Clustering of high-dimensional genomic data

PHAN Duc Thanh[1]
supervised by
Christophe RIGOTTI[1], Marion LELEU[2], Jacques ROUGEMONT[2]

[1] BEAGLE (LIRIS-EPC INRIA)
[2] Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

Résumé Ce travail effectué dans le cadre d'un projet en collaboration entre l'équipe Beagle et l'équipe BBCF de l'EPFL qui visait à développer une méthode de clustering sur les données génomiques de très forte dimensionnalité fournies par la plateforme HTSstation. Pour cela, nous avons étudié différentes méthodes et décidé de travailler avec les techniques de subspace clustering. Nous avons proposé et mis en place ensuite un framework permettant de comparer et évaluer les résultats fournis par différents algorithmes de cette approche. Nous avons aussi proposé et implémenté différentes mesures pour l'évaluation de l'intérêt des clusterings obtenus. Sur les données réelles de référence, notre approche fournit des clusterings en accord avec des groupements connus des experts.

Keywords: Fouille de données de forte dimensionnalité, subspace clustering, analyse de données génomiques

Abstract In the context of a joint research project between the Beagle team and the BBCF team from the EPFL to study techniques of cluster analysis of high-dimensional genomic data provided by the HTSstation platform, we searched for a method that can find meaningful clusters from these data to be presented to biologists. In order to do so, we studied a number of possibilities and decided to work with subspace clustering techniques. We proposed and implemented a framework allowing to compare and evaluate clusterings found by different subspace clustering algorithms of different approaches. We also designed measures based on spatial coherence to rank clusterings. On real data, our method leads to results similar to groups found by experts.

Keywords: high-dimensional data-mining, subspace clustering, analysis of genomic data

1 Introduction

This document describes the outcome of my research project that has been carried out so far at the Beagle Team, INRIA Rhones-Alpes Laboratory, under

the supervision of Dr. Christophe Rigotti from LIRIS Laboratory and in close collaboration with Dr. Marion Leleu and Dr. Jacques Rougemont from the École Polytechnique Fédérale de Lausanne (EPFL), Switzerland.

1.1 Context of the Project

Dr. Leleu and Dr. Rougemont are affiliated with The Bioinformatics and Biostatistics Core Facility at the EPFL School of Life Sciences whose main missions include developing novel methods regarding data management and statistical data analysis in genomics and genetics, notably complex high-density DNA array and high-throughput sequencing data.

In this thesis, we mostly work on subspace-clustering, a paradigm of cluster analysis which deals exclusively with high-dimensional data, and on its application on genomic data. The proposed method has been applied on datasets retrieved from Sexton et al. [37].

1.2 Structure of the document

The next section will present the biological aspects of the project. Once the reader has gained a clear understanding of the motivations underlying this project, a dedicated section will develop some background information about subspace clustering and methods of clustering evaluation and validation. It will then give a brief overview of the contribution of this work and show in details how the experiments have been conducted, along with the obtained results. To conclude, we will present the results and give some in-depth discussions before concluding with potential future research.

2 Motivation

In their paper in Cell [37], Sexton et al. study genomic data of *Drosophila*[3], in particular *Hi-C* and epigenetics data (*Chip-Seq* and *ChIP-chip* mostly). On one hand, *Hi-C* is an advanced technique used to identify genomic locations that are spatially close to each other, in other terms, it looks for long-range DNA-interactions. On the other hand, *Chip-seq* and *ChIP-chip* give information about interactions between proteins and DNA locations. Based on those data, the resulting analysis identified a set of clusters in the Drosophila's genome that were significant and biologically meaningful.

As shows Figure 2, considering a certain chromosome (a part of the genome), in this clustering, an object is simply a portion of this chromosome (i.e. a sequence of the so-called base pairs). Each object is described by a multidimensional vector of feature values where each feature corresponds to an epigenetic profile along the chromosome. The feature value of the object here is simply the average of the values of this profile over the portion of chromosome associated.

The methods applied in their analysis are quite complex but could be summarized as follows :

3. Scientific name for a small kind of fly

- Based on previous studies that suggested a link between the contact map's physical domains and epigenetic data [11, 21], Sexton et al. analyzed these data using epigenetic profiles thanks to clustering techniques.
- First, a statistical model called "scaling" was proposed to systematically identify contiguous genomic physical domains. A total of 1169 such physical domains have been identified among the genome by using this technique. Each physical domain is then used as an object (i.e. portion of the genome) to perform cluster analysis.
- Next, they selected 315 most correlated profiles, with regard to physical domains, among 403 available linear epigenetic profiles based on chi-squared test score.
- They then performed a *hierarchical clustering* on the objects using these selected 315 profiles which resulted in finding apparently four major domain classes. Based on this results, they proceeded to select 4 profiles (based on their biological characteristics) among these 315 and then perform a clustering with *k-means* using these 4 profiles remaining with $k = 4$. Four clusters found are called "4 classes" and labeled as "Active", "Null", "HP1" and "HP1/Centromere". (see Figure 1)
- Thanks to this annotation, they continue to study their characteristics and association with epigenetic marks to obtain several further interesting results.

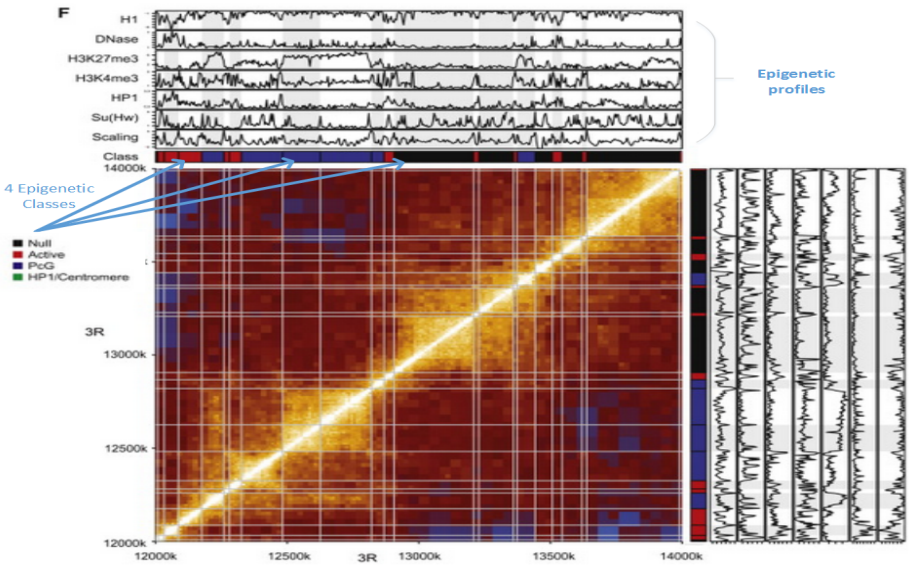


FIGURE 1: Contact map for a region of chromosome 3R, showed alongside selected epigenetic profiles and olor-coded indication of physical domains and their 4 classes found by the cluster analysis step. Figure taken from [37]

This whole study is tightly linked to the initial division of the genome into physical domains obtained from Hi-C data. However, such data are not always available and we can wonder whether such clusters could be discovered considering only epigenetic data.

