homogeneous. To overcome this limitation, the integrative NMF (iNMF) was introduced. iNMF is an extension to jNMf that considers both homogeneous and heterogeneous effects.

Finally, UINMF is an extension to integrative NMF that incorporates unshared features. The unshared features allow the integration to take into account extra

genes, or any other data type measured in only one of the datasets. The algorithm UINMF reduces the dimension of the dataset by factorizing the data matrices into lower dimentional matrices. It also enlighten covariance among features through the latent factors inferred from both shared and unshared.

In this paper we will test the algorithm UINMF to assess its capacity to extract the heterogeneous signal from the homogeneous signal in heterogeneous data while accounting for unshared features. To evaluate the algorithm we will construct multiple input matrices. We will then perform the integration on the input matrices. Finally, we will assess and discuss the quality of the solutions. The purpose is to evaluate if the algorithm is reliable and can be used on real biological data. We will therefore test if the algorithm can recognize patterns and cluster among test datasest. We will construct dataset and attribute them a biological interpretation. We will test possible proprieties of real biological data as, for instance, a high number of samples (e.g. the cells) and features (e.g. the genes). Moreover we will test unbalance of modalities, i.e modalities with a lager number or features and/ or sample that the other. We will then discuss how the solution can be improved using regularization and initialization. Besides we will talk through the effect of regularization on the relevance and the intelligibility of the solution.

# Chapter 2

# Biology Background

## 2.1 Gene expression

The main source of information in the cells is DNA. The DNA contains the whole genetic information. This information is separated into units, the genes. Each gene contains information for making specific proteins that have a particular function. To make a protein, the gene is first transcripted into an RNA copy of the DNA sequence coding for the gene. The RNA copy carries the information needed to create the protein. The protein is then obtained by the translation of the RNA transcripts, or messenger RNA (mRNA), into a sequence of amino acids, the base unit of the protein. Then post-translational modifications can further define the activity of the protein. Proteins have a function, and a structure, they are very complex and insightful sources of information for the cells.

In this paper, we will mostly analyze omics modalities from transcriptomics and proteomics. Transcriptomics is the study of the transcriptome, the set of all RNA transcripts. Our transcriptomics data is collected with scRNA-seq, single-cell RNA sequencing. When scRNA-seq is performed on a batch of cells, we obtain a matrix that contains, for every gene considered, the count of its corresponding mRNA for each cell analyzed. The second modality, proteomics, is the study of proteins. For this, we will use the protein abundance measured by mass spectrometry. As the protein have different functions in the cells knowing their abundance can give important information. The data matrix will contain for every gene the amount of the corresponding protein.[21]

## 2.2 Data

The first concept for the analysis of the data is its variability. The variability is what we want the observe and analyze. The interesting genes are those which vary.

To characterize this variability, three types of variability must be distinguished. Technical variability is due to measurement errors and inaccuracies caused by the imperfections of the technologies used to extract the information. Biological variability is the actual variability of gene expression. And finally, the total variability, which is the sum of the two previous ones, is the variability that is measured. We are not interested in technical variability, ideally, it should be zero. What we are interested in is biological variability, because the genes that have great biological variability are those that react, and are modified following a biological event. For the analysis to make sense, the biological variability must be greater than the technical variability.

Multiple big challenges arise from the integration of multi-omics data. Firstly, the collection of the data is not perfect and induced some technical variability in the data. As the techniques used to extract the data of the different modalities are different, there are discrepancies between the technical variability is each data type. Removing this variability is a challenge.

To cope with those challenges, different types of multi-omics data integrations were defined and discussed in [1, 21, 32]:

- **The horizontal integration** corresponds to the joint analysis of the same modalities from independent batches of cells. Therefore the features are shared between data matrices but not the cells. With this strategy, technical variability can be identified. The heterogeneity between the samples can come from the experimental protocols, the platforms, or the tissues that generated the data. Even though the heterogeneity should be removed when it is technical variability, heterogeneity, as discussed in [21], is often an important source of information. For example, if the cells are collected from different tissues, the tissue will contain some tissue-specific bio-marker that is a very important source of information that correspond to biological variability.

- **The vertical integration** is the process of analyzing together different modalities from the same data batch. The cells are shared between data matrices but not the features. Jointly analyzing different layers of genomics information in the same sample can give a good insight into the activity and the operation of the cells. However, depending on the data types (i.e. proteomics data, transcriptomics data, ..) and the biological question that needs to be answered, a lot of different approaches can be designed.

As discussed in [13], horizontal and vertical integration methods have some restrictions. While horizontal integration is restricted to analyzing the same modalities, vertical integration is restricted to analyzing a common set of genes or features shared in all the data-set. In many cases integrating a dataset with

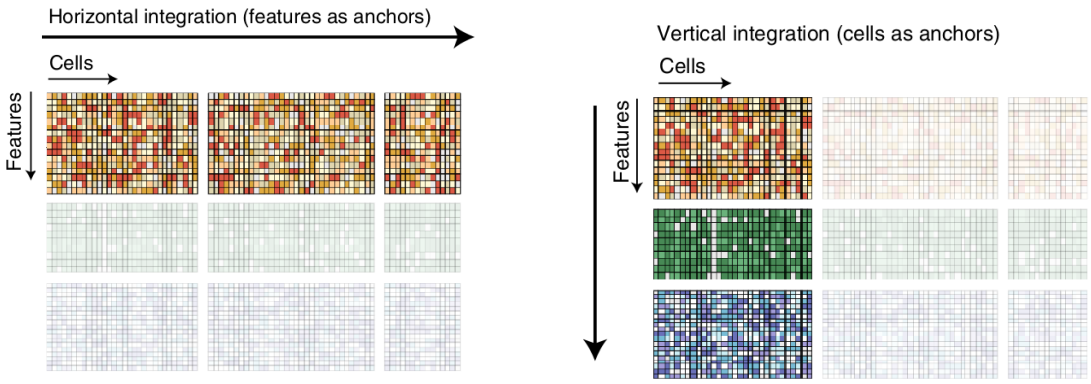


Figure 2.1: Horizontal and vertical integrations[1]

neither the same number of features nor the same number of samples can give even more insight into the organisms. By restricting the analysis to shared features, some very important modality-specific information is not taken into account. For example, the scRNA-seq contains many more features than the protein abundance modality. Restricting to only shared features does not benefit from the higher resolution provided by scRNA-seq.

- **The diagonal integration** is the case where neither cells nor features are shared between data batches. Both data set contains different features and different samples.
- **The mosaic integration** is the most general integration technique. It generalizes each of the other techniques, i.e. horizontal, vertical, and diagonal integration. In mosaic integration, datasets can either share all, some, or none features, and all, some, or none samples can be from the same batches.

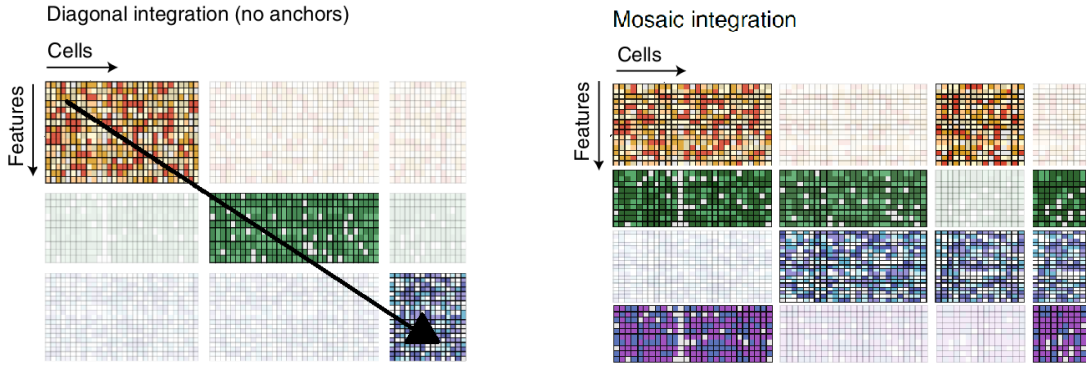


Figure 2.2: Diagonal and mosaic integrations[2]

## 2.3 Single-cell technologies

Until recently, the cells were mostly analyzed by bulk technologies that provide an averaged signal of a sample. A sample contains a tremendous number of cells, averaging all the cells of the sample will result in aggregating different types of cells together. For example, in a tumor, bulk technologies samples would contain cancerous cells as well as immune system cells and other cells. Tumors consist of a multitude of cellular types [6, 22] and analyzing the cells averaged together is much harder to interpret than analyzing each cell individually. By averaging, bulk approaches make the assumption of cellular homogeneity, which is a big limitation.

Single cells offer a completely new way of analyzing cells. Working cell-wise enable the possibility to account for cellular heterogeneity and to get a closer view of cellular dynamics. Single cells offer some promising perspectives for the future. Indeed with the unprecedented resolution they provide, we can expect to understand a lot of processes that were still considered a black box until now[4, 18, 25].

Even if the current single-cell technologies cannot measure all aspects of the cells and all relevant information. Emerging technologies can capture multiple omics data from the same cells. Combining all the information collected by all the different technologies creates a lot of different and complex data integration challenges [13].

# Chapter 3

# Matrix Factorization Background

In this section, we will present the theoretical knowledge that will lead to the algorithm in section 4. We will talk about matrix factorization and machine learning principles that will be useful for the analysis of the algorithm. Finally, we will conclude with a presentation of the state of the art of factorization methods.

A machine learning algorithm is a computational process that uses input data to achieve the desired task without being explicitly programmed to produce a particular outcome. The algorithm can have various numerical parameters that are adjusted and optimized iteratively[5]. Techniques based on machine learning can be applied to diverse fields like computational biology for biomedical and medical applications and that can do a large variety of tasks.

In our case, we want our analyses to enlighten patterns and clusters among the multi-omics data set. To address this problem we can use clustering. Clustering assigns each observation to one of the $k$ groups based on some similarity criteria. Clustering can be used to identify a group of cells showing similar patterns in their expression for a set of genes.

## 3.1 Nonnegative Matrix Factorization

Non-negative matrix factorization (NMF) has been shown to be useful for clustering. NMF is a powerful tool for data reduction that is widely used to analyse genomics data. The fact that the factorization keeps positive values gives interpretable results. Many real-world data are non-negative and the corresponding hidden components have a physical or biological meaning only when they are non-negative. The principle of NMF is to decompose the datasets into the components that underlie them to discover the structures and to extract hidden information [3, 16].

### 3.1.1 NMF problem

Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ be a non-negative matrix and $k$ the desired lower dimension. The goal of NMF is to find two matrices $\mathbf{W} \in \mathbb{R}^{m \times k}$ and $\mathbf{H} \in \mathbb{R}^{k \times n}$ that have non-negative elements such that $\mathbf{X} \approx \mathbf{WH}$

The matrices $\mathbf{W}$ and $\mathbf{H}$ can be found by solving the following optimization problem :

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{H}} ||\mathbf{X} - \mathbf{W}\ \mathbf{H}||_F^2 \\ &\text{s.t.} \quad \mathbf{W} \geq 0,\ \mathbf{H} \geq 0 \end{aligned} \tag{3.1}$$

- $\mathbf{H}$ : $\mathbf{H} \in \mathbb{R}^{k \times n}$ contains the basic components of the data
- $\mathbf{W}$ : $\mathbf{W} \in \mathbb{R}^{m \times k}$ latent factors associated with the components of H

Thus, NMF approximates each observation, contained in the row of $\mathbf{X}$, by a linear combination of the basic components, contained in the rows of $\mathbf{H}$, weighted by the latent factors, contained in the rows of $\mathbf{W}$ [12, 30].

### 3.1.2 Solving NMF

As discussed in [9, 12, 28], solving the problem of NMF as defined in the equation 3.1 and with respect to both variables $\mathbf{W}$ and $\mathbf{H}$ is a non-convex optimization problem. And it was proven in [9] that finding its global minimum is NP-hard. Therefore a good algorithm is expected to compute a local minimum of the problem.

However even though NMF is a non-convex problem, the objective function is convex with respect to only one of the variables $\mathbf{W}$ and $\mathbf{H}$. In other words, finding the optimal matrix $\mathbf{W}$ (resp. $\mathbf{H}$) with a fixed matrix $\mathbf{H}$ (resp. $\mathbf{W}$) reduces to a convex optimization problem [3].

The NMF problem is nontrivial, there can be multiple ways to solve it, therefor multiple algorithms have been developed.

### 3.1.3 Multiplicative update rule

One of the first and a very popular method was introduced by Lee and Seung [15]. The algorithm, based on multiplicative update rules, is very popular because it is easy to implement and the convergence properties are guaranteed. It also provides a natural and flexible basis and can be intuitively adapted. The multiplicative steps are the following :

$$\mathbf{H} \leftarrow \mathbf{H}\frac{\mathbf{W}^T\mathbf{V}}{\mathbf{W}^T\mathbf{W}\ \mathbf{H}} \quad \mathbf{W} \leftarrow \mathbf{W}\frac{\mathbf{V}\ \mathbf{H}^T}{\mathbf{W}\ \mathbf{H}\ \mathbf{H}^T}$$

The steps were proven to give convergence for the minimization objective. It was proven in [15] that the algorithm guarantees to converge to at least locally optimal solutions. Local minima are the best performance we can expect for the NMF problem as it is not convex. However, one drawback of this method is the overall computation time because of the slow convergence of the multiplicative updating rule.

### 3.1.4 Alternating Least Squares

The ALS method is describe in [3] as the *basic workhorse approach*. As the objective functions with respect to both $\mathbf{W}$ and $\mathbf{H}$ is non-convex but is convex with respect to either $\mathbf{W}$ or $\mathbf{H}$, the algorithm solves iteratively the objective function for one of the variables with the other fixed.

$$\min_{\mathbf{H}} \|\mathbf{X} - \mathbf{WH}\|_F^2, \quad \text{with fixed } \mathbf{W}$$

Then all the negative elements of $\mathbf{H}$ are set to zero or a small positive value.

$$\min_{\mathbf{W}} \left\|\mathbf{X}^T - \mathbf{H}^T\mathbf{W}^T\right\|_F^2, \quad \text{with fixed } \mathbf{H}$$

Then all negative elements of $\mathbf{W}$ are set to zero or a small positive value.

One big drawback of the method, that was pointed out by [3], is that this algorithm does not guarantee to converge to a global minimum, as the NMF problem in non-convex, but neither to a stationary point. It converges to a solution where the objective function stops decreasing. The ALS algorithm as stated above does not provide stable convergence proprieties and, in most cases, gives a sub-optimal solution.

### 3.1.5 Alternating Nonnegative Least Squares

The ANLS framework is another way of solving the NMF problem. It defines one sub-problem for each variable and iteratively minimizes the objective function with respect to one variable at a time as ALS, but with the addition of the non-negativity constraint.

$$\mathbf{W} \leftarrow \arg\min_{\mathbf{W}\geq 0} f(\mathbf{W}, \mathbf{H}) \text{ and } \mathbf{H} \leftarrow \arg\min_{\mathbf{H}\geq 0} f(\mathbf{W}, \mathbf{H})$$

These sub-problems are :

$$\min_{\mathbf{W}\geq 0} \left\|\mathbf{X}^T - \mathbf{H}\mathbf{W}^T\right\|_F^2 \quad \text{and} \quad \min_{\mathbf{H}\geq 0} \left\|\mathbf{X} - \mathbf{W}\mathbf{H}^T\right\|_F^2$$

Each sub-problems are Nonnegativity constrained Least Squares (NLS or NNLS) problems. They are convex problems but because of the nonnegativity constraint, they are a nontrivial optimization problems. Multiple approaches have been defined to solve the NLS problem.

The ANLS algorithms have been proven to be theoretically sound and efficient in practice as shown in [11].

### 3.1.6 Initialisation NMF

As the best algorithms can only converge to a local minimum, the initialization of the matrices is very important. The choice of initialization of $\mathbf{W}$ and $\mathbf{H}$ will affect the convergence rate as well as the quality of the solution. A bad initialization will in many cases give slow convergence and will converge to an irrelevant solution. Moreover, the quality of the initialization directly depends on the dataset. A good initialization for a dataset can be a poor one for another dataset. A lot of different types of initialization have been defined.

One initialization technique, named seeding technique, is to iteratively use random starting points. And to perform the factorization with the different initialized matrices $\mathbf{W}$ and $\mathbf{H}$. However as discussed in [7], while this method is very intuitive and simple to implement, the fact that it has to perform multiple factorizations increases significantly the computation time.

### 3.1.7 Unicity of the solution

One issue of NMF is that the solution is non-unique because every invertible matrices $\mathbf{R}$ satisfying

$$\|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2 = \left\|\mathbf{X} - \mathbf{W}\mathbf{R}^{-1}\mathbf{R}\mathbf{H}\right\|_F^2$$

will provide another non-negative factorization $\mathbf{X} \approx \tilde{\mathbf{W}}\tilde{\mathbf{H}}$ [3, 24].

To overcome this issue, some techniques were discussed in [3, 28] to obtain a solution unique, up to permutation and scaling :

- Normalization of the data $\mathbf{X}$
- Normalization of the columns of $\mathbf{W}$ and/or the rows of $\mathbf{H}$ to unit length.

- Adding constraints of sparsity and/or smoothness to the factorization.
- Adding constraints as regularization terms to the factorization. It will reduce the degrees of freedom and therefore reduce the set of solutions to promote uniqueness. However to add such constraints some prior knowledge about the problem is required as it will influence the solution, the regularization must be coherent and reflect the characteristics of the issues.

**Regularization**

Regularization is a useful tool to add some a priory knowledge of the problem in the cost function. It can also reduce the degrees of freedom of the problem and therefore enhance uniqueness.

The principle of regularization is to add one or multiple penalty terms such as :

$$\min_{\mathbf{W}\geq 0,\mathbf{H}\geq 0} \left\|\mathbf{X}-\mathbf{W}\mathbf{H}^T\right\|_F^2 + \alpha(\mathbf{W}) + \beta(\mathbf{H}),$$

With $\alpha()$ and $\beta()$, the regularization functions.

The regularization can then take multiple forms. For example, it can use different norms:

- The Frobenius-norm regularization can prevent each elements of $\mathbf{W}$ or $\mathbf{H}$ from getting large absolute values. It was proven in [12] that it can be used to stabilize the BCD methods, therefore it can stabilize the ANLS method.
- The $l_1$-norm regularization will enhance sparsity in the factorized matrices. As mentioned above sparsity enhance uniqueness. It also helps with interpretation and promotes clustering performances as shown in [10].

### 3.1.8 Large-scale NMF

In many applications of data reduction, for instance, using omics data, the input matrices can be very large and decomposed with a much smaller number of components such that $k << m$ and $k << n$. The problem, in this case, is said to be highly redundant. In this case, using the whole data matrix does not make sense, only using a subset, random or properly chosen will provide a better result for most cases. [3, 28]

## 3.2 jNMF

Joint NMF (jNMF) was developed to go further than NMF. Indeed jNMF integrates multiple datasets with the same principle as NMF, each data set is represented by

a basic components matrix ($\mathbf{H}_i$) and a latent factors matrix ($\mathbf{W}$). However, the integration is made such that the matrix $\mathbf{W}$ is shared among the data set. $\mathbf{W}$ is computed such that it corresponds to the latent factors of all the datasets. And $\mathbf{H}_i$ is the basic components matrix specific to the data-set $i$ for $i \in \{0, d\}$ where $d$ is the number of datasets. However, to make sense, jNMF must be performed on datasets that share their set of observations [30].

$$\begin{aligned} \min_{\mathbf{W},\, \mathbf{H}_1,\ldots,H_d} & \sum_{i=1}^{d} \|X_i - W H_i\|_F^2 \\ \text{s.t.} \quad & W \geq 0, \\ & H_k \geq 0, \quad i = 1, \ldots, d \end{aligned}$$

## 3.3 iNMF

Yang and Michailidis [30] derived, from the algorithms NMF and jNMF, the integrative NMF (iNMF) that uses multiple types of data sources to capture the heterogeneous signal.

The idea behind this algorithm was that in multiple datasets, where the assumption of homogeneity holds. A common integration challenge is the elimination of technical variability. In those cases, the signal of interest is usually the homogeneous signal, i.e. the signal common among all sources, while the heterogeneous signal, i.e. the signal that is different across sources, is unwanted. The purpose of iNMF is to separate the homogeneous and heterogeneous effects among the sources. To do so, iNMF considers homogeneous effects $\mathbf{W}\mathbf{H}_i$ and heterogeneous effects $\mathbf{V}_i\mathbf{H}_i$ for all modalities $i \in 0, d$ in the factorization as follows:

$$\begin{aligned} \min_{\substack{\mathbf{W},\mathbf{H}_1,\ldots,\mathbf{H}_d,\\ \mathbf{V}_1,\ldots,\mathbf{V}_d}} & \textstyle\sum_{i=1}^{d} \|\mathbf{X}_i - (\mathbf{W} + \mathbf{V}_i)\,\mathbf{H}_i\|_F^2 + \lambda \sum_{i=1}^{d} \|\mathbf{V}_i\mathbf{H}_i\|_F^2 \\ \text{s.t.} \quad & \mathbf{W} \geq 0, \quad \mathbf{H}_i \geq 0, \quad \mathbf{V}_i \geq 0, \quad i = 1, \ldots, d \end{aligned}$$

### 3.3.1 Regularization

iNMF takes into account both homogeneous and heterogeneous signals. With the parameter $\lambda$, the heterogeneous effect can be tuned. Indeed, increasing the value of $\lambda$ will result in a higher penalization of the heterogeneous effects[23, 30].

### 3.3.2 Discussion

Yang and Michailidis [30] warn us about some limitations we need to have in mind while analysing the data. Indeed, they show that the matrices $\mathbf{W}$ and $\mathbf{V}$ might not perfectly capture respectively the homogeneous and heterogeneous signals. This result comes from the fact that the matrix $\mathbf{W}$ is the element-wise minimum of the approximated factors $\mathbf{W} + \mathbf{V}_i$ for $i \in \{0, d\}$ where $d$ is the number of data matrices. Because of that the matrix $\mathbf{W}$ is overestimated and might capture too much of the signal and therefore the matrices $\mathbf{V}_i$ might not capture enough information.

As the matrix $\mathbf{W}$ tend to be overestimated and the matrices $\mathbf{V}_i$ underestimated, one must be careful when analysing the heterogeneous and homogeneous. They might not be correctly represented in the matrix $\mathbf{W}$ and in the matrices $\mathbf{V}_i$.

The algorithm was introduced in [30] in the context of horizontal integration, used to remove heterogeneity between samples assumed homogeneous. However, it can also be used for vertical integration, to distinguish homogeneous signals from heterogeneous signals in datasets that are not assumed homogeneous. In many cases, both the similarities and differences are biologically important and therefore both types of signals are of interest. To analyse both signals, iNMF can be used to separate the homogeneous and heterogeneous effects among the sources. [29]

Reviews of iNMF [17, 23], discussed that iNMF outperforms jNMf and that iNMF had successfully identified patterns and clusters in multiple studies. However, Subramanian and Verma [23] suggested doing multiple runs of the optimization as the objective function is non-convex and therefore cannot commit to finding the global minimum. Indeed doing multiple runs with different parameters, different values for the regularization, and different initializations can provide a better understanding of the problem and its sensitivity.

## 3.4 Overfitting

Another possible issue pointed out by Mirza and Wang [20], is that when integrating a multi-omics dataset, the number of variables or features to study is increased, while the number of samples can be the same if the measurements are made in the same biological sample. If the number of variables gets higher than the number of samples the algorithm can suffer from the *curse of dimensionality*, a common in machine learning issue. The increase of dimensionality of the variables makes most machine learning techniques vulnerable to an overfitting problem. Overfitting problems lead to poor generalization capacity from observed data to new data [27, 31].

## 3.5 Features Selection

A widely used solution to avoid overfitting and reducing the high dimensionality of a problem is feature selection. Selecting a smaller subset of the feature, randomly or thoughtfully chosen, can be a solution to improve the quality of a machine learning application.

Moreover, the accuracy of the algorithms was proven to be reduced when some features are irrelevant. Deleting the irrelevant or redundant features improves the accuracy and also reduces the complexity of the problem[27]

# Chapter 4

# UINMF

The UINMF algorithm is an extension of the integrative nonnegative matrix factorization that accounts for both shared and unshared features. The goal of UINMF is to integrate multi-omics data-set even if they do not have the same number of features and/ or the same number of samples. UINMF does not require prior knowledge about the shared and unshared features.

However the samples do not have to contain only different features, indeed they have to have a common set of features. UINMF is therefore a type of mosaic integration technique. As discussed in Section 2.3, allowing the data matrices to have unshared features is very useful when it comes to integrating scRNA-seq and protein abundance as the scRNA-seq modality as a high feature dimensionality and can therefore potentially provide a lot of information and give an ensemble view of the organism. Keeping all information while integrating can yield a higher resolution view of the cells.

The model of the algorithm UINMF as introduced by Kriebel and Welch [13] is as represented on Figure 4.1. UINMF factorizes, for each modality $i$, the data matrices, that are separated into shared features matrices $\mathbf{E}_i$ and unshared features matrices $\mathbf{P}_i$, in block matrices: the latent factors matrices $\mathbf{H}_i$, the basic component of homogeneous signal between the shared features matrix $\mathbf{W}$, the basic component of heterogeneous signal between the shared features matrices $\mathbf{V}_i$ and the basic component of the unshared features matrices $\mathbf{U}_i$.

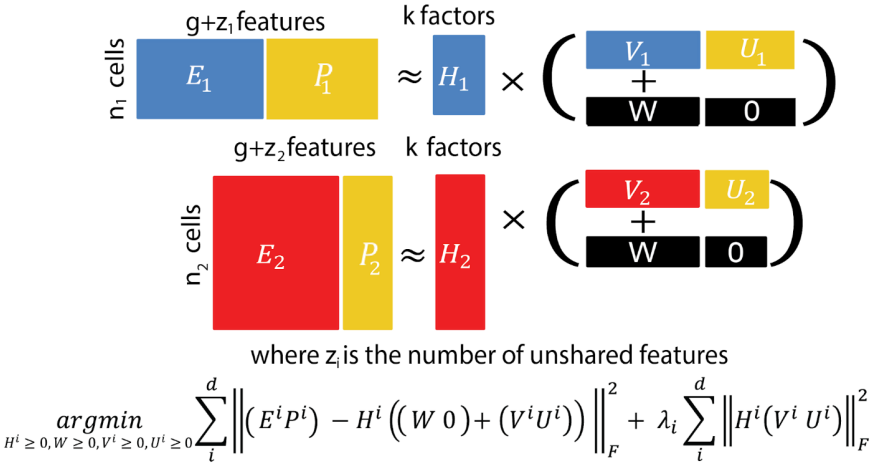


Figure 4.1: Model UINMF[1]

## 4.1 Problem Statement

UINMF is an amelioration of iNMF. The advatage of UINMF is that data from unshared genes can be integrated together with data from shared genes. The unshared features are accounted for in the factorization with the matrices $\mathbf{U}$. The model is the following

$$[\mathbf{E}_i \mid \mathbf{P}_i] \approx \mathbf{H}_i \cdot \left[\begin{array}{c|c} \mathbf{V}_i & \mathbf{U}_i \\ + & + \\ \mathbf{W} & \mathbf{0}_i \end{array}\right]$$

Where

- $\mathbf{E}_i \in \mathbb{R}^{n_i \times g}$: the data matrix for shared features
- $\mathbf{P}_i \in \mathbb{R}^{n_i \times z_i}$: the data matrix for unshared features
- $\mathbf{V}_i \in \mathbb{R}^{k \times g}$ : the basic components matrix for heterogeneous effect of shared features
- $\mathbf{W} \in \mathbb{R}^{k \times g}$ : the basic components matrix for homogeneous effect of shared features (same matrix for all modalities)
- $\mathbf{U}_i \in \mathbb{R}^{k \times z_i}$ : the basic components matrix for unshared features
- $\mathbf{H}_i \in \mathbb{R}^{n_i \times k}$ : the latent factors matrix
- $\mathbf{0}_i \in \mathbb{R}^{k \times z_i}$ : matrix of zero values

With $i \in \{\text{RNA}, \text{Protein}\}$

And where

- $\mathbf{d}$ : number of modalities
- $\mathbf{k}$ : number of factors
- $\mathbf{g}$ : number of shared features
- $\mathbf{z_i}$ : number of unshared features for the modality $i$
- $\mathbf{n_i}$ : number of cell in the dataset of the modality $i$

The objective function of the UINMF problem is the following minimization:

$$\underset{\substack{\mathbf{V}^i \geq 0, \mathbf{U}^i \geq 0 \\ \mathbf{H}^i \geq 0, \mathbf{W} \geq 0,}}{\operatorname{argmin}} \sum_{i}^{d} \left\| \left( \mathbf{E}^i \mathbf{P}^i \right) - \mathbf{H}^i \left( (\mathbf{W0}) + \left( \mathbf{V}^i \mathbf{U}^i \right) \right) \right\|_F^2 + \lambda_i \sum_{i}^{d} \left\| \mathbf{H}^i \left( \mathbf{V}^i \mathbf{U}^i \right) \right\|_F^2 \tag{4.1}$$

## 4.2 Solving UINMF

To solve the UINMF, the method follows the framework of the coordinate block descent (BCD) as defined in [12]. Each matrix of the factorization is a block. The algorithm finds iteratively the optimal matrices by updating each block according to the corresponding sub-problem while holding the other blocks fixed. As each block-wise optimization is convex, the iterative resolution of each sub-problem guarantees to converge to a local minimum.

Each sub-problem is a nonnegative least squares optimization, that is solved with an implementation of the methods described in [11].

## 4.3 Optimize UANLS

The algorithm is implemented in the open-source *LIGER* R package[2] as the function *optimizeUANLS*. To solve the problem 4.2, the problem is divided in an optimization sub-problem with respect to each variables $\mathbf{H}_i$, $\mathbf{W}$, $\mathbf{V}_i$ and $\mathbf{U}_i$ for $i \in (1, \ldots, d)$. Each sub-problems is a nonnegative least-square optimization problem.

The algorithm optimizes for each iteration one of the sub-problems with the others fixed. It iterates until the objective function, that is defined below, ceases to decrease.