In our project, we are hence interested in working on methods of large-scale clustering of high-dimensional epigenetic data. On the other hand, and in contrast to Sexton et al.'s work, we would like to be able to select not only clusters but also relevant features that help "explain" them in a more systematic manner. We are also interested in techniques of evaluation and validation, in order to facilitate the selection of the most meaningful results to be presented to biologists.

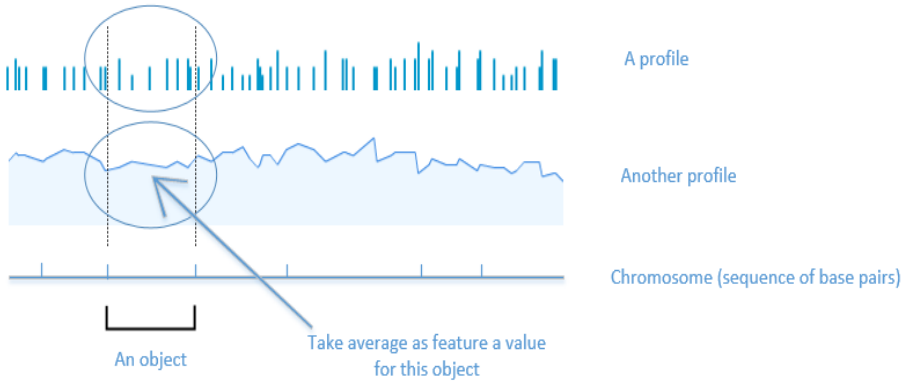


FIGURE 2: Epigenetic profile data

3 High-dimensional cluster analysis

Knowledge discovery in databases (KDD) is an interdisciplinary subfield of computer science that aims to help identifying valid and meaningful patterns from databases. The KDD process consists usually of the following steps [10, 42] :

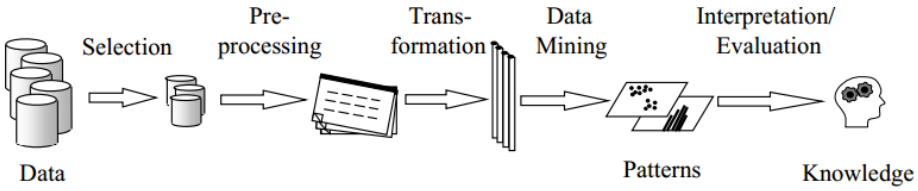


FIGURE 3: Typical KDD Process. Figure taken from [42]

The core of such process is the Data Mining step of which data clustering (also called cluster analysis) is a traditional technique for automatic grouping of objects into subsets such that objects in a same group (called cluster) are similar to each other (according to some distance measure) whereas objects in different clusters are dissimilar, for the purposes of summarization or improved understanding [19, 34].

While clustering has a long history and a very large number of techniques have been developed which have enjoyed a tremendous amount success across many fields, many challenges still remain. Recent advances in technology have greatly enhanced the generation and acquisition, as well as the processing, storage and transmission of data, resulting in the availability of a tremendous amount of massive datasets with many objects and dimensions. These high-dimensional datasets are known to pose many problems to traditional clustering techniques, especially those accounting for the full data dimension spaces [4], due to effects attributed to the so-called *curse of dimensionality*. It is essential to have better techniques for identifying patterns in these kinds of data which could be found in many fields such as molecular biology, web mining, text mining, image processing, among others.

The expression "curse of dimensionality" coined by Richard E. Bellman refers to a set of various problems that arise when dealing with data in high-dimensional spaces [8]. According to [22], when it comes to cluster analysis, there are four major problems that need addressing :

- High dimensionality makes it difficult to rationalize and almost impossible to visualize. Furthermore, with an increasing number of dimensions, many traditional classical algorithms will be broken as systematic searching through high-dimensional spaces will become untraceable, among other reasons [8].
- Deterioration of expressiveness of the concept of distance : given any data point, as the number of dimensionality increases, the distance to its nearest neighbor and the distance to its farthest neighbor converge. In other words, the discrimination in distance between data points becomes less and less meaningful.
- Local feature relevance problem : given a large number of dimensions, it is expected to have many that are irreverent which might interfere with and obfuscate the results of the task of pattern finding. A method commonly-used to overcome this problem is to perform a dimension selection and/or reduction before the mining task. On the other hand, the relevance of certain dimensions may differ for different groups of objects within the same dataset, in other words different clusters might be found in different subspaces. Therefore, global feature selection or dimensionality reduction techniques (PCA for example) would not be sufficient.
- Given a large number of attributes, it is likely that some attributes are correlated. Hence, clusters might exist in arbitrarily oriented linear or affine subspaces.

4 Subspace clustering

These aforementioned challenges have led to the development of new specialized techniques one of which called subspace clustering [4]. Subspace clustering refers to a group of clustering methods in subspaces of potentially high-dimensional data. In contrast to traditional clustering techniques, subspace clustering goes beyond full space clustering by searching for *subspace clusters* that exist in multiple, possibly overlapping subspaces. In principles, algorithms of this area must deal with a two-fold problem : simultaneous identification of clusters and subspaces in which they exhibit.

Definition 1. ***Subspace cluster :*** *let $\mathbb{D} = \mathbb{O} \times \mathbb{A}$ be a dataset presented in the form of a matrix where $\mathbb{O}, \mathbb{A}$ are sets of clustered objects and relevant dimensions (or features, variables as interchangbly refered in litterature) respectively, then a subspace cluster is defined as a sub matrix $C = O \times A$ where $0 \subseteq \mathbb{O}, A \subseteq \mathbb{A}$ such that 0 is homogenous in A [38, 22].*

We assume also every dimension has the same domain : $dom(\mathbb{O}) = [l, u] \subseteq [-\infty ... + \infty]$

Definition 2. ***Clustering result :*** *A* clustering result *(or clustering for short) is a set of subspace clusters $\{C_1, C_2, .., C_i\}$ which are produced by an execution of a subspace clustering algorithm.*

In subspace clustering, different clusters may have different subspaces and overlapping of cluster is allowed (i.e. an object can be part of different cluster in different subspace). Another term that has been equally used in *subspace clustering* literature is *projected clustering* but algorithms belonging with the latter do not allow overlapping. More details will be presented in the next subsection. Note also that a run of a *projected clustering* algorithm gives only a clustering while a run of a *subspace clustering* counterpart gives us a set of clusterings, each of which is a projection of the whole set of clusters on a subspace.

4.1 Taxonomy of subspace clustering algorithms

Since there are an infinite number of arbitrarily oriented subspaces, it is computationally impossible to test all possible subspaces. Certain heuristics and assumptions must therefore be made in order to conquer this infinite search space. The following taxonomy is based on [22] which categorizes algorithms in term of their subspace strategy.

4.1.1 Axis-parallel subspaces Many subspace algorithms make assumptions that clusters exist only in axis-parallel subspaces, which shrink down the infinite search space to "only" exponential. The reason for being addressed as

4. Note that we present only with techniques from the data mining community, for a machine learning and computer vision point of view, please refer to [39]

axis-parellel is due to an geometrical intuition : we can imagine along irrelevant dimensions, the clusters points tend to be uniformly distributed so they would form a hyper plane that are parallel to these axes. There are many ways to classified algorithms in this approach, one of which is based on the underlying cluster definition and the parameterization of the clustering task [29] :

4.1.1.1 Cell-based Algorithms in this category search for dense sets of objects which contains more than a certain threshold of objects. Typically, they first partition the dataset into equal-sized units of width ξ and define a cluster as a set of (dense) cells each of which has at least τ (fixed or adaptive) elements. A subspace is then a restriction of this set in a subset of dimensions. Formally, a cell-based cluster $C = (O, S)$ is specified by a set of intervals I where $I_i, = [l_i, u_i] \subseteq [l, u] \forall i \in S$ and $I_j = [l, u] \forall j \notin S$ i.e the cell is restricted only in dimensions in S and $|O| \geq \tau$.

The most famous algorithm in this approach is the pioneering CLIQUE [2] which defines a cluster as a maximal set of connected dense grid cells. Since grid cell density satisfies the downward closure property (if a grid cell is dense in S it must all so be dense in $\forall S' \subseteq S$), *APRIORI-like* pruning style technique can be applied. It also utilizes another heuristic based on the minimum description length (MDL) principle to discard subspace candidates with low coverage (the fraction of the dataset covered by the dense units in the subspace).

ENCLUS [7] is another technique that also searches for fixed-size grids but it does not use directly the notions of coverage and density but a measure of entropy instead. It is based on the observation that a subspace with clusters typically has a lower entropy score than those without clusters. On the other hand, it defines clusterability of a subspace in term of coverage, density and correlation (i.e. dimensions of that subspace should be correlated) all of which could be measured by entropy. The pruning step is accomplished using the downward closure property of entropy and the upward closure property of subspace correlation (minimally correlated subspace are supposed to be most interesting). The upward closure pruning style is based on [6].

Other variants of CLIQUE include MAFIA [13], MINECLUS [40] and SCHISM [36]. MAFIA enhances the performance of CLIQUE by exploiting the use of histogram in order to fix the number of grids to be created for each dimension, instead of a uniform grid size for every dimension. MINECLUS take advantages of using FP-trees to achieve better runtimes. SCHISM addresses the problem of using a global threshold by adopting the support and Chernoff-Hoeffding bounds as the (variable) density thresholds and searches for maximal subspaces in a depth-first search manner.

4.1.1.2 Density-based Algorithms in this paradigm define cluster based on the model of DB-SCAN [9] as a dense region separated by sparse one. They compute the density of a given object by counting the number of other objects within its $\varepsilon-$neighborhood ($N_\varepsilon(p) = \{q \in \mathbb{O} | dist(p, q) \leq \varepsilon\}$ for some distance *dist*). It proceeds to define a binary relation called *density-connected* between points. A cluster is then defined as a set of objects that are *density-connected* to each

other. It is shown that these density-connected sets satisfy the downward closure property which allows the use of APRIORI-like style in pruning subspaces. Algorithms in this category are able to find clusters of arbitrary shape and are robust to noise. [28].

The most prominent algorithm in this approach is SUBCLU [20] an extension of DBSCAN to subspace clustering which computes all clusters for each subspace using DBSCAN. SUBCLU, like others in this approach, does not require discretization step but needs expensive database scans to compute $\varepsilon-$neighborhood for each point.

Other similar algorithms include FIRE [23] and INSCY [3] which are based on the same idea but using different techniques to improve computational performance.

4.1.1.3 Clustering-oriented Clustering-oriented algorithms tend to search for clusters that optimize a certain objective function for the whole clustering result set. This function is constructed based on the desired properties for the results to satisfy : the number of clusters, overlapping condition, statistics-oriented properties...

One of the earliest algorithms in this class is PROCLUS [1] which, unlike most of the algorithms that have been described, does not allow cluster overlapping. PROCLUS can be viewed as an extended version of $k-mean$ for subspace clustering since it partitions the data into k disjoint clusters with average dimensionality of l and then refines them by a hill climbing approach. ORCLUS [33] is similar to PROCLUS but uses random sampling to improve the processing time.

Another related algorithm named $SC-Kmeans$ [30] which also an extension of $k-means$ which is based on dimension sampling to choose subspace in which it project the objects in order to perform $k-means$.

4.1.2 Correlation-clustering Correlation clustering aims at finding clusters in arbitrarily oriented subspaces. Such clusters appear as hyper planes of arbitrary dimensionality in the data space which exhibit common correlations between different linear or affine subsets of features. As the problem is two-fold, many algorithms for correlation clustering work by combining the concepts of clustering and correlation detection. Therefore, *ORCLUS* can be viewed as a correlation clustering algorithm which integrates *PCA* into *k-means* clustering while 4C [5] integrates *PCA* into a density-based clustering algorithm.

4.1.3 Bi-clustering In contrast to axis-parallel subspace clustering, bi-clustering [26] (also called co-clustering or two-mode clustering) treats dimensions and objects equally in a symmetric way, thus allowing simultaneous clustering of the rows and columns of a matrix. The clustering task consists of searching for submatrices (biclusters) that exhibit similar behavior across a subset of columns, and vice versa. This technique is widely used in bio-informatics for analyzing micro-array database. Other methods of taxonomy have been used in [31, 22] include but not limited to :

- Projected clustering vs Subspace Clustering : *Projected clustering* aims at identifying the locally relevant reduction of attributes for each object. Specifically, each object is assigned to exactly one cluster (or noise) and a corresponding projection. It results in disjoint clustering results. *Subspace clustering*, on the other hand, allows identifying several possible subspaces for any object. Thus, a point can belong to multiple clusters.
- Top-down vs bottom-up : The rational of *top-down* approaches are to determine the subspace of a cluster starting from the full-dimensional space which has to rely on the locality assumption. This condition assumes that the subspace of a cluster can be derived each cluster can be learned from its local neighborhood cluster members. The *bottom-up* approaches, on the other hand, are based on the observation that searching for subspace cluster pair of objects-dimensions is similar to searching for frequent item sets in market basket analysis in the area of transaction databases. The idea is then to use APRIORI-like pruning methods for discarding non-clusterable dimensions.