[2] `https://github.com/welch-lab/liger`

$$\underset{\substack{\mathbf{V}^i\geq 0,\mathbf{U}^i\geq 0\\ \mathbf{H}^i\geq 0,\mathbf{W}\geq 0}}{\operatorname{argmin}} \sum_i^d \left\|\left(\mathbf{E}^i\mathbf{P}^i\right) - \left((\mathbf{W0}) + \left(\mathbf{V}^i\mathbf{U}^i\right)\right)\mathbf{H}^i\right\|_F^2 + \lambda_i \sum_i^d \left\|\left(\mathbf{V}^i\mathbf{U}^i\right)\mathbf{H}^i\right\|_F^2$$

(4.2)

## Initialization

The factorization matrices are initialized as follows:

$\mathbf{V}$ : shared feature of $\mathbf{k}$ random cells of each modalities

$\mathbf{W}$ : $\mathbf{g} \times \mathbf{k}$ random values uniformly distributed in $[0, 2]$

- $\mathbf{U}$ : unshared feature of $\mathbf{k}$ random cells of each modalities
- $\mathbf{H}$ : $\mathbf{k} \times \mathbf{n_i}$ random values uniformly distributed in $[0, 2]$

The objective function is defined as : $\text{obj}_{\text{train}} = \text{obj}_{\text{approx}} + \text{obj}_{\text{penalty}}$
where :

$$\text{obj}_{\text{approx}} = \textstyle\sum_i^d \left\|((\mathbf{E}_i^0\mathbf{P}_i^0) - ((\mathbf{W}^0\mathbf{0}) + (\mathbf{V}_i^0\mathbf{U}_i^0))\mathbf{H}_i^0\right\|_F^2$$

$$\text{obj}_{\text{penalty}} = \textstyle\sum_i^d \lambda_i \left\|\left(\mathbf{V}_i^0\mathbf{U}_i^0\right)\mathbf{H}_i^0\right\|_F^2$$

## Sub-problems

After the initialization, the algorithm solves the following sub-problems. Each sub-problem corresponds to a non-negative least square (NNLS) problem and is solved with respect to one of the matrices of the factorization, holding the other matrices fixed. The algorithm iteratively performs the following sub-problem optimization until the criterion of convergence is met, as discussed below, or until the maximum number of iterations, defined by *iter*, is reached.

$$\mathbf{H}_i^t = \underset{\mathbf{H}_i^{t-1}\geq 0}{\operatorname{argmin}} \left\| \begin{pmatrix} \begin{pmatrix} \mathbf{W} \\ \mathbf{0}^{(z^i\times k)} \end{pmatrix} + \begin{pmatrix} \mathbf{V}^i \\ \mathbf{U}^i \end{pmatrix} \\ \sqrt{\lambda_i}\begin{pmatrix} \mathbf{V}^i \\ \mathbf{U}^i \end{pmatrix} \end{pmatrix} \mathbf{H}^i - \begin{pmatrix} \mathbf{E}^i\mathbf{P}^i \\ \mathbf{0}^{(g+z^i)\times n_i} \end{pmatrix} \right\|_F^2$$

$$\mathbf{V}_i^t = \underset{\mathbf{V}_i^{t-1} \geq 0}{\operatorname{argmin}} \left\| \begin{pmatrix} \mathbf{H}^{i^T} \\ \sqrt{\lambda_i}\mathbf{H}^{i^T} \end{pmatrix} \mathbf{V}^{i^T} - \begin{pmatrix} \left(\mathbf{E}^i - \mathbf{W}\mathbf{H}^i\right)^T \\ \mathbf{0}^{g \times n_i T} \end{pmatrix} \right\|_F^2$$

$$\mathbf{U}_i^t = \underset{\mathbf{U}_i^{t-1} \geq 0}{\operatorname{argmin}} \left\| \begin{pmatrix} \mathbf{H}^{i^T} \\ \sqrt{\lambda_i}\mathbf{H}^{i^T} \end{pmatrix} \mathbf{U}^{i^T} - \begin{pmatrix} \mathbf{P}^{i^T} \\ \mathbf{0}^{z_i \times n_i T} \end{pmatrix} \right\|_F^2$$

$$\mathbf{W}^t = \underset{\mathbf{W}^{t-1} \geq 0}{\operatorname{argmin}} \left\| \begin{pmatrix} \mathbf{H}^{1^T} \\ \vdots \\ \mathbf{H}^{d^T} \end{pmatrix} \mathbf{W}^T - \begin{pmatrix} \left(\mathbf{E}^1 - \mathbf{V}^1\mathbf{H}^1\right)^T \\ \vdots \\ \left(\mathbf{E}^d - \mathbf{V}^d\mathbf{H}^d\right)^T \end{pmatrix} \right\|_F^2$$

**Increment and convergence check**

$\text{obj}_{\text{train-prev}} = \text{obj}_{\text{train}}$

$\text{obj}_{\text{approx}} = \sum_i^d \left\| \left( \left(\mathbf{E}_i^0 \mathbf{P}_i^0\right) - \left( \left(\mathbf{W}^0 \mathbf{0}\right) + \left(\mathbf{V}_i^0 \mathbf{U}_i^0\right) \right) \mathbf{H}_i^0 \right\|_F^2$

$\text{obj}_{\text{penalty}} = \sum_i^d \lambda_i \left\| \left(\mathbf{V}_i^0 \mathbf{U}_i^0\right) \mathbf{H}_i^0 \right\|_F^2$

$\text{obj}_{\text{train}} = \text{obj}_{\text{approx}} + \text{obj}_{\text{penalty}}$

The algorithm continues to iterate if $t \leftarrow t + 1$ is inferior than *iter* and if the convergence criterion is not met :

$$\text{delta} = \frac{abs(\text{obj}_{\text{train-prev}} - \text{obj}_{\text{train}})}{mean(\text{obj}_{\text{train-prev}}, \text{obj}_{\text{train}})} \leq \text{tolerance}$$

## 4.4 Scaling and normalization

The UINMF algorithm as defined in [13] requires the data to be scaled and normalized. The normalization step divides each value of each sample (the columns in the algorithm) by the sum of all values of the sample. Then the scaling step scales the samples to unit variance. As the data must be non-negative the data is not centered around zero.

# Chapter 5

# Practical analyis of UINMF

In this section, we will explore the model UINMF by performing multiple experiments with basic modelizations to see the limits and strengths of the algorithm. The algorithm we test is the one introduced in [13]. We work all through this paper with the implantation of the algorithm from the R Library *LIGER*.

We will first get some intuition about the model and the factor matrices obtained. Then we will define a framework model. We will then show the importance of scaling and normalization. Then we will test the sensitivity of the initialization to different values of the seed. Then we will talk about the effects of regularization and the different solutions that are computed for different regularization parameters. And finally, we will test the algorithm with unbalanced modalities, modalities with different numbers of features, and different sizes of samples.

The model of the algorithm UINMF as introduced by Kriebel and Welch in [13], and as presented in Chapter 4 is represented in Figure 5.1.

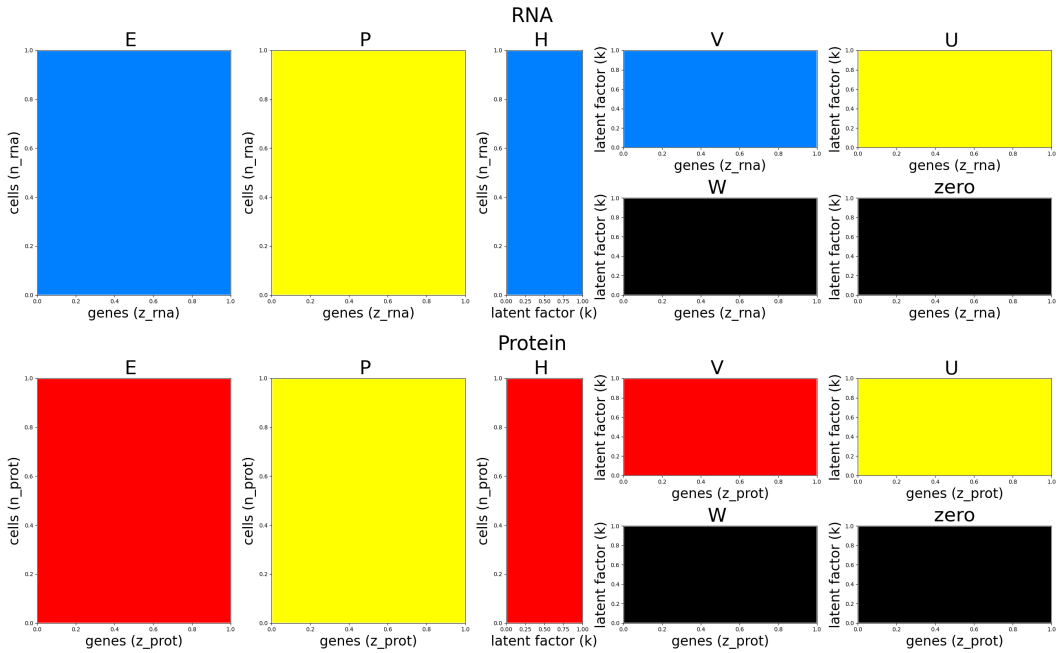


Figure 5.1: Template of the modelization for the UINMF model for scRNA-seq and protein abundance modalities

Figure 5.1 defines the template that will be used for the modelization of the next sections. For a given modality, the data matrix is expressed as the concatenation of the matrix $\mathbf{E}$ and the matrix $\mathbf{P}$. The matrix $\mathbf{E}$ contains the shared features and $\mathbf{P}$ the unshared features. Each modality is described by a data matrix $[\mathbf{E}|\mathbf{P}]$ where for each cell the first $g$, number of shared features, entries of the corresponding row are the values for the shared features then the rest of the row corresponds to the unshared features.
The data matrix is then factorized as the product of the matrix $\mathbf{H}$ and the concatenation the matrix $(\mathbf{V} + \mathbf{W})$, resulting from the element-wise sum of $\mathbf{V}$ and $\mathbf{W}$, with the matrix $\mathbf{U}$.

## 5.1 Notation

Through all this section, the notation $\mathbf{H}$ will refer to both matrices $\mathbf{H}_{\text{rna}}$ and $\mathbf{H}_{\text{protin}}$. The notation used to discuss about the matrix $\mathbf{H}$ for a precise modality will be $\mathbf{H}_{\text{rna}}$ or $\mathbf{H}_{\text{protein}}$ for respectfully the scRNA-seq modality or the protein abundance modality. That will hold for all matrices in the model (except $\mathbf{W}$ that is always uniquely defined by $\mathbf{W}$ as it is shared among the modalities)

## 5.2 Strategy of modalization

The modelizations will consist of two steps: the initialization and the factorization. We will present two types of figures to illustrate the modalizations, the *initialization* figures and the *result* figures that will show the result of the factorization. All figures will follow the template defined by Figure 5.1.

### Initialization

To have better maneuverability and for the modalizations to be easier to understand. We will first model the matrices **H**, **W**, **V** and **U**. By doing so, we will directly see the specificity of the modelization. The initialized factor matrices will be presented on the right-hand side of the *initialization* figures.

Then we will compute the data matrix [**E**|**P**]. By computation of the factor matrices :

$$[\mathbf{E}|\mathbf{P}] = \mathbf{H} \cdot \left[\begin{array}{c|c} \mathbf{V} & \mathbf{U} \\ + & + \\ \mathbf{W} & \mathbf{0} \end{array}\right] \tag{5.1}$$

The data matrices that will be the input of the algorithm will then be expressed, on the left-hand side of the figures *initialization*.

### Results

We will then factorize the data matrices into factor matrices performing UINMF. The resulting matrices will be shown on the left-hand side of the figures *results*.

Finally, we will compute the approximated data matrices by performing the calculation 5.1 on the factor matrices obtained. The approximated matrices will provide a quick idea of the quality of the approximation.

## 5.3 First intuitions

We will start with the modelization of a very simple scenario that will provide some intuition about the different parameters and matrices of the factorization.

### Initialization

We begin this modelization by the initialization of the factor matrices **H**, **V**, **U** and **W**, on the right-hand side of Figure 5.2. The matrices **H** contain three latent factors, each representing a cluster of cells of the size of the third of all cells. The

matrix $\mathbf{W}$ contains three components, each expressing a third of the shared genes. The matrices $\mathbf{V}$ contain only zero value. The entries in the matrices $\mathbf{V}$ can be safely set to zeros because the matirces $\mathbf{V}$ are added to the matrix $\mathbf{W}$, as $\mathbf{W}$ contains non-zeros value the problems will not suffer from ill-conditioning. Finally the matrices $\mathbf{U}$, similarly to the matrix $\mathbf{W}$, contains three components, each with a third of the unshared genes.

After the initialization of the factor matrices, the data matrices $[\mathbf{E}_{\text{initial}}|\mathbf{P}_{\text{initial}}]$, on the left-hand side of Figure 5.2 are created by computing

$$[\mathbf{E}_{\text{initial}}|\mathbf{P}_{\text{initial}}] = \mathbf{H} \cdot \left[\begin{array}{c|c} \mathbf{V} & \mathbf{U} \\ + & + \\ \mathbf{W} & \mathbf{0} \end{array}\right] \tag{5.2}$$

We can see that the data matrices $\mathbf{E}_{\text{initial}}$ and $\mathbf{P}_{\text{initial}}$ contain for both modalities three groups of cells, each of the groups expresses a third of the set of genes.

### Factorization

After that, we perform the integration. The first step is the scaling and normalization of the data matrices. Indeed as discussed in Section 3.1.7, and as we will demonstrate in Section 5.5, normalization is an important step that enhances uniqueness and prevents irrelevant solutions. The scaled and normalized matrices are then factorized by the algorithm UINMF.

In Figure 5.3, we can see the initial data matrices $\mathbf{E}_{\text{initial}}$ and $\mathbf{P}_{\text{initial}}$ on the right-hand side and the data matrices after scaling and normalization $\mathbf{E}_{\text{scaled}}$ and $\mathbf{P}_{\text{scaled}}$ in the middle of the figure.

### Results

The result of this factorization can be seen on the right-hand side of Figure 5.4. We can observe that the algorithm finds successfully the three clusters. Indeed the three factors in $\mathbf{H}$ successfully represent the three cell clusters. And the matrices $\mathbf{W}$, $\mathbf{U}$ and $\mathbf{V}$ successfully represent the corresponding basic components for the corresponding genes.

We can finally observe on the left hand side of Figure 5.4 the approximated matrices $\mathbf{E}_{\text{approx}}$ and $\mathbf{P}_{\text{approx}}$. They were obtained by performing the same computation as the one made to obtain $\mathbf{E}_{\text{initial}}$ and $\mathbf{P}_{\text{initial}}$, but with the matrices $\mathbf{H}$, $\mathbf{W}$, $\mathbf{U}$ and $\mathbf{V}$ calculated by UINMF. The approximated matrices $\mathbf{E}_{\text{approx}}$ and $\mathbf{P}_{\text{approx}}$ are also compared to the initial data matrix and scaled data matrix on Figure 5.3.

$$[\mathbf{E}_{\text{approx}}|\mathbf{P}_{\text{approx}}] = \mathbf{H} \cdot \left[\begin{array}{c|c} \mathbf{V} & \mathbf{U} \\ + & + \\ \mathbf{W} & \mathbf{0} \end{array}\right] \tag{5.3}$$

We can see that they show the same cells cluster as the one represented in $\mathbf{E}_{\text{init}}$ and $\mathbf{P}_{\text{init}}$.

As we said previously the solution is unique up the scaling and rotation. We can here easily understand the definition of uniqueness up to permutation and scaling. As defined in [14], for NMF :

*A matrix* $\boldsymbol{X}$ *has a unique NMF factorization* $\boldsymbol{X} \approx \boldsymbol{WH}$ *if the ambiguity is a permutation and a scaling of the columns in* $\boldsymbol{W}$ *and rows in* $\boldsymbol{H}$.

We can intuitively expect the same result for UINMF. In this example, we can observe that even though the columns of $\mathbf{H}$ and the rows of $\mathbf{W}$, $\mathbf{V}$ and $\mathbf{U}$ are permuted, the product of the factorized matrices leads to the same matrices $\mathbf{E}_{\text{approx}}$ and $\mathbf{P}_{\text{approx}}$. It must be noted that the columns of $\mathbf{H}$ and the rows of $\mathbf{W}$, $\mathbf{V}$ and $\mathbf{U}$ represented in the figures are permuted for the sake of faster intuition about the result of the factorization.

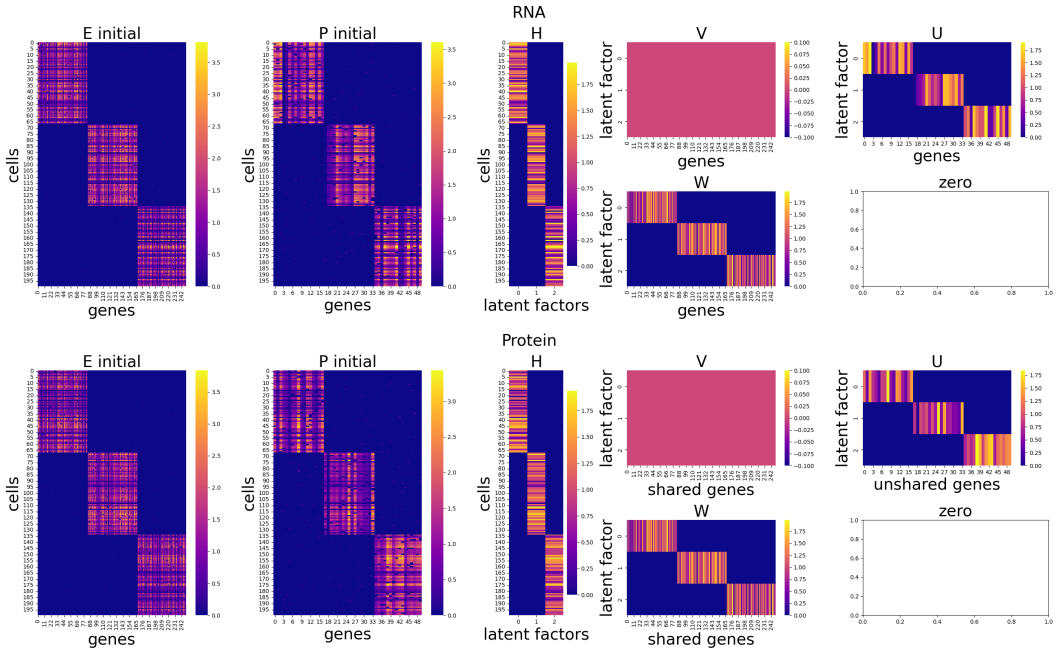


Figure 5.2: Initial matrices for the basic modalization for the scRNA-seq and protein abundance modalities

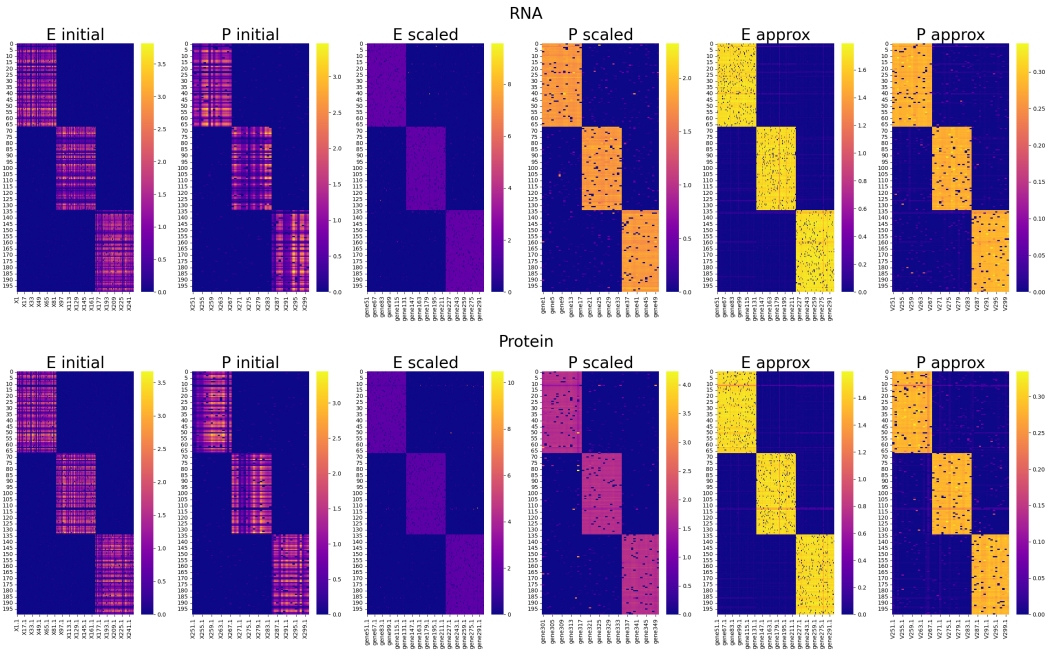


Figure 5.3: **E** and **P** matrices initial, scaled and approximated

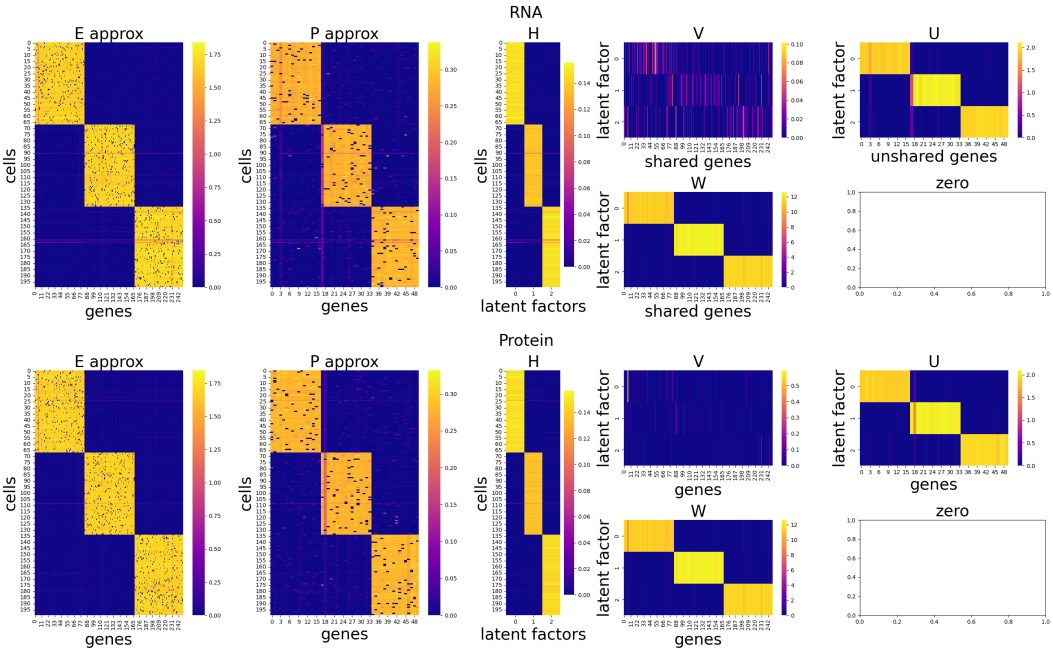


Figure 5.4: Matrices obtained by UINMF factorization for the basic modelization for the scRNA-seq and protein abundance modalities

### 5.3.1 Getting to know the factor matrices

In this section, a few modelizations are presented to get acquainted with the factor matrices and what they represent. We will initialize differently the factor matrices to understand the effect on the data matrices. Therefore we will understand the effect and the scope of the data matrices and understand what they represent.

We will start the modelization with the same model as the one defined in the previous section and tune some parameters to understand their effects and their scopes.

#### Initialization H : latent factors for all the other matrices

The matrices **H** are the latent factors matrix. It multiplies the blocks that contains the matrices **W**, **V** and **U**. Therefore it represents the hidden component of the whole dataset. Its scope is both the shared and unshared features.

The matrices **H**, on the right-hand side of Figure 5.5, represent three clusters of cells. However, the matrix $\mathbf{H}_{\text{rna}}$ has one large cluster and two small clusters. The matrices **W** and **U** represent the corresponding component for the genes of each cell cluster, a third of the genes set is expressed by each components. As the entries of both matrices **V** are empty, there is no heterogeneous signal for both modalities.

On the left-hand side of Figure 5.5, we can see the corresponding data matrices, they result from the calculation of the equation 5.2 with the initialized factor matrices. We can observe as expected three groups of cells are defined for each modality, with one larger than the two others for the scRNA-seq modality. Each of the groups expresses a third of the set of genes for both modalities.

We can add that the change in the matrix $\mathbf{H}_{\text{rna}}$ affects both matrices $\mathbf{E}_{\text{rna}}$ and $\mathbf{P}_{\text{rna}}$. Indeed the latent factors of the matrices **H** are the latent factors for both shared and unshared features, a larger cluster expressed in the matrices **H** represent larger clusters in the matrices $[\mathbf{E}|\mathbf{P}]$ of both modalities.

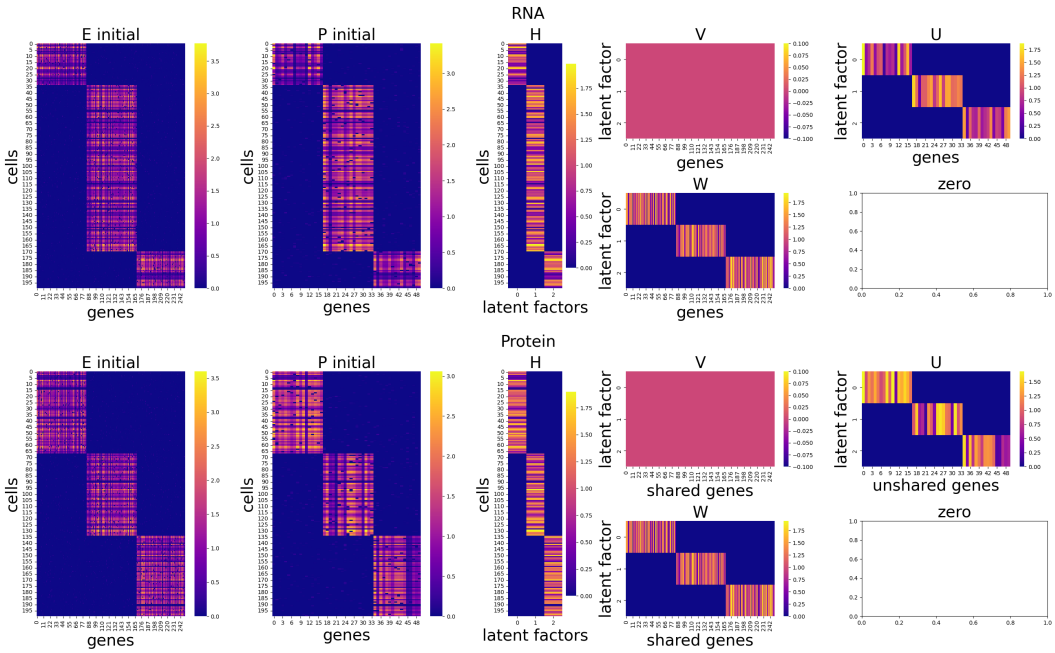


Figure 5.5: Initial matrices for the scRNA-seq and protein abundance modalities

### Initialization W : Homogeneous effect shared by both modalities

The matrix **W** is the basic components matrix that captures the homogeneous signal of shared features. The matrix **W** is shared in the factorization of all datasets. Its scope is the shared features data matrices **E** of both modalities.

On the right-hand side of Figure 5.6, we can see the matrix **W** was initialized representing components that express a fraction of the features, the genes, with one component that has nonzero values for a larger number of genes that the two other components.

The corresponding calculated data matrices, on the left-hand side of Figure 5.6 show three groups of cells, with one group expressing a larger fraction of the set of genes for both modalities. We can also say that **W** only effects the shared feature matrices $\mathbf{E}_{\text{rna}}$ and $\mathbf{E}_{\text{protein}}$ and does not affects the matrices $\mathbf{P}_{\text{rna}}$ and $\mathbf{P}_{\text{protein}}$.

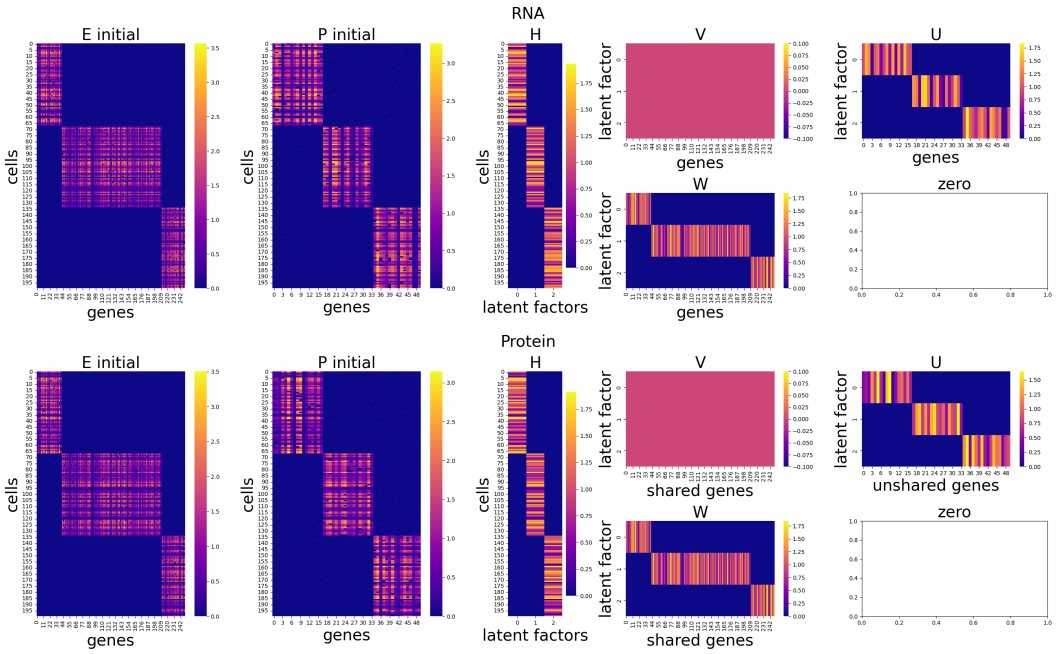


Figure 5.6: Initial matrices for the scRNA-seq and protein abundance modalities

### Initialization V : Heterogeneous effect specific to the modalities

The matrix $\mathbf{W}$ is the basic components matrix that captures the homogeneous signal of shared features. The matrix $\mathbf{W}$ is shared in the factorization of all datasets. Its scope is the shared features data matrices $\mathbf{E}$ of both modalities.

On the right-hand side of Figure 5.7, we can see that the scRNA-seq modality is initialized with some heterogeneous signal, represented in the matrix $\mathbf{V}_{\text{rna}}$.

The corresponding calculated data matrices are on the left-hand side of Figure 5.7. We can see for the matrices $\mathbf{E}_{\text{protein}}$, $\mathbf{P}_{\text{protein}}$ as well as $\mathbf{E}_{\text{rna}}$ three groups of cells expressing each a thirds of the genes. For the shared features matrix of the scRNA-seq modality, $\mathbf{E}_{\text{rna}}$, we have three groups of cells. However, one group expresses an additional third of the genes.