4.2 Subspace clustering comparison and evaluation

Regardless of techniques, the final steps of any KDD process must include the validation and interpretation of the results. Since the clustering results are not known a priori and bad clusters may arise in many common situations, the step of evaluating results (using criteria and measures) is required . Furthermore, in many cases the ground truths are not known either, a consultation with domain experts need to be carried out in order to validate result quality for drawing conclusion. [18]

The cluster validation analysis tasks include internal and external validation[32]. The former refers to the process of comparing the found clustering results with a set of true clustering if available. The latter refers to the process of measuring the quality when the ground truth is not available. There are several ways to perform such test, for instance techniques that measure the stability of the results by sampling the input data. [24]

4.2.1 Subspace clustering quality criteria There are many aspects to consider when it comes to evaluating clustering results : runtime performance, scalability, clustering result quality and so on. The most important and most difficult is determining clustering quality mainly because usually there is no ground truth to with we can compare the clustering result. Even in case it is available, for subspace clustering we have to take into account not only cluster objects but also the relevancy of dimensions associated.

A general model for external subspace clustering evaluation is given in [16] :

- All and only : a good subspace clustering algorithm must be able to find *only* meaningful clusters yet have to find *all* such clusters. Consequently, we have to solve a two-fold problem : coverage of hidden clusters, coverage of cluster objects and the purity of cluster. In other words, for each hidden

cluster, all of it objects should be found and should be found in a cluster (every split is punished).
- Subspace awareness : for each cluster it has to find a set of all relevant dimensions. This aspect is different from traditional clustering.
- Redundancy awareness : a good measure should be able to determine and punish redundancies of a clustering. For example a clustering may report multiple times a cluster in different subspace projection.

4.2.2 Subspace clustering evaluation measures In the literature, there is no lack of evaluation measures for traditional clustering, varying from the classic such as Rand-index, entropy-score, f1-measure or accuracy to the more recent like information variation [27] among others. However, techniques for evaluating subspace clustering are not yet mature. In [29], a comprehensive survey on subspace clustering comparison was conducted where some measures that seem appropriate for evaluating subspace clustering were listed. If we denote hidden "clusters" (or classes) as $H_i = (O_{H_i}, S_{H_i}), i = \overline{1,m}$ in contrast to found clusters $C_j = (O_j, S_j), j = \overline{1,k}$, then Entropy and F-score can be defined as follows :

- Entropy [41] : an adapted version of the notion of entropy measure for evaluating the purity of a clustering. A found cluster should mainly contain objects from one hidden cluster and any merging (splitting) of several hidden clusters to one (different) found cluster (clusters) should be punished. Formally, let $C_j = (O_j, S_j)$ a cluster, its Entropy is defined by : $E(C_j) = -\sum_{i=1}^{m} p(H_i|C_j).log(p(H_i|C_j))$ where $p(H_i|C_j) = \frac{|O_{H_i} \cap O_{C_j}|}{|O_{C_j}|}$. The overall quality of the clustering w.r.t this criterion is obtained as the average over all clusters $C_i \in R$ weighted by the number of objects per cluster, normalized by dividing the maximal entropy (which equals to $log(m)$) for m hidden clusters : $\frac{\sum_{i=1}^{m} |C_j|.E(C_j)}{log(m).\sum_{i=1}^{m} |C_j|}$
- F-score [12] : inspired by *F-Measure*, this measure evaluates simultaneously the purity and the coverage of clusters found. It involves the concepts of *Recall* and *Precision* as well as *mapping*. Suppose cluster C_i is *mapped* to hidden cluster H_j then Recall is the portion of items from H_j that are present in C_i, thus measuring how much C_i covers w.r.t H_j. Similarly, Precision is calculated as the portion of C_j that are present in H_i, thus measuring how pure C_i is w.r.t H_j. Formally, given H a hidden cluster, $mapped(H)$ [5] the set of all found clusters that are mapped to H, then $recall(H) = \frac{|O_H \cap O_{m(H)}|}{|O_H|}$ and $precision(H) = \frac{|O_H \cap O_{m(H)}|}{|O_{m(H)}|}$ where $O_H, O_{m(H)}$ are the sets of objects of H and all objects from all cluster in $mapped(H)$, respectively. The F-score is then given by $\frac{1}{m} \sum_{j=1}^{m} \frac{2.recall(H_j).precision(H_j)}{recall(H_j)+precision(H_j)}$.

Both measures that have been mentioned cannot distinguish two clustering with the same cluster but projected into two different subspaces, thus unable to take into account the relevance of dimensions. There have been efforts to develop

5. The explicit definition of this mapping process is omitted, interested readers are invited to see [29]

measures that can take into account also dimensions as in [32] which resulted in *CE* and *RNIA*. Let $C = (O, S), O = (o_1, ..., o_n), S = (s_1, .., s_k)$ a cluster of n objects, k dimensions, the idea is to represent it as $C_{O,S} = \begin{pmatrix} o_{1,s_1} & o_{1,s_2} & \cdots & o_{1,s_k} \\ o_{2,s_1} & o_{2,s_2} & \cdots & o_{2,s_k} \\ \vdots & \vdots & \ddots & \vdots \\ o_{n,s_k} & o_{n,s_2} & \cdots & o_{n,s_k} \end{pmatrix}$

Denote those o_{i,S_j} as micro-objects, we have a matrix $n \times k$ of micro-objects. Based on this new representation, CE and RNIA compare micro-objects of hidden clusters to found clusters, thus taking into account the dimensions.

- RNIA : as stated above, we can consider 2 clusterings as 2 sets of matrices whose elements are micro-objects. Let denote $m(H), m(C)$ as the unions of all micro-objects of the hidden cluster and the found clusters, respectively. RNIA computes actually the *Jaccard distance* between these $m(H), m(C)$: $RNIA = \frac{|U|-|I|}{|U|}$ where $U = m(H) \cup m(C), I = m(H) \cap m(C)$. According to [25], *Jaccard distance* is a proper metric, therefore RNIA is also a metric.
- CE : stands for Clustering Error which extends RNIA. One problem for the RNIA measure is that one cannot distinguish if several found clusters cover a hidden cluster or only exactly one found cluster matches this hidden cluster. To address this, the basic idea is to find a $1:1$ mapping between the hidden and found clusters. In [27], the authors used the *Hungarian method* of assignment (which optimizes the trace of the confusion matrix) to implement this mapping. For each (C, H) pair of mapped hidden cluster and found cluster, denote $ci(C, H)$ as the cardinality of the intersection of their sets of micro-objects. Then CE is defined as $CE = \frac{U - I_{max}}{U}$ replacing in RNIA I by $I_{max} = \sum ci(C, H)$ summing over all mapped pairs.

As many other external validation techniques, both *CE* and *F-Score* rely on a *matching* step that assigns a cluster found by the algorithm to a hidden one. The issue with this is that it evaluates not only the goodness of clustering but also the quality of this cluster assignment. Furthermore, it only accounts those clusters that are matched to a given target hidden clusters. This so-called matching problem has been well addressed in [27]. There is a quite elegant solution to this problem by using a technique called V-Measure [35] which, similar to *F-Score*, computes at the same time the purity and coverage but does not require this matching step to achieve.