### Results V : Heterogeneous effect specific to the modalities

After their initialization, the data matrices have been scaled then normalized, and afterwards factorized with UINMF.

We can see, on the right-hand side of Figure 5.8, that the algorithm has some trouble identifying the heterogeneous noise. Indeed some noise appears in the matrix $\mathbf{H}_{\text{rna}}$ and the matrix $\mathbf{V}_{\text{rna}}$ does not contains the heterogeneous signal as we would expect. We can see that the heterogeneous signal is captured by the matrix

$\mathbf{H}_{\text{rna}}$. The matrix $\mathbf{W}$ is correctly estimated.

On the left-hand side of Figure 5.8 we can see the approximated matrices $\mathbf{E}_{\text{approx}}$ and $\mathbf{P}_{\text{approx}}$ they correspond to the computation of the equation 5.3. We can observe in the matrix $\mathbf{P}_{\text{rna}}$ an unwilling signal, similar to the one in the matrix $\mathbf{E}_{\text{rna}}$ of the scRNA-seq modality that was expected as it was due to the heterogeneous signal in the matrix $\mathbf{V}_{\text{rna}}$. We can observe that this noise is due to the noise in the matrix $\mathbf{H}_{\text{rna}}$. Furthermore, noise appears in the matrix $\mathbf{P}_{\text{rna}}$. Indeed as the matrix $\mathbf{H}_{\text{rna}}$ multiplies the matrix $\mathbf{U}_{\text{rna}}$, the errors in the matrix $\mathbf{H}_{\text{approx}}$ affects the matrix $\mathbf{P}_{\text{approx}}$.

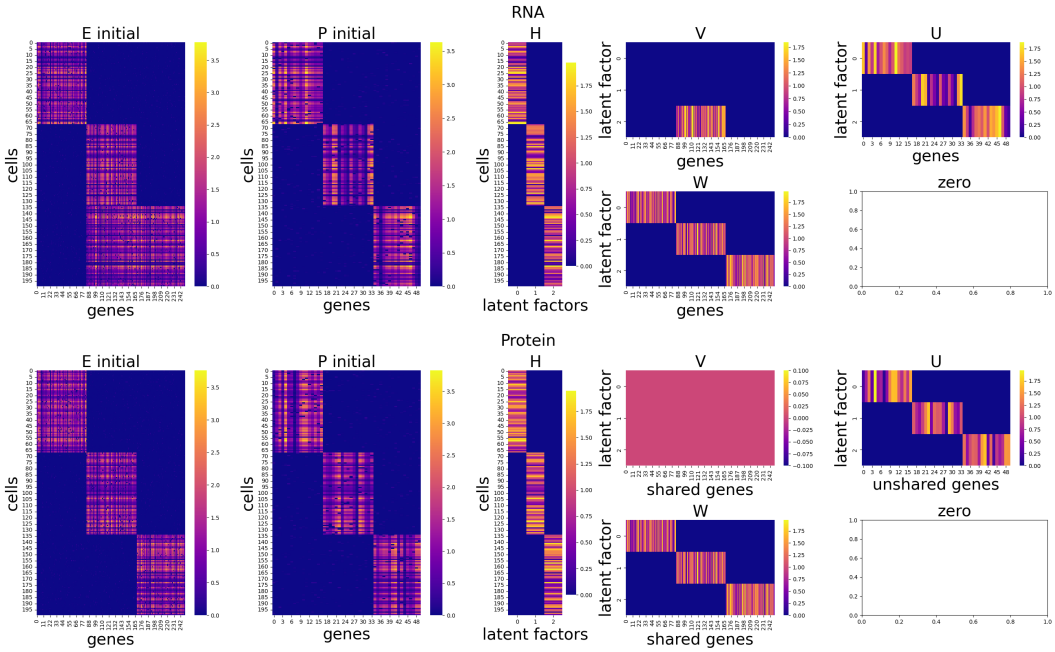


Figure 5.7: Initial matrices for the scRNA-seq and protein abundance modalities

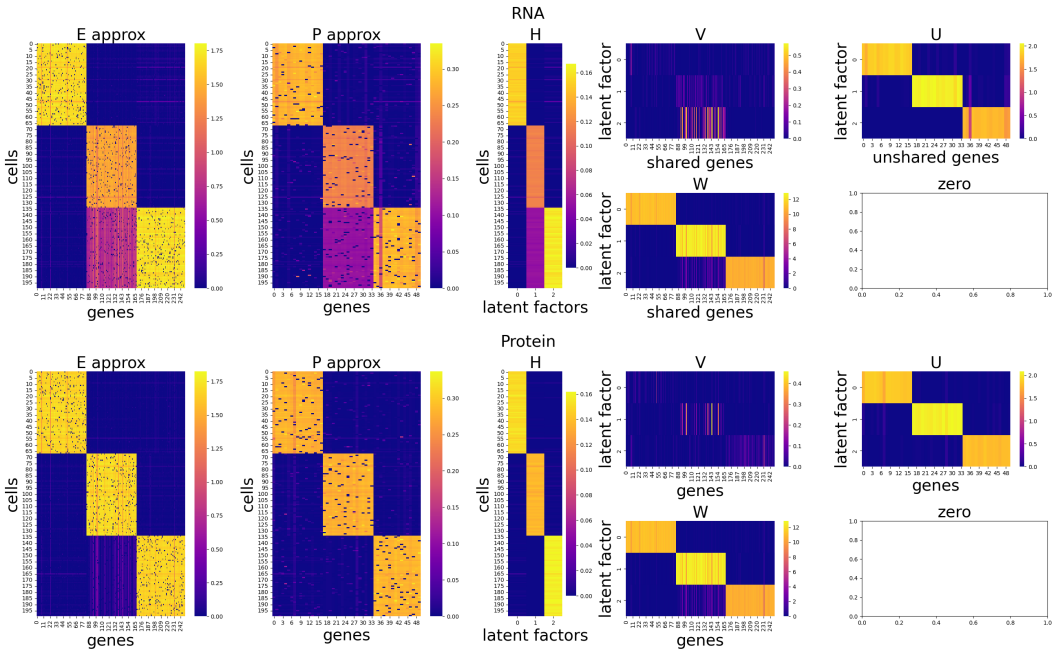


Figure 5.8: Matrices obtained by UINMF factorization for the scRNA-seq and protein abundance modalities

**Initialization U : basic component of unshared features**

The changes in a matrix $\mathbf{U}$ will result in a change in the matrix $\mathbf{P}$. Of the corresponding modalities. The same experience as the one made for the matrix $\mathbf{W}$, defining a component expressing a larger subset of the genes than the two others in one of the matrix $\mathbf{U}$, e.g. $\mathbf{U}_{\text{rna}}$, will lead to a lager value of genes expressed by the corresponding cell group in the corresponding matrix $\mathbf{P}$, e.g. $\mathbf{P}_{\text{rna}}$.

## 5.4 Framework model

In this section, we will define a model inspired by the model defined in [30][1] and that will be the framework for the modelizations in all of the following sections.

### Initialization

The initialization of the model is shown in Figure 5.9, the matrices $\mathbf{H}$, $\mathbf{W}$ ,$\mathbf{V}$ and $\mathbf{U}$ are defined as they were in Section 5.3.

[1]The model is tested in Figure 1 of [30] and is explained in detail in the supplementary notes of the article

The matrices $\mathbf{V}$ contain some heterogeneous noise for both modalities. To make the heterogeneous noise easier to interpret, we make it always the same for all simulations. Another note to make about the noise is that it is strictly non overlapping in the matrices $\mathbf{V}_{\text{rna}}$ and $\mathbf{V}_{\text{protein}}$. Indeed if some signal is in $\mathbf{V}_{\text{rna}}$ as well as in $\mathbf{V}_{\text{protein}}$ it would contradict the assumption that the signal in matrices $\mathbf{V}$ is heterogeneous. Therefore the aforementioned signal would be homogeneous and should be in the matrix $\mathbf{W}$. If the factorization is not realistic it should not be the solution we expect from the optimization problem. Therefore we could not compare the solution with the initialized matrices. For a similar reason, that is however more open to discussion, the matrix $\mathbf{W}$ and both matrices $\mathbf{V}$ have no intersection.

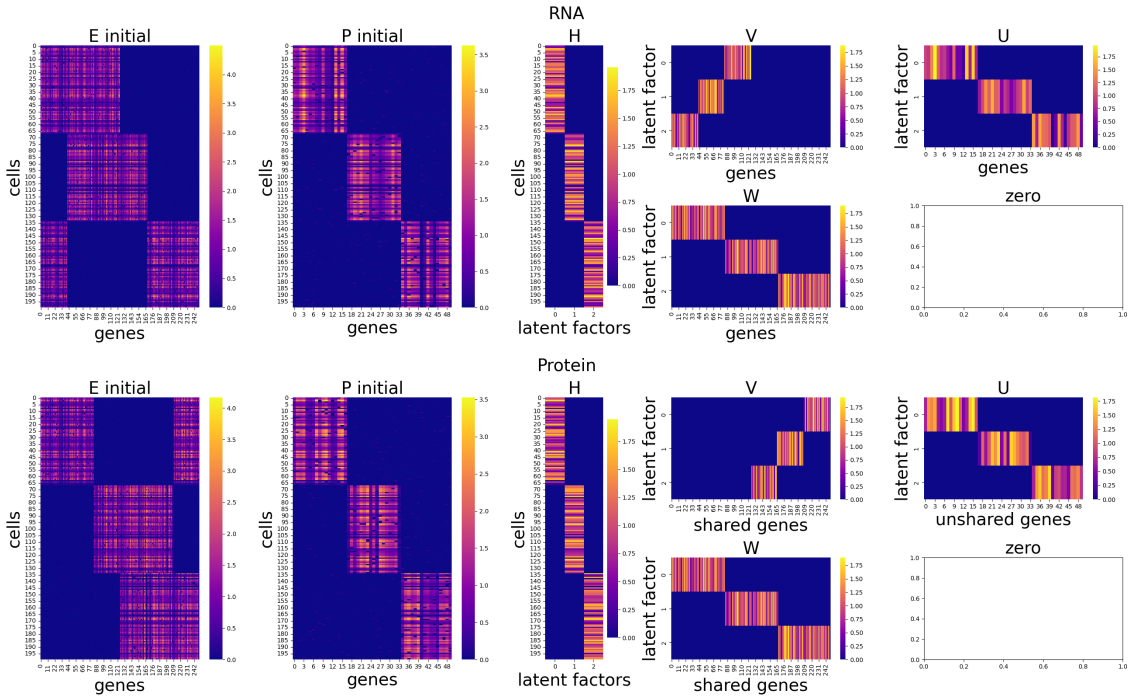


Figure 5.9: Initial matrices for the scRNA-seq and protein abundance modalities for the framework model

## Results

In Figure 5.10, we can see the solution of the factorization for the framework defined above. On the right-hand side, we have the factorized matrices and on the left-hand side, the approximated data matrices are computed with the factorized matrices as in the equation 5.3.

We can see in Figure 5.10 that the factorization does not provide a great solution. Firstly, the matrix $\mathbf{W}$ captures more than the homogeneous effect, it also captures the heterogeneous effects of both matrices $\mathbf{V}$. The matrices $\mathbf{V}$ contains some of the heterogeneous signal defined in the model, but also some other signals. The matrices $\mathbf{H}$ contain some off-diagonal block signals that were not defined. Those noise impact the matrices $\mathbf{P}_{\text{approx}}$. Finally we can see in matrices $\mathbf{E}_{\text{approx}}$ that some addition signal appears in comparison to the initial matrices. This additional signal is mostly induced by the off-diagonal block element of the matrices $\mathbf{H}$ and the fact that the matrix $\mathbf{W}$ captures the heterogeneous signal of both modalities.

A similar result was discussed in [30] and in Section 3.3.2. For the iNMF method, that is defined in Section 3.3, the integration scenarios with heterogeneous signal the matrix $\mathbf{W}$ tends to be overestimated and $\mathbf{V}$ underestimated. We can see here that $\mathbf{V}$ seems to be underestimated.

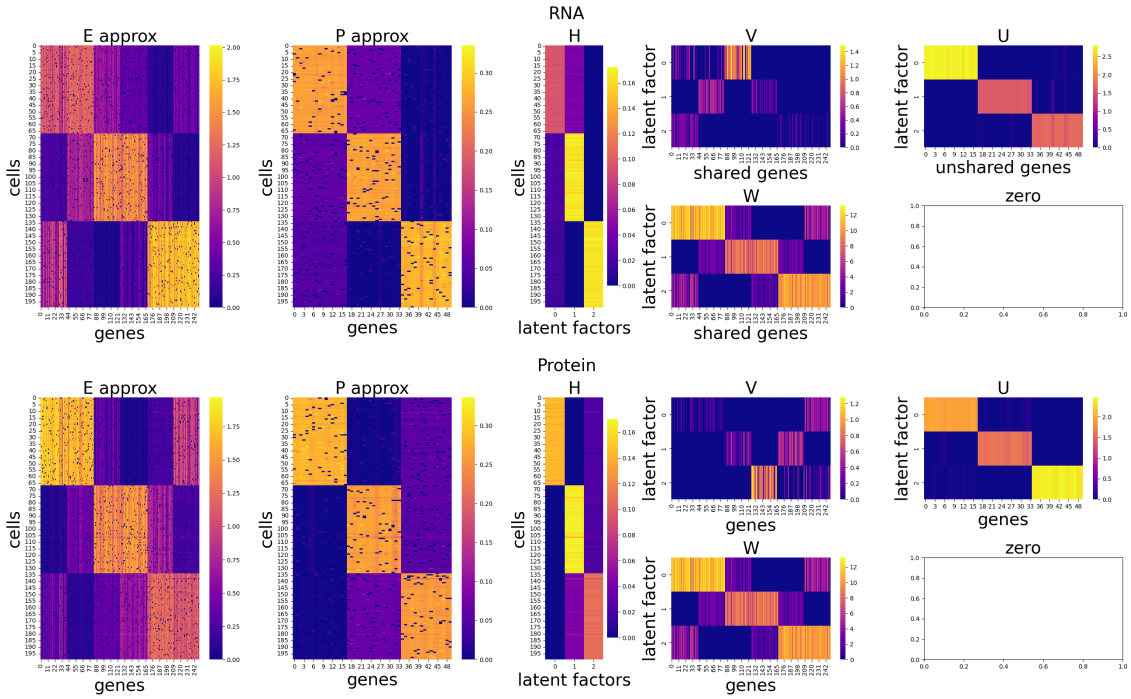


Figure 5.10: Matrices obtained by UINMF factorization for the scRNA-seq and protein abundance modalities for the framework model

## 5.5 Effect of scaling and normalisation

In this section we will discuss the drawbacks and advantages of scaling and normalisation. In the UINMF algorithm the data must be first normalized by dividing each values of a sample by the sum of all values of the sample. Then the samples

are scaled to unit variance. As the data must be non-negative the data is not centered around zero.

## Initialization

To demonstrate the effects of scaling we will do the following modelization. The matrices are initialized as in the framework model defined in Section 5.4 with three cells groups defined and heterogeneous signal, different for scRNA-seq and protein abundance modalities. However each group will show a different intensity of expression of the genes.