5 Framework proposal

In this project, we investigate the prospect of systematically applying multiple subspace clustering algorithms on large-scale heterogeneous genomic data. In particular, we propose a framework focusing on post-processing that can facilitate the automation of this process and explain how to use it to group and compare results found by different algorithms with different parameterization. The problem of *meaningfulness* as well as *redundancies* of result were also addressed. Besides, we propose some measures that might be useful for the evaluation task. Finally, we investigate the results obtained by subspace clustering

and compare them to those from [37] and discuss the relationship between these different evaluation measures.

As the field of subspace clustering is rather young, there are not yet many open source implementations of subspace algorithms available to the community in order to use, study or extend. There have been some effort to address this problem, for instance the project *OpenSubspace* by Muller et al. [30]. Originally implemented as plugins for the Weka platform, this project has now been ported to Knime [15], another widely-used data mining platform. It aims to provide implementations of subspace clustering algorithms, as well as some other comparing and visualization techniques.

The original version that was implemented in Weka lacks the capacity to output the obtained clustering results for further analysis. This is deemed not to be very useful for our work which involves heavy post-processing of the results ; therefore we chose to work on the Knime-based version. We proposed and implemented on top of the Knime-based *OpenSubspace* a specialized framework which fits better our needs, in particular for post-processing tasks and which also provides some alternative.

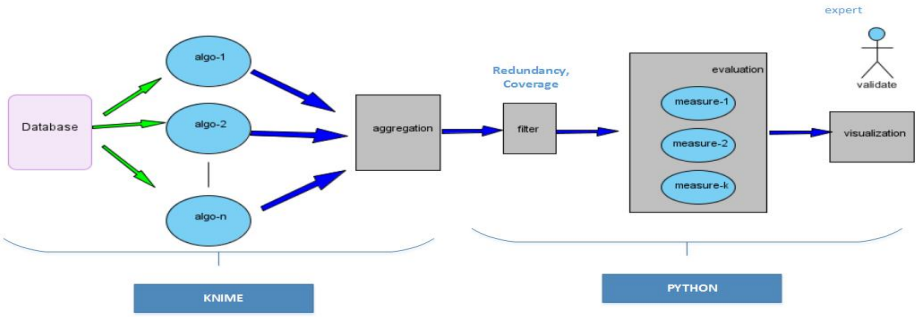


FIGURE 4: The proposed framework overview. Below denoted implementation choice (KNIME and Python)

The framework workflow is somewhat straightforward as it follows the same spirit as a KDD process described in Figure 3. In the most general form, the input genomic data will be first pre-processed to meet requirements. This pre-processing includes but is not limited to dimension filtering and data normalization.

In the mining task, since hidden patterns are not known in general, one cannot be sure where certain techniques are more suitable than others. We opted therefore to employ a variety of subspace clustering algorithms from diverse approaches to study their results. In order to do so, we execute these algorithms and analyze clusters found from different sources or from different runs of the same algorithm.

In specific, the results obtained from running those algorithms are grouped all together by a aggregator and will undergo some post-processing a centralized manner (including filtering, feature addition). After having finished its set of tasks, the aggregator will pass the results into the evaluation process where they are evaluated and validated using different measures.

Some methods of visualization are also implemented to help this process of validation. Relevant information (related to runtime, evaluation performance) is recorded since they might be useful for guiding future execution.

5.1 Data pre-processing, clustering and results aggregation

We implement this first part as workflows in Knime platform to be able to take advantages of available subspace algorithm components provided by [15].

Knime is a modern workflow-oriented data mining and machine learning platform that has a very user-friendly interface. A workflow in Knime is defined through its modular data pipelining concept which integrates various components called Nodes. Each node has a specific functionality (Extraction, Transformation, Loading, Visualizing...) and can be connected to others. We can therefore connect a set of corresponding nodes in order to establish a workflow in Knime, which can be stored and reused at ease.

Our workflow in Knime looks like the one described in Figure 5 :

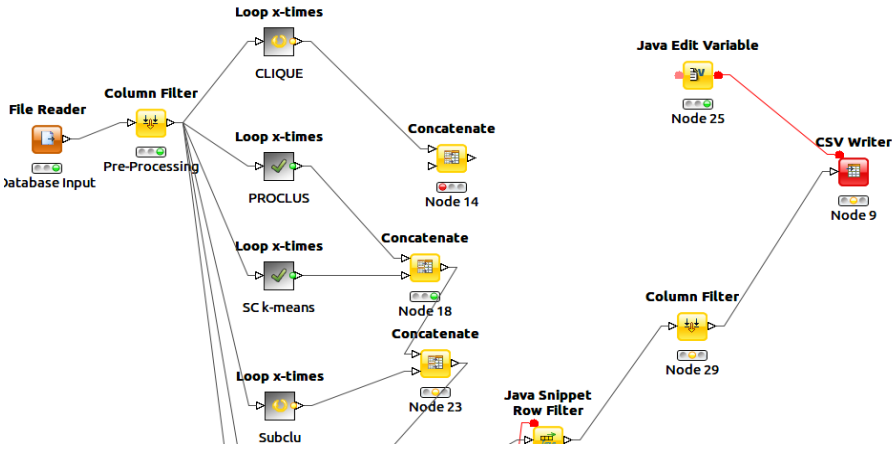


FIGURE 5: Screen capture of Part A implementation in Knime

5.1.1 Data mining process On this figure 5, “CLIQUE”, “PROCLUS”, “SCKMeans” and “SUBCLU” represent sub-processes that executes corresponding algorithms with different parameterization. Like most data mining algorithms,

parameter setting is a difficult task ; therefore, we try to provide a feature of parameter bracketing in order to facilitate the process which involves using nested loop. Given an algorithm, one should first define domains for its parameters and then discretize them into sequences of real values. We will then iterate over these values to take them as parameters for our algorithm. This choice of implementation is justified by the lack of flexibility of Knime regarding looping over a set of limited values. Also, we repeat several time the execution of an algorithm if it has non-deterministic behaviors.

5.1.2 Aggregation All the clustering results produced by different algorithms must be grouped all together by the aggregator. On the other hand, we are not interested in clusters that have either too many or too few objects. For instance, if we ever set the density threshold parameter of $CLIQUE$ too low, it might happen that we get a cluster that contain all the objects the input data. Obviously, this cluster is unlikely to be an interesting. Furthermore, a clustering composed of clusters of large-size is likely to be redundant. Therefore, we decide to add filter on this coverage criteria which retain only clusters that cover at least 1% and less than 70% of the total number of input objects. These thresholds are subject to modification.

At the end of this process, all the combined results will be written on a text file containing clusters with dimensions, name of the algorithms, values of parameters used and other related information. This text file will then be processed by the evaluation and visualization part.

5.2 Evaluation and visualization

Specifically, we define several criteria of goodness for a given clustering which will help us evaluate :

1. it should not be overly-redundant : As discussed in Section 4.2.1, a redundant clustering is undesirable since it may be overwhelming to process and provide little novel information. There are many reasons for redundancies to occur. For example, CLIQUE and many other grid-based algorithms are redundancy-prone since if a set of objects is dense in subspace S then it must also be dense in $S' \subset S$, resulting in having many clusters that are reported multiple times.
2. it should either :
   - exhibit high spatial coherence : a good cluster should also contains objects whose genomic location are spatially close to each other
   - have clusters that similar to results found in [37] which already proved to be make sense biologically, therefore it can be considered to be good reference clusterings. In this case we will use external measures presented in Section 4.2.2.

Based on ideas proposed in [14] which defines clustering redundancy in term of structural information, we will propose a redundancy removal procedure that

will help "filter" redundant clusters from clusterings. The redundancy of cluster is defined as a binary relation as follows : C is said to be redundant with regard to C' if C is similar to C' (in term of overlapping) and C is less interesting than C''s where cluster interestingness is measured by some quality function f (i.e. $f(C) < f(C')$)

Definition 3. ***Quality function :*** *A quality function f is a function that takes a cluster and return a value that measures its interestingness in term of some aspect. $f : \mathbb{CL} \to \mathbb{R}$ where $\mathbb{CL}$ the set of all subspace clusters.*

Such functions help quantify the concept of "interestingness" of a cluster. Now we discuss the concept of similarity :

Definition 4. ***Cluster similarity :*** *Given similarity parameters $r_{obj}, r_{dim} \in [0,1]$, a subspace cluster $C = (O,S)$ is said to be similar to $C' = (O',S')$, $sim_{c_{obj},c_{dim}}(C,C')$ iff $\frac{|O \cap O'|}{|O|} > c_{obj} \wedge \frac{|S \cap S'|}{|S|} > c_{dim}$*

Given a clustering, C is similar to C' if a fraction of its objects O as well as its relevant dimensions C have already be covered by C'. Note that this relation is neither transitive nor symmetric.

The key idea for our filtering process is that if the degrees of coverage $\frac{|O \cap O'|}{|O|}, \frac{|S \cap S'|}{|S|}$ are high enough and the quality of C is worse than C' then we will not "lose" much meaningful information if we remove C from the clustering. C is said to be *redundant* w.r.t C'.

Definition 5. ***Redundancy relation*** *Given a quality function f and 2 redundancy parameters r_{obj}, r_{dim} as above, binary relation $\prec_{f,red}$ is defined as follows : given $C = (O,S)$ and $C' = (O',S')$ $C \prec_{f,red} C' \iff sim_{c_{obj},c_{dim}}(C,C') \wedge f(C) < f(C')$.*

In this case, we say also that C is dominated by C' w.r.t f and r_{obj}, r_{dim}. It is easy to verify that, like the relation of similarity, this relation is neither transitive nor symmetric.

Definition 6. ***Redundancy-free clustering :*** *Use the same notions as above, a clustering $Cl = \{C_1, C_2, .., C_k\}$ is said to be redundancy-free iff $\forall C_i \in Cl, \neg(\exists C_j \in Cl : C_i \prec_{f,red} C_j)$.*

We note that a cluster is a member of such clustering if and only if either it is a dominant cluster (i.e. it dominates every other clusters to which it is similar) or it is isolated (similar to no other cluster). We can see it is related to the concept of *non-dominated* in multivariable-optimization.

Given an arbitrary non-empty clustering, it may or may not be redundancy-free but it has at least one such sub-clustering (for instance, those containing only one cluster). Consequently, we can filter redundant clusters from a clustering to obtain redundancy-free sub-clusterings. Evidently, we are interested only in finding the maximal sub-clustering and our redundancy removal filter will perform as such.

The clustering quality depends necessarily on 2 redundancy parameters and the quality function. We decided to opt to utilize $f(C) = |O_C| \times |S_C|$ based on the *quality of a twofold cluster* model in [17] (for $(a, b, c) = (0, 1, 1)$). The idea is to realize a trade-off between the number of objects and the number of dimensions. Indeed, for algorithms like $CLIQUE$ that depends on of fixed global density thresholds, it tends to bias clusters with lower dimensionality. After trying several combinations, we fixed $(r_{obj}, r_{dim}) = (0.5, 0.5)$. Evidently, a further study should be carried on to optimize these choices.

5.2.1 Evaluation measures In order to assert our second criteria previously defined in 2, we need to establish a measure of spatial coherence. We designed a very simple model to evaluate the genomic spatial compactness of a clustering. We project these objects on the chromosomal axe (visually it looks as displayed in our Figure 2). Next, we paint these ordered objects using colors corresponding to their cluster to which they belong (i.e. objects of cluster C_i are painted by color col_i). Then we "walk" along the axe counting the total number of color changes between 2 adjacent objects. We assume in addition that the process of assigning objects to clusters is random and follows the uniform distribution; we define our spatial coherence as the ratio between the numbers of color change (which is a random variable) over the expected value of this variable.

Definition 7. ***Spatial coherence :*** *Let $\mathbb{O} = \{o_1, ..., o_n\}$ a dataset of of n genomic objects and Cl a clustering of $k \leq n$ clusters (including the noise cluster). We use these k clusters to annotate the objects $\forall o \in \mathbb{O}, col(o) = i \iff o \in C_i$. Let nb the total number of color changes, the spatial coherence sc is defined as $sc = \frac{nb}{E(nb)}$ where $E(nb)$ expected of color change.*

Under the same hypothesis, we can actually compute the value of that expected value : $E = \frac{n^2 - \sum_{i=1}^{k} s_i^2}{n}$ where s_i the number of objects in cluster C_i.

Proof. We use the same notions as above, we will prove the formula using indicator variables and the linearity of expectation. Let X_t variables indicating whether object o_t and o_{t+1} have the same color. We have $X_t = \begin{cases} 1, & \text{if } col(o_t) = col(o_{t+1}) \\ 0, & \text{otherwise} \end{cases}$, we can verify that $nb = \sum_{i=1}^{n-1} X_i \rightarrow E(nb) = E(\sum_{i=1}^{n-1} X_i)$. For each $i \in \{1, \ldots, n-1\}$, the probability that the i^{th} and $(i+1)^{th}$ objects have the same label (belong to the same cluster) is $\frac{\sum_{j=1}^{k} s_j(s_j - 1)}{n(n-1)}$. Subtract from 1 to get the complement and multiply by $n - 1$ since they are all equal, we have $E = (\frac{n(n-1) - \sum_{i=1}^{k} s_i^2 - \sum_{i=1}^{k} s_i}{n(n-1)})(n-1)$. After simplifying using $\sum_{i=1}^{k} s_i = n$, we will get the required result (Q.E.D).

As nb can yield a value between k and n, we can deduce easily the range of this measure. We define furthermore an adjusted version of this measure such that it takes 0 as value when the total number of color changes equal to the expected value : $sc_{adj} = \frac{nb - E(nb)}{n - E(nb)}$. Intuitively, a spatially compact clustering will likely

have a small value in both cases. Another measure that might potentially be useful to consider is the *p-value* to see how likely it is to get a result at least as extreme as the one actually obtained, assuming the null hypothesis is true.