## Results

On the left-hand side in Figure 5.11, we can see that the data matrices for the scRNA-seq modality the initial data matrices $\mathbf{E}_{\text{initial}}$ and $\mathbf{P}_{\text{initial}}$. In the middle of Figure 5.11 $\mathbf{E}_{\text{scaled}}$ and $\mathbf{P}_{\text{scaled}}$, are the matrices obtains after the scaling and the normalization of the initial data matrices. They are the input matrices of the algorithm UINMF. On the right-hand of side of the figure are the matrices $\mathbf{E}_{\text{approx}}$ and $\mathbf{P}_{\text{approx}}$. The approximated matrices are calculated from the matrices $\mathbf{H}$, $\mathbf{W}$, $\mathbf{V}$ and $\mathbf{U}$ computed by UINMF.

We can see that the information of intensity of gene expression int the initial data matrices gets lost in the scaling process. This is a limitation of scaling that one has to keep in mind when performing UINMF.

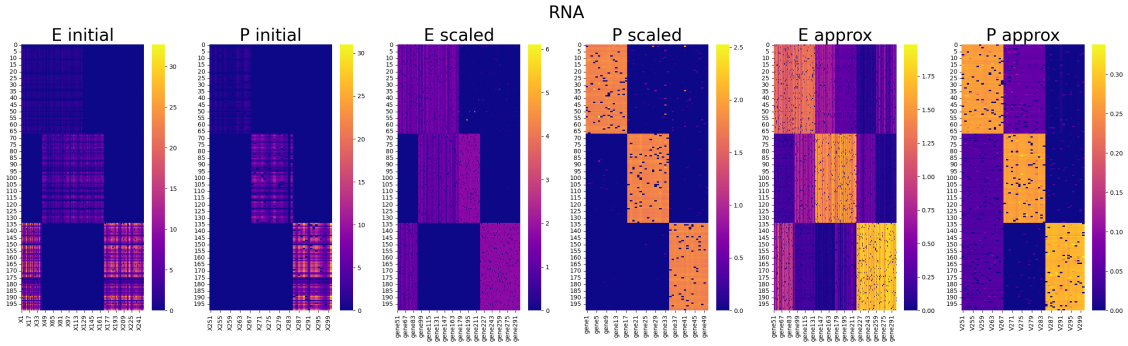


Figure 5.11: $\mathbf{E}$ and $\mathbf{P}$ matrices initial, scaled and approximated

However skipping the scaling of the matrices lead to irrelevant results. Multiple runs were made without scaling and the results are really irrelevant.

We can see in Figure 5.12, the result of the framework modelization for the modality scRNA-seq without scaling and normalization. We can see that all factor matrices are miscalculated

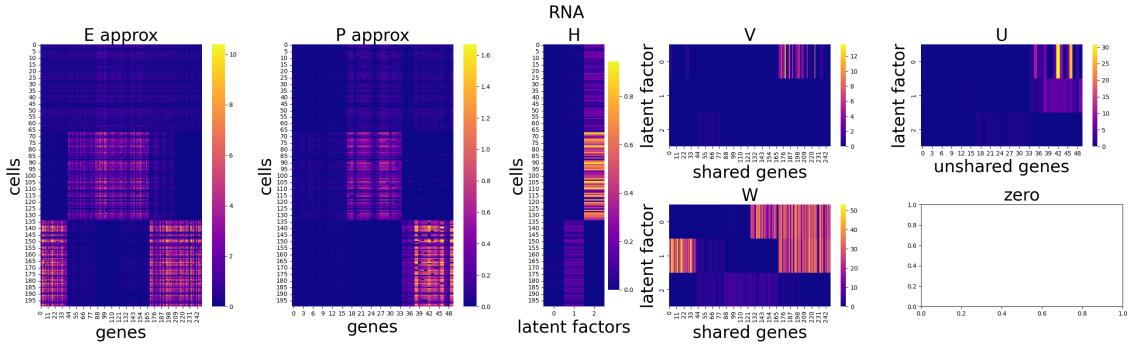


Figure 5.12: Matrices obtained by UINMF factorization for the scRNA-seq and protein abundance modalities for the framework model without scaling

## 5.6 Effect of seeds

Discussed by Wang and Zhang in [28] as an open issue for the NMF problem, the initialization can influence the quality of the solution. For NMF, depending on the initialization of the matrices $\mathbf{W}$ and $\mathbf{V}$, the convergence can lead to different local minima and therefore lead to irrelevant solutions. In UINMF the initialization of $\mathbf{H}$, $\mathbf{W}$, $\mathbf{V}$ and $\mathbf{U}$ can also lead to different solutions. Intuitively, we can say than the fact that the factorization contains more matrices give more degrees of freedom to the problem and will therefore lead to a greater or equal number of local minima.

### Initialization

The algorithm is initialized as discussed in Section 4.3, the matrices $\mathbf{V}$ and $\mathbf{U}$ are initialized with $k$ samples in their rows, where $k$ the number of latent factors. The part of each sample that contains the values for the shared genes are in the matrix $\mathbf{V}$ and the part that contains the unshared genes are in the matrix $\mathbf{U}$.

The matrices $\mathbf{H}$ and $\mathbf{W}$ are initialized with random values following a uniform distribution $\mathcal{U}(0, 2)$. The choice of sample with witch we initialize the matrices is random and depend on a seed value that is defined in the algorithm signature.

To test the robustness against the different choices of samples that will initialize $\mathbf{V}$ and $\mathbf{U}$, multiple runs on the framework modelization are performed with a range of seed values.

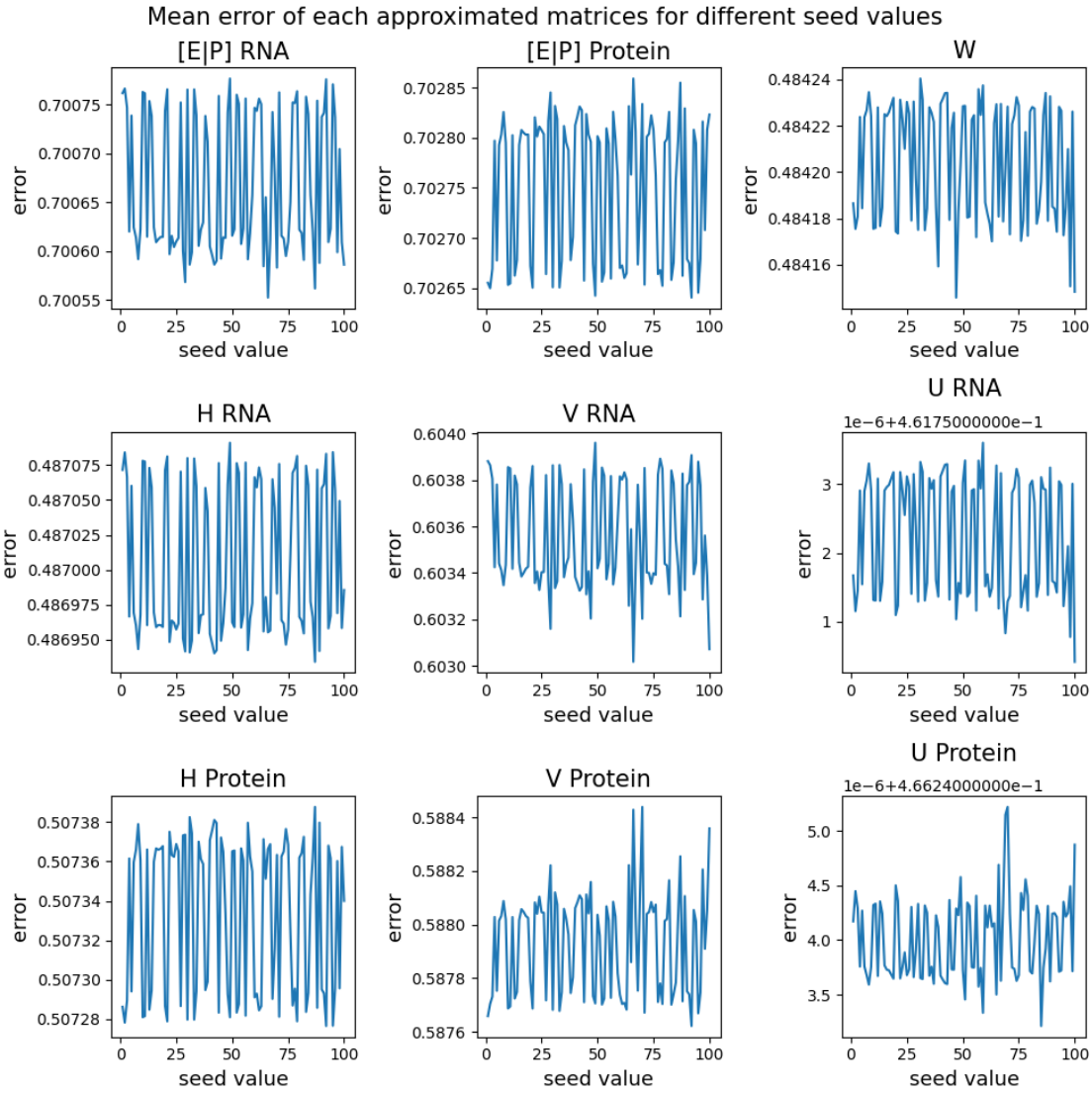


Figure 5.13: Error on each matrix of the UINMF factorization for different values of seed

## Results

In Figure 5.13, we can see for each values of seed the error between each approximated matrix and the corresponding initial scaled matrix for each factor matrices [**E**|**P**], **H**, **W**, **V** and **U** and for each modality. The error is the Frobenius norm of the difference between one input matrix and the corresponding approximated matrices divided by the square root of the number of element of the matrix. Therefore the error is the mean error of the elements of the approximated matrix. The

matrices $\mathbf{X}$, $\mathbf{H}$, $\mathbf{W}$, $\mathbf{V}$ and $\mathbf{U}$ are normalized before they are compared as the factor matrices, $\mathbf{H}$, $\mathbf{W}$, $\mathbf{V}$ and $\mathbf{U}$, are unique up to scaling and permutation. Therefore the factor matrices are also permuted before the computation of the error. They are permuted with the permutation that minimizes the error.

We can see on the figure that the algorithm converges to different solutions that provide very similar errors. This result gives a good sign of local minima. The choice of initialization seems relatively robust to modification of the choice of sample.

## 5.7 Effect of regularization

In this section, we will analyse the effect of regularization on the model. To do so, the algorithm was tested for the framework model defined in Section 5.4. The model is tested with different values for the regularization parameter $\lambda$. In the algorithm, $\lambda$ can either be a value that will be used for the regularization of all modalities or a vector with different values used for the regularization of the different modalities.

### 5.7.1 Uniform regularization

We will start with the regularization with a value of the factor $\lambda$ common for all modalities. The default value proposed by *LIGER* is $\lambda = 5$.

In Figure 5.14, we can see for each of number of sample tested the mean error element-wise, calculated as in Section 5.6, for each factor matrices $\mathbf{H}$, $\mathbf{W}$, $\mathbf{V}$ and $\mathbf{U}$ and for the approximated matrices $[\mathbf{E}|\mathbf{P}]$ obtained with the algorithm.

#### Results

We can observe that with a higher value of regularization, the error in $[\mathbf{E}|\mathbf{P}]$ increases. This very straightforward result as more constraints are added to the problem.

Looking at the error plots of the matrices $[\mathbf{E}|\mathbf{P}]$ , it seems like the lower the regularization term the better. To evaluate is this intuition is right, we will display the result of the factorization for different values of the factor $\lambda$. In Figure 5.15, we can see the values for the matrices $\mathbf{W}$ and $\mathbf{V}$ for the scRNA-seq and protein abundance modalities, and for three different values of the factor $\lambda$.

On the top of the figure are the modelization for $\lambda = \frac{1}{50}$. We can see that $\mathbf{V}$ matrices have most of the signal and contains both shared and unshared signals. On the middle and bottom of the figure the case where $\mathbf{W}$ captures most of the signal is observed, for respectively $\lambda = 4$ and $\lambda = 100$. In both cases the matrix $\mathbf{W}$ and the matrices $\mathbf{V}$ contain some heterogeneous and homogeneous signals.

Two cases are observed, for a very high value of the factor $\lambda$, the matrix $\mathbf{W}$ contains most of the signal, and for a small value of the factor $\lambda$ the signal is shared between $\mathbf{W}$ and that matrices $\mathbf{V}$. However, it is not shared forming a distinct separation of homogeneous and heterogeneous signals. Indeed the matrices contain both some of the heterogeneous and the homogeneous signal.

Therefore very low regularization term does not necessarily provide the most relevant solution even thought they seem to minimize the errors.

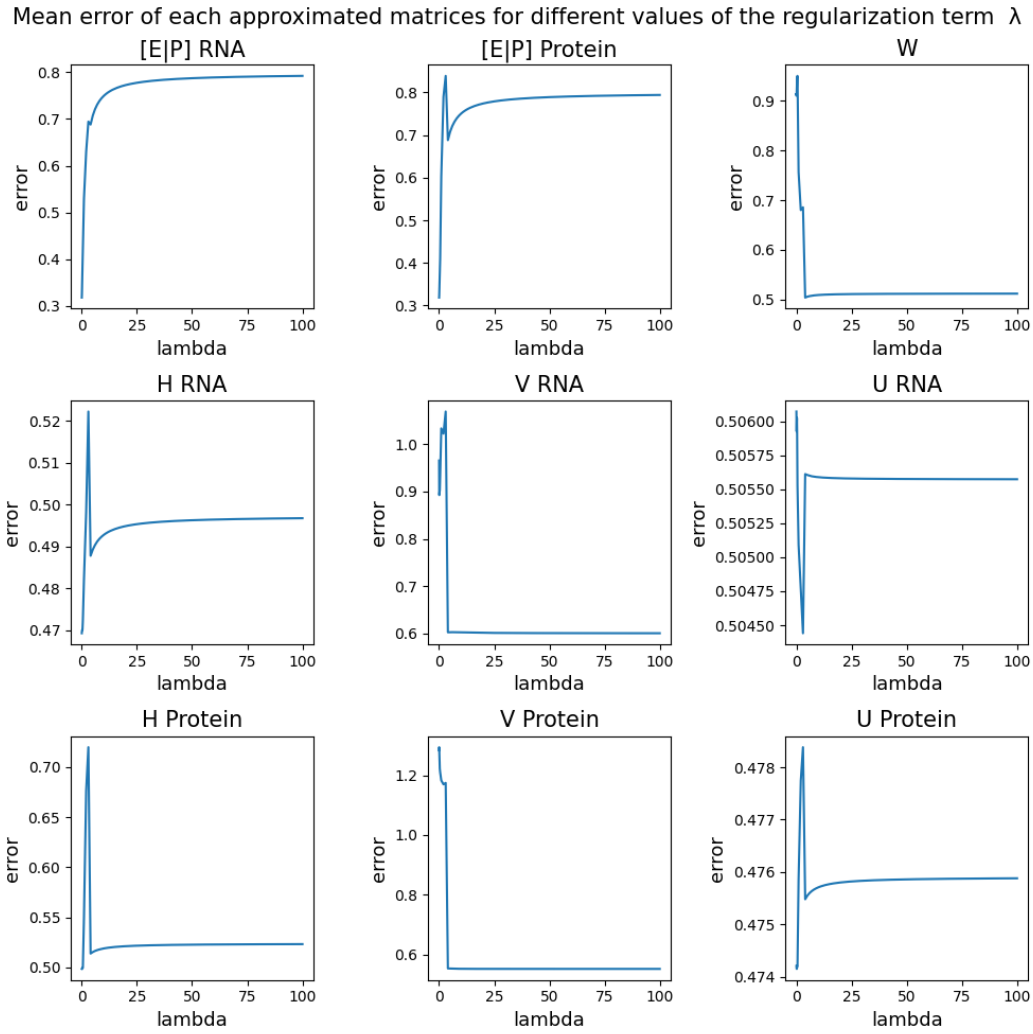


Figure 5.14: Error on each matrix of the UINMF factorization for different values of the regularization term

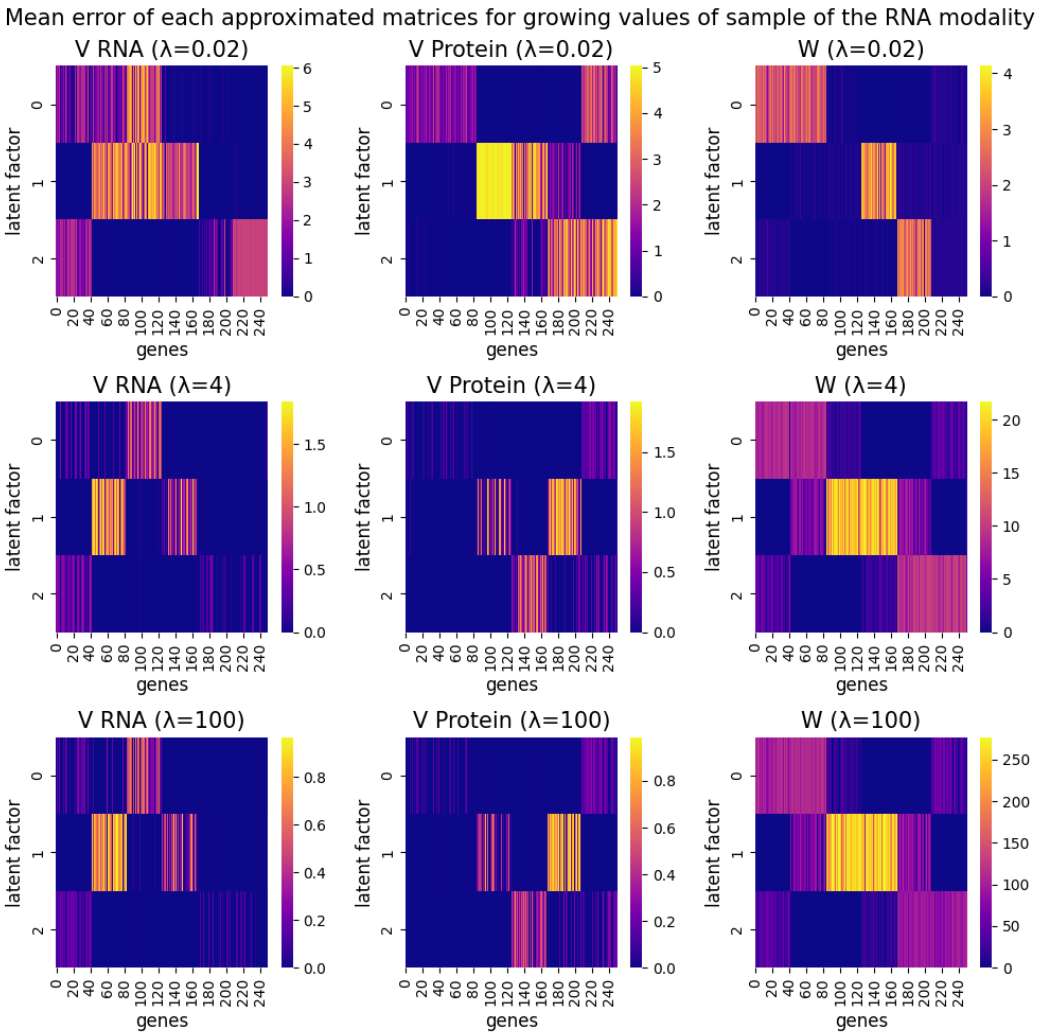


Figure 5.15: Matrices $\mathbf{V}_{\text{RNA}}$, $\mathbf{V}_{\text{Protein}}$ and $\mathbf{W}$ for different values of $\lambda$

### 5.7.2 Modalities specific regularization

We will now discuss the case where the regularization term $\lambda$ is not the same for all modalities. We will test is it can can bring more flexibility to adjust or favor a modality.

#### Initialization

To understand the potential role that the regularization could take we performed the UINMF factorization on the framework model as in the previous subsection but with one modality more regulated that the other.

**Results**

In Figure 5.16, we can see the matrices $\mathbf{V}$ and $\mathbf{W}$ for the uniform regularization with $\lambda = 4$ and for a regularization of $\lambda = 100$ and $\lambda = 4$ for the scRNA-seq modality and protein abundance modality respectively.

We can see that with a higher regularization on the scRNA-seq modality, the factorization gives a relatively better solution for the heterogeneous effects of the protein abundance modality. Modality specific can indeed favor the quality of the solution for one modality at the expense of the other.

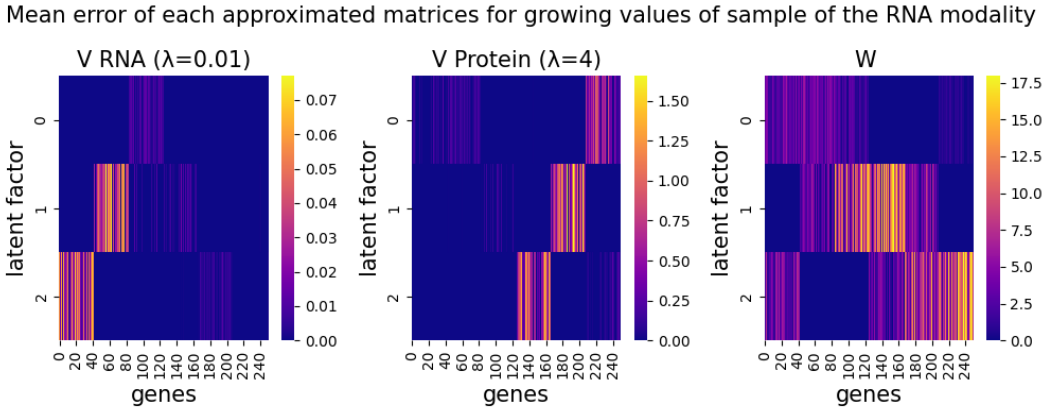


Figure 5.16: Matrices $\mathbf{V}_{\text{rna}}$, $\mathbf{V}_{\text{protein}}$ and $\mathbf{W}$ for modality-specific values of $\lambda$

## 5.8 Effect of unbalanced modalities

In this section we will discuss and analyse the effect of unbalanced modalities, i.e. when the dimensions of the dataset is not the same for both modalities. In the case of the scRNA-seq and protein abundance modalities, the scRNA-seq modality usually have a larger number of features than the protein abundance modality. As discussed in Section 3.4, the high dimensionality of features is a common problem in machine learning. Indeed too many features compared to the number of samples can lead to the curse of dimension problem. As discussed in [26] the amount of sample necessary for a given number of features grows exponentially with the number of features. The number of samples needed to have a simulation that is relevant is therefore much higher for the scRNA-seq modality.

We will in this section discuss about the consequences of a higher number of feature and what happens if a modality have more sample than the other.

### 5.8.1 Unbalanced size of sample

In this subsection we will perform the factorization of the framework model as defined in Section 5.4 with a growing number of sample for the scRNA-seq modality and with all the other variables fixed.

In Figure 5.19, we can see form each of number of sample tested the mean error element-wise for each factor matrices and for the approximated matrices obtained with the algorithm.

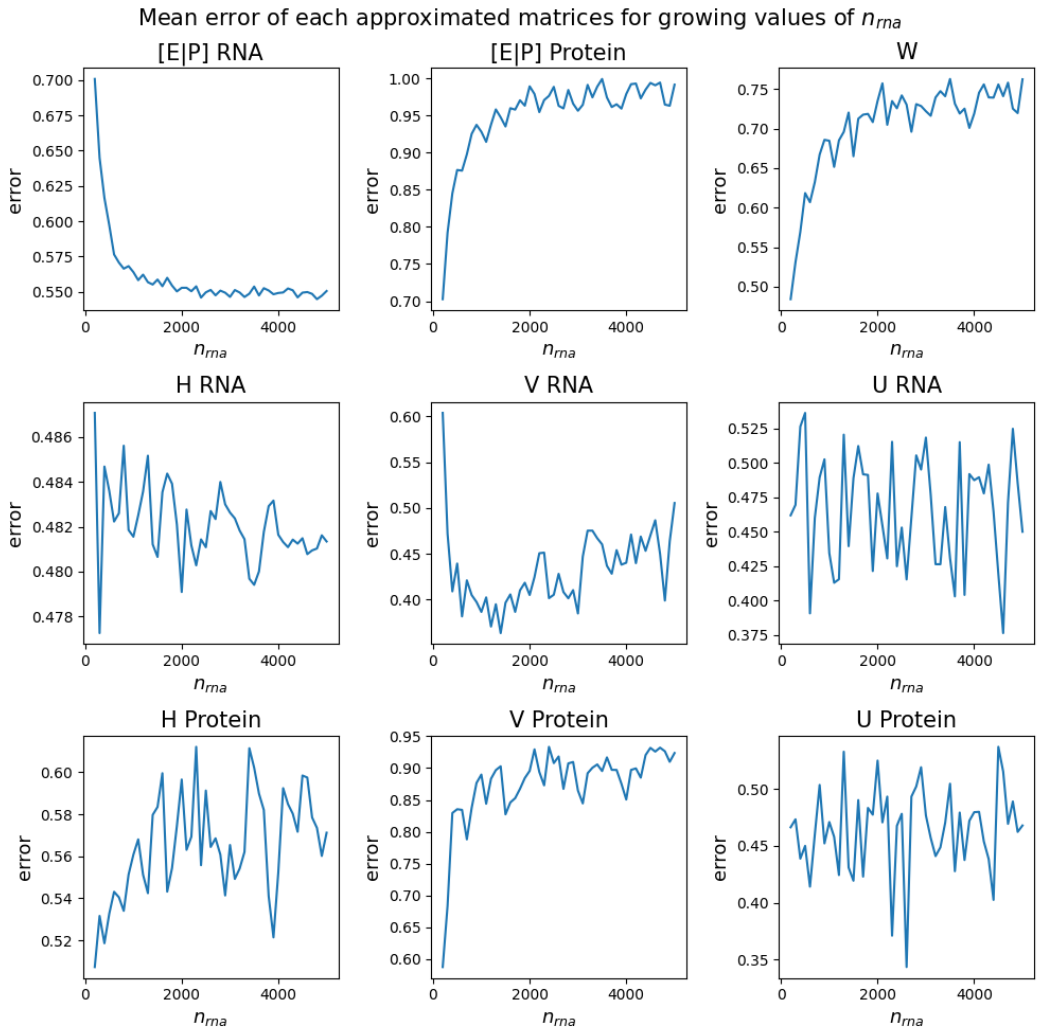


Figure 5.17: Error on each matrix of the UINMF factorization for different values of $n_{\text{rna}}$

From the mean error of matrices $[\mathbf{E}|\mathbf{P}]_{\text{rna}}$ and $[\mathbf{E}|\mathbf{P}]_{\text{protein}}$, we can see that increasing the number of samples for the modality scRNA-seq but with a fixed

number of sample for the protein abundance modality leads to a improvement in the approximation of $[\mathbf{E}|\mathbf{P}]_{\text{rna}}$ but it leads to a downgrade for the approximation of the protein modality. Its seem to infer some inequality during the process of optimization.

We can also observed on the sub-figure that show the mean error of the matrix $\mathbf{W}$ as $n_{\text{rna}}$ grows, the values of $\mathbf{W}$ does not improve. This result is intriguing, to understand it better we can see on Figure **??** the factorization of the lager value of $n_{\text{rna}}$ that was performed for Figure 5.19. We can see that the matrix $\mathbf{W}$ capture all the information, both the homogeneous and heterogeneous effect of the RNA modality.

To try to get an intuition about why this phenomenon happens and from where it could come from, let us dive back to the optimization problem defining UINMF, and more specifically to the sub-problem with respect to $\mathbf{W}$,

$$\mathbf{W}^t = \underset{\mathbf{W}^{t-1}\geq 0}{\operatorname{argmin}} \left\| \begin{pmatrix} \mathbf{H}_{\text{rna}}^T \\ \mathbf{H}_{\text{protein}}^T \end{pmatrix} \mathbf{W}^T - \begin{pmatrix} (\mathbf{E}_{\text{rna}} - \mathbf{V}_{\text{rna}}\mathbf{H}_{\text{rna}})^T \\ (\mathbf{E}_{\text{protein}} - \mathbf{V}_{\text{protein}}\mathbf{H}_{\text{protein}})^T \end{pmatrix} \right\|_F^2$$

We can have the intuition that is the dimension of $\mathbf{H}_{\text{rna}}^T$ and $\mathbf{E}_{\text{rna}}$ is much higher that the dimension of $\mathbf{H}_{\text{protein}}^T$ and $\mathbf{E}_{\text{protein}}$, the minimization could produce a $\mathbf{W}$ that focuses on minimizing the objective function only considering the RNA modality :

$$\left\| \mathbf{H}_{\text{rna}}^T \mathbf{W}^T - (\mathbf{E}_{\text{rna}} - \mathbf{V}_{\text{rna}}\mathbf{H}_{\text{rna}})^T \right\|_F^2$$

### 5.8.2 Unbalanced features

In this section we will analyse the effect of an unbalanced number of features for the scRNA-seq modality. All the other variables are fixed except the number of sample for the scRNA-seq modality $n_{\text{rna}}$ grows. Indeed the avoid having too many features compared to the number of sample and to keep a relevant problem, the sample size of the scRNA-seq modality also grows. Additionally to $n_{\text{rna}}$, only the number of unshared values $z_{\text{rna}}$ grows, the number of shared features $g$ remains fixed.

The model we will used here is slightly different from the framework model defined in Section 5.4. The unshared features matrix of the scRNA-seq modality will be changed. The pattern for $\mathbf{U}_{\text{rna}}$ will be a little bit more complex. We can see the initialization in Figure 5.18[2]:

[2]The initialization on Figure 5.18 represent the initialization the biggest value of $z_{\text{rna}}$ an $n_{\text{rna}}$, for the following experiment multiple dataset were factorized with the same pattern but with different sizes

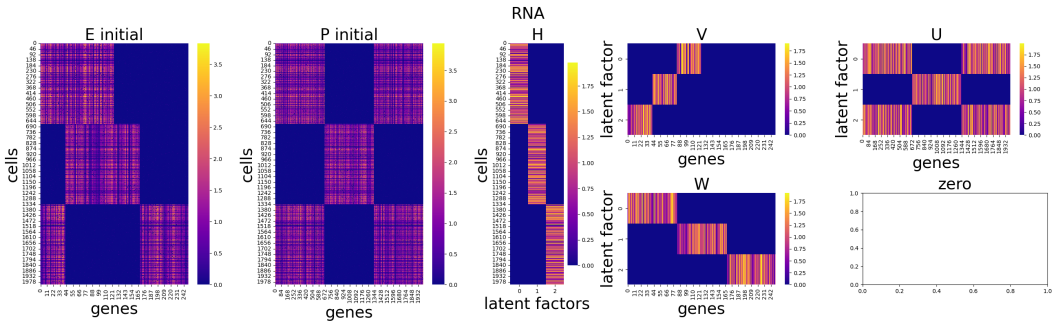


Figure 5.18: Initialization of the scRNA-seq data matrices for the framework model with a more complex pattern of unshared features

### Results

In Figure 5.19, we can see for each number of sample tested the mean error element-wise, calculated as in the Section 5.6, for each factor matrices and for the approximated matrices obtained with the algorithm.