5.2.2 Visualization We develop specialized tools to visualize results. Each clustering result are projected on a the genomic axe and painted with cluster colors, like described in the previous section, depicted with relevant information. This way we can have a visual overview about the spatial compactness and the difference results between clustering.

All components of this *Evaluation and visualization* part were implemented in Python, including evaluation measures and visualization tools.

6 Experiments

6.1 Subspace Clustering Algorithms

In our experiments, we decided to choose among algorithms that were described in Section 4.1.1 a list of algorithms to perform cluster analysis. They were available as Knime components provided by [15] : CLIQUE, PROCLUS, SC-KMeans and SUBCLU. The algorithm STATPC was also chosen but due to a bug in their implemented, we could not manage to get it executed in a timely manner, despite our efforts to work in collaboration with the authors to fix it. However, we think that the remaining selected algorithms can still represent different approaches that were presented.

6.2 Data sources

As a reminder, Sexton et al. [37] performed cluster analysis on the whole genome using hierarchical (with average linkage) and k-means clustering to obtain and annotate 4 clusters which they called "4 classes", namely "Active", "Null", "HP1" and "HP1/Centromere". For k-mean clustering, only 4 profiles, namely H3K27me3, H3K4me, HP1 and H1 were used. We can argue that these four dimensions are revelant to the clustering that was found.

It justifies our choice to perform subspace clustering on supersets of these 4 profiles to see whether we can re-discover them. In order to do so, we demanded help from domain experts to be able to select epigenetic profiles for our datasets. We decided to work on 3 datasets in total. The first dataset contains 10 different epigenetic profiles ; the second one contains the 10 profiles of the first dataset and another additional 5 ; the third is the superset of the second with 5 other profiles. These profiles are supposed to be somehow relevant but it remains to be seen whether interesting clusters exist and in which subspace of dimensions.

The the experiments were conducted on a personal PC with a configuration of *Intel Core i7 CPU 920 -2.67GHz, 16Gb* memory, using the following parameters :

- *CLIQUE* : ξ (number of intervals per dimension) ranging from 5 to 62 taking 20 values, τ (density threshold) ranging from 8.10^{-3} to 1 taking 60 differents values.

- *PROCLUS* : k (number of clusters) ranging from 3 to 9 taking 7 values, average number of dimensions ranging from 2 to 8 taking 7 values and *the number of repetitions* is set to 5
- *SC-Kmeans* : k (number of clusters) ranging from 3 to 9 taking 7 values, the *subspace dimension size* ranging from 2 to 8 taking 7 values. The *maximum number* of combinations is set to 100 and *the number of repetitions* is set to 15.
- *SUBCLU* : ϵ (neighborhood radius) was set to 2^{-4},2^{-3}, 2^{-2}, 2^{-1} and 1, *minPts* (minimum number of points required in a neighborhood to form dense region) was set to 2^3,2^4, 2^5 and *the number of repetitions* is set to 5.

The experiment on the third dataset did not finish in time (more than two days, mostly due to SUBCLU) and would not be reported in this document.

6.3 Clustering of 3R chromosome using 9 dimensions

6.3.1 Data manipulation For the first step of our analysis, we imported a dataset consisting of the following 10 epigenetic profiles : H3K27me3, H3K4me3, HP1, Suw(Hw), DNase, H3K27ac, CP190 and H1, CTCF, Beaf-32. The first 7 were provided by the Vital-IT high performance computing platform of the Swiss Institute for Bioinformatics, whereas the last 3 were extracted from the datasets from [11], available on the Gene Expression Omnibus (GEO)- under the reference GSE22069. Unfortunately, one of the dataset (Su(Hw)) turned out to be un-usable, thus reducing our dataset to 9 profiles.

The 7 epigenetic profiles retrieved from the platform Vital-IT (H3K27me3, H3K4me3, HP1, DNase, H3K27ac, CP190) which were originally in the bigWig format (.bw) were first converted into the bedGraph (.bedgraph) using the Linux version of the UCSC's stand-alone converter tool, available at `http://hgdownload.cse.ucsc.edu/admin/exe/linux.x86_64/`. The bedGraph is a csv-like line-oriented file format which provides four bits of information for epigenetic data with each line represent an object. The first three bits of information refer to the object's coordinates : the chromosome, the start position, the end position ; while the last bit refers to a continuous value, in this case the frequency of interaction (signal) between the object in question w.r.t some protein.

The datasets from [11] were processed by filtering irrelevant profiles to keep only those 3 profiles of interest (H1, CTCF, Beaf-32) and then converting them to bedGraph format.

We limited our interest for the time being only on a region of 2.10^6 base pairs spanning from $12e6$ to $14e6$ of the 3R chromosome. The reason behind this choice is that this region was well covered by Sexton et al. and we wish to compare our result with theirs. All the profiles were split to intervals of size 1000 base pair which gives us 2000 equal-size objects (or bins). The value of a feature for any of our objects is then simply the average of the signal of the corresponding profile over the interval associated ot this object.

Next, since our datasets were generated by multiple techniques in different labs and are of different nature, they should be first normalized. We transformed using the log-quantiles following instructions given in [37] : $x_{norm} = -2log(1 -$

$\frac{\text{rank}(x_{raw})}{\max(\text{rank}(x_{raw}))}$). When rank=max(rank), $log(0)$ is not defined, f takes then the maximal value. We also used the average tie strategy when computing the rank.

Finally, these datasets were joined together by their genomic location, to form a unique data file (which looks like 6). Most of the work in step this was performed using R.

chr	start	end	H1	H3K4Me3	H3K27Me3	HP1	H3K27ac	DNASe	CP190	BEAF32	CTCF
chr3R	12000000	12001000	8.455464e-01	1.94619356	1.282789701	2.070966521	1.275786313	1.248107862	1.011587974	1.875672e+00	4.930934e-01
chr3R	12001000	12002000	1.948976e+00	0.77102743	1.015957574	0.944804346	0.413115187	0.190174004	0.979942348	1.592919e+00	5.426684e-01
chr3R	12002000	12003000	1.096578e+01	0.28015944	1.171368418	0.795859283	0.066427362	1.298672743	0.503282012	1.771027e+00	8.520421e-01

Genomic coordinates

9 epigenetic profiles

FIGURE 6: Bedgraph-like datafile

6.3.2 Cluster analysis We performed the subspace cluster analysis using different parameter set while repeating several times (5 for PROCLUS and SUBCLU, 15 for SC-Kmeans) for algorithms of nondeterministic behaviors (SC-Kmeans, PROCLUS and SUBCLU). After having finished, we applied the coverage filter and redundancy removal that was described in Section 5.1.2 which resulted in obtaining 1706 CLIQUE, 250 PROCLCUS and 226 SC-KMeans clusterings. We noted that none of the SUBCLU clusters are retained as they either have too many or too few objects that most of the CLIQUE clusterings contain only one single cluster (1079 clusters for 1076 clusterings) since they got too many subspaces.