We can see that the unbalanced sizes of the modalities seems to disfavor the minimization of the optimization of the protein abundance modality. The computations of the basic component for the homogeneous effects of shared features, the matrix $\mathbf{W}$, as well as the one for the heterogeneous effects of the protein abundance modality of shared features, the matrix $\mathbf{V}_{\text{protein}}$, suffer from this increase of unshared features.

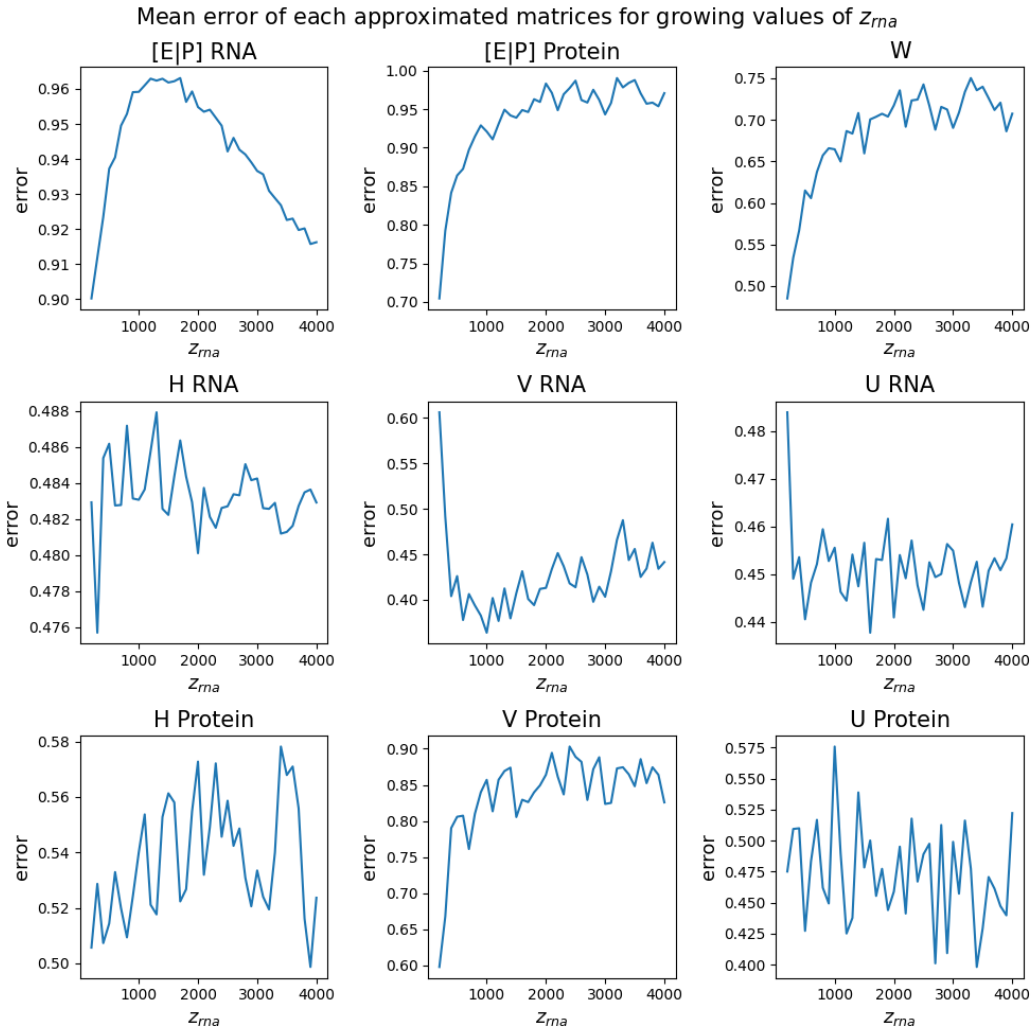


Figure 5.19: Error on each matrix of the UINMF factorization for different values of $n_{\text{rna}}$

However, as mentioned in [26], the sample size should grow exponentially with the number of features to avoid dimentionality problem. Here, as the number of features is very high it is not possible to have as much sample as required. In the modalization the number of sample grows linearly with the feature dimension.

# Chapter 6

# Discussion

The modelizations and tests performed in Chapter 5 showed some imperfections in the algorithm. Indeed the signal is not always captured by the matrices as we would expect. The algorithm finds an approximation of the matrix that satisfied the convergence criterium. The convergence criterium is that the error ceases to decrease.

The solution does not always represent both unshared and shared features. Indeed the algorithm sometimes miscalculates the matrices $\mathbf{H}$, leading to noise in the unshared feature matrices. Furthermore, the solution does not always represent the homogeneous, heterogeneous signal. Indeed the matrices $\mathbf{W}$ and $\mathbf{V}$ do not always represent distinctly the respectively homogeneous and heterogeneous signal.

This last result could have been expected as the model is derived from iNMF. This phenomenon was mentioned in [30]. They discussed the fact that the matrix $\mathbf{W}$ was expected to capture too much. Leading to an overestimated matrix $\mathbf{W}$ and underestimated matrices $\mathbf{V}$. Therefore the distinction between homogeneous and heterogeneous signals cannot directly be inferred from the factor matrices $\mathbf{W}$ and $\mathbf{V}$. This result seems to apply here. Indeed we can see that in most cases (not in those with extremely low regulation values), the matrix $\mathbf{W}$ captures more than the homogeneous effect while the matrices $\mathbf{V}$ usually do not capture enough information. Therefore the matrix $\mathbf{W}$ cannot directly be used to observe the homogeneous signal. Combining all of the experiments done on the datasets we used, all the pertinent information can emerge. However, the algorithm does not provide always a clear and correct clustering of the data, as all the clusters we defined were evident clusters which is a big concern. Using the tuning parameters to explore the data-set can bring some idea of the heterogeneous and homogeneous signal but the algorithm often fails to distinguish heterogeneous and homogeneous signals in $\mathbf{V}$ and $\mathbf{W}$.

The regularization was only tested for the penalization of the heterogeneous effects of the shared feature and the unshared features jointly. Indeed the penalty term defined in Charpter 4 is a function of the matrices $\mathbf{H}$ and $(\mathbf{VU})$ :

$$\sum_{i}^{d} \lambda_i \sum_{i}^{d} \left\| \mathbf{H}^i \left( \mathbf{V}^i \mathbf{U}^i \right) \right\|_F^2$$

The regularization could be defined differently by defining independently the penalization on the heterogeneous effects of the shared feature, $\mathbf{V}$ and the unshared features $\mathbf{U}$. The different sizes of the matrices could also be penalized, and the imbalance of modalities could be regulated with an additional penalty term depending on the matrices $\mathbf{U}$

As discussed in [28], optimizing an objective function is not obviously and necessarily equivalent to identifying the underlying actual components of the data sets. The factor matrices come with high dimensionality and finding a good decomposition of the input matrix might not necessarily be the decomposition of heterogeneous and homogeneous signal we hope for. To obtain a better solution one might start by finding the most optimal regularization term and adding constraints from additional prior knowledge to move the solution toward the global minimum we expect.

As said by Libbrecht and Noble in [19] *"There is no optimal machine learning algorithm that works best for all problems"* . The selection of the method must be carefully chosen accordingly to the problem and the prior knowledge about the problem.

Another limitation of the method UINMF implmented in *LIGER* to take into account has been pointed out by [2] and [8]. The UINMF method as implemented in *LIGER* can only integrate two modalities and that the subset of shared feature cannot be empty (i.e. $\mathbf{E}_1$ and $\mathbf{E}_2$ must be non-empty).

# Chapter 7

# Conclusion

To conclude, we can say that the algorithm could be used to analyse real biological data. However, one must be careful when it comes to the analysis of the result. Indeed the clustering is not always reliable and the separation of the heterogeneous and homogeneous signals is not always distinct.

Although the results were not satisfying some aspects have to be taken into account to improve the integration and the quality of the analysis.

- **The initialization :** the initialization defined by *LIGER* might not be the most efficient for all the problems. Indeed the initialization can be crucial to converge to a relevant solution. A good initialization is specific to the problem. The initialization can be also a way to incorporate prior knowledge or, in the case where factor matrices are initialized with samples as in *LIGER*, we could choose directly the sample to favor some relevant samples or samples that were identified as a framework. Moreover performing multiple runs with multiple seeds or multiple different initializations is highly recommended.

- **The regularization :** the regularization can also be chosen specifically for the problem that must be tackled. Performing UINMF with *LIGER*, the regularization term can be adapted by choosing the best value as in Section 5.7, by performing multiple runs with different values and analyse the quality of the different solutions. However the regularization can be even more specific to the problem, some additional terms can be incorporated into the problem from the prior knowledge determined by the problem.

- **The formulation :** the statement of the problem determines its solution, and adding some prior knowledge to the objective function could refine the solution toward a more desired solution. It can be done by adding some regularization that is adapted to the problem studied.

- **The uniqueness :** the uniqueness of the solution is an unresolved issue, reducing the set of possible solutions is an important step toward uniqueness. Adding regularization on the smoothness of the sparsity with penalization terms can reduce the set of possible solutions and had been proven to be a good solution
- **The unbalanced modalities :** avoiding having unbalanced modalities is important, the number of samples and the number of features in the same range of values gives a more reliable solution. To do so the most straightforward option is the selection of a random or properly chosen subset of the data.
- **The dimension of features and samples :** having a too high dimension is important in machine learning. The analysis has been proven to be more difficult with a high number of features. A model trained on too few data points with respect to the dimension of the features gives a solution too specific to the training data and that might not be relevant. Choosing a subset of the features can be a solution to the problem.

# Bibliography


[1] Ricard Argelaguet, Anna SE Cuomo, Oliver Stegle, and John C Marioni. Computational principles and challenges in single-cell data integration. *Nature Biotechnology*, pages 1–14, 2021.

[2] Mingbo Cheng, Zhijian Li, and Ivan Gesteira Costa Filho. Mojitoo: a fast and universal method for integration of multimodal single cell data. *bioRxiv*, 2022.

[3] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Non-negative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[4] Michael Eisenstein. The secret life of cells. *Nat Methods*, 17:7–10, 2020.

[5] Issam El Naqa and Martin J Murphy. What is machine learning? In *machine learning in radiation oncology*, pages 3–11. Springer, 2015.

[6] Jean Fan, Kamil Slowikowski, and Fan Zhang. Single-cell transcriptomics in cancer: computational challenges and opportunities. *Experimental & Molecular Medicine*, 52(9):1452–1465, 2020.

[7] Renaud Gaujoux. An introduction to nmf package. *Version 020*, 6, 2014.

[8] Shila Ghazanfar, Carolina Guibentif, and John C Marioni. Stabmap: Mosaic single cell data integration using non-overlapping features. *bioRxiv*, 2022.

[9] Nicolas Gillis and François Glineur. Nonnegative factorization and the maximum edge biclique problem. *arXiv preprint arXiv:0810.4225*, 2008.

[10] Hyunsoo Kim and Haesun Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares. Technical report, Georgia Institute of Technology, 2006.

[11] Hyunsoo Kim and Haesun Park. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM journal on matrix analysis and applications*, 30(2):713–730, 2008.

[12] Jingu Kim, Yunlong He, and Haesun Park. Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework. *Journal of Global Optimization*, 58(2):285–319, 2014.

[13] April R Kriebel and Joshua D Welch. Nonnegative matrix factorization integrates single-cell multi-omic datasets with partially overlapping features. *bioRxiv*, 2021.

[14] Hans Laurberg, Mads Græsbøll Christensen, Mark D Plumbley, Lars Kai Hansen, and Søren Holdt Jensen. Theorems on positive data: On the uniqueness of nmf. *Computational intelligence and neuroscience*, 2008, 2008.

[15] Daniel Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2000.

[16] Tao Li and Chris Ding. The relationships among various nonnegative matrix factorization methods for clustering. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 362–371. IEEE, 2006.

[17] Yifeng Li, Fang-Xiang Wu, and Alioune Ngom. A review on machine learning principles for multi-view biological data integration. *Briefings in bioinformatics*, 19(2):325–340, 2018.

[18] Yunjin Li, Lu Ma, Duojiao Wu, and Geng Chen. Advances in bulk and single-cell multi-omics approaches for systems biology and precision medicine. *Briefings in Bioinformatics*, 22(5):bbab024, 2021.

[19] Maxwell W Libbrecht and William Stafford Noble. Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 16(6):321–332, 2015.

[20] Bilal Mirza, Wei Wang, Jie Wang, Howard Choi, Neo Christopher Chung, and Peipei Ping. Machine learning and integrative analysis of biomedical big data. *Genes*, 10(2):87, 2019.

[21] Sylvia Richardson, George C Tseng, and Wei Sun. Statistical methods in integrative genomics. *Annual review of statistics and its application*, 3:181, 2016.

[22] Erwin M Schoof, Benjamin Furtwängler, Nil Üresin, Nicolas Rapin, Simonas Savickas, Coline Gentil, Eric Lechman, John E Dick, Bo T Porse, et al. Quantitative single-cell proteomics as a tool to characterize cellular hierarchies. *Nature communications*, 12(1):1–15, 2021.

[23] Indhupriya Subramanian, Srikant Verma, Shiva Kumar, Abhay Jere, and Krishanpal Anamika. Multi-omics data integration, interpretation, and its application. *Bioinformatics and biology insights*, 14:1177932219899051, 2020.

[24] Leo Taslaman and Björn Nilsson. A framework for regularized non-negative matrix factorization, with application to the analysis of gene expression data. *PloS one*, 7(11):e46331, 2012.

[25] Andy Tran, Pengyi Yang, Jean YH Yang, and John Ormerod. Computational approaches for direct cell reprogramming: from the bulk omics era to the single cell era. *Briefings in Functional Genomics*, 2022.

[26] Michel Verleysen and Damien François. The curse of dimensionality in data mining and time series prediction. In *International work-conference on artificial neural networks*, pages 758–770. Springer, 2005.

[27] Lipo Wang, Yaoli Wang, and Qing Chang. Feature selection methods for big data bioinformatics: a survey from the search perspective. *Methods*, 111:21–31, 2016.

[28] Yu-Xiong Wang and Yu-Jin Zhang. Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on knowledge and data engineering*, 25(6):1336–1353, 2012.

[29] Joshua D Welch, Velina Kozareva, Ashley Ferreira, Charles Vanderburg, Carly Martin, and Evan Z Macosko. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell*, 177(7):1873–1887, 2019.

[30] Zi Yang and George Michailidis. A non-negative matrix factorization method for detecting modules in heterogeneous omics multi-modal data. *Bioinformatics*, 32(1):1–8, 2016.

[31] Xue Ying. An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, page 022022. IOP Publishing, 2019.

[32] Ziqi Zhang, Haoran Sun, Xinyu Chen, Ragunathan Mariappan, Xi Chen, Mika Jain, Mirjana Efremova, Vaibhav Rajan, Sarah Teichmann, and Xiuwei Zhang. scmomat: Mosaic integration of single cell multi-omics matrices using matrix trifactorization. *bioRxiv*, 2022.

**UNIVERSITÉ CATHOLIQUE DE LOUVAIN**
**École polytechnique de Louvain**
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**