Next, we computed the score of Entropy, F-Measure, Spatial Coherence, CE and RNIA (details given in Section 4.2.2) for each of the 1552 obtained clusterings and ranked them w.r.t each of the mentioned measures. We plotted the results using our visualization tools which give figures where the x-axis is the genomic location order and the objects are painted in colors corresponding to their cluster.

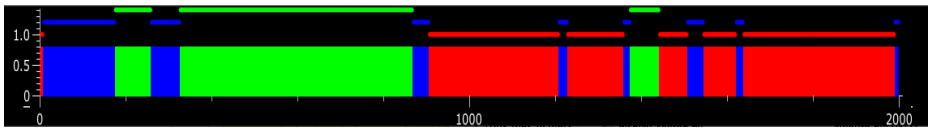


FIGURE 7: Clustering obtained by Sexton et al. for the 12e6-14e6 region of the 3R chromosome

We also looked at the relationship between these measures by computing the Spearman's rank correlation coefficients between them. We use R to display the scatter-plot matrix which contains the correlation coefficient between pairs of

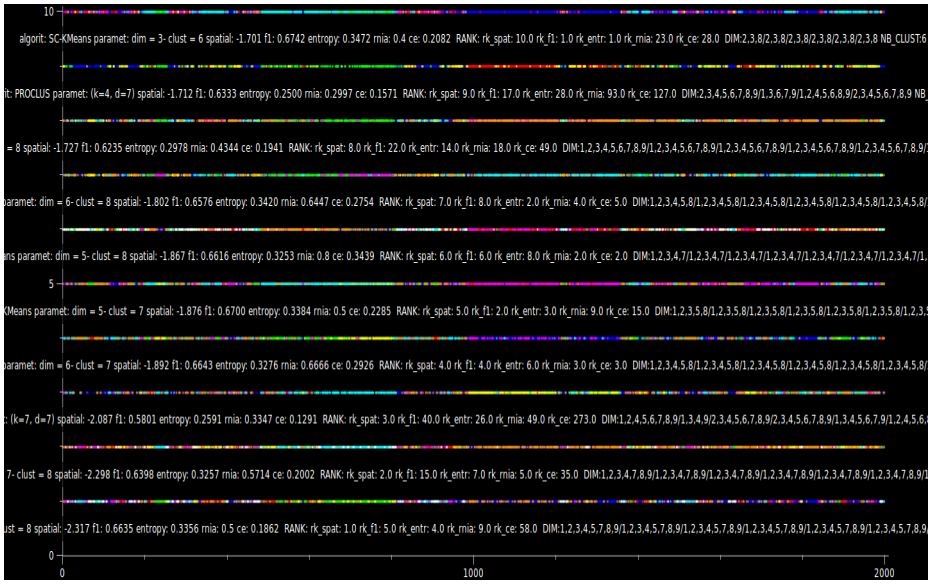
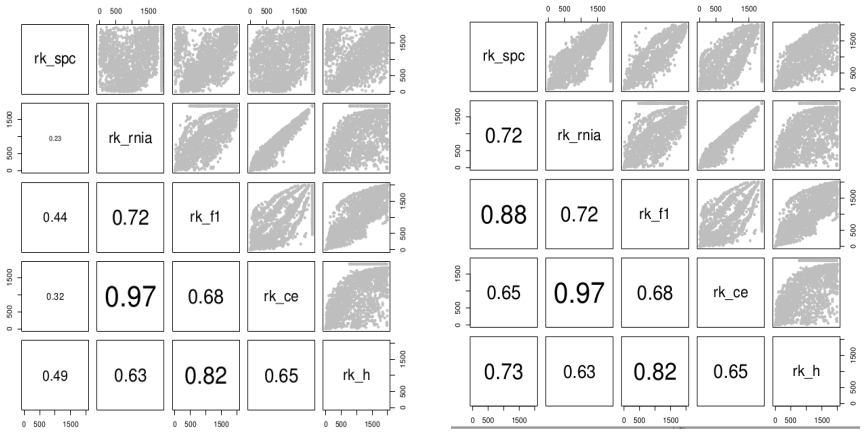


FIGURE 8: Visualization of the top 10 performing clustering w.r.t spatial coherence with associated information on 3R, 12e6-14e6, 9 dimensions

FIGURE 9: Visualization of the top 10 performing clustering w.r.t CE score on 3R, 12e6-14e6, 9 dimensions

variables, making it easier to investigate them pairwise. We plotted both adjusted and non-adjusted version of spatial coherence in Figure 10. In this figure, *spc*, *rnia*, *f1*, *ce*, *h* refer to *spatial coherence*, *RNIA*, *F1*, *CE* and *Entropy* respectively.



(a) Not adjusted spatial coherence (b) Adjusted spatial coherence

FIGURE 10: Spearman's rank correlation coefficients between measures on 3R, 12e6-14e6, 9 dimensions

As we can see on the figure, all variables seem to have a strong positive correlation with one another with the highest coefficient between RNIA and CE. In particular, the spatial coherence measure proves to agree with other measures and the adjusted version is slightly more correlated to others in comparison to the non-adjusted one. Indeed, for the adjusted version, it achieves coefficients from 0.65 against CE to a pretty high 0.88 against F1-score. We are now interested only in the adjusted spatial coherence.

Next, we analyzed the top ranked clusterings by adjusted spatial coherence to see how they performed against other measures, in particular with CE and F1-score. As discussed earlier, CE is the improved version of RNIA while F1-score evaluates purity and coverage. It turns out that the top ranked clusterings w.r.t spatial coherence are also ranked pretty high w.r.t both CE and F1, as displays Table 1 where each cell denotes the rank of the clustering by the corresponding measure (out of 1552 clusterings).

Analyzing these top 10 ranked by spatial coherence, 8 were found by SC-Kmeans with a average dimensionality of 6.25. The other 2 were found by PROCLUS. We found that the four dimensions selected in [37] with the help of experts were also considered as relevant in 8 out of the top 10 clusterings (ranked by adjusted spatial coherence). A careful look shows that they were actually obtained from running different parameter sets (recall that we repeat several

Spatial	RNIA	F1	CE	Entropy
1	9	5	58	4
2	5	15	35	7
3	49	40	273	26
4	3	4	3	6
5	9	2	15	3
6	2	6	2	8
7	4	8	5	2
8	18	22	49	14
9	93	17	127	28
10	23	1	28	1

TABLE 1: Top 10 out of 1552 total clusterings by adjusted spatial coherence on 3R along with the ranks w.r.t other measures

times for each parameter set, it might occur that some good performers are just repetitions).

6.4 Clustering of 3R chromosome using 14 dimensions

6.4.1 Data manipulation Following the same procedure as given in the Section , we first imported in addition to the existing 9 profiles another 5s namely : PolII, CBP, H3K9Ac, H3K9Ac, H3K27ac and ChIP.H3K9me2.normH3. Then we repeated our previous experiments on this new dataset.

The number of total subspace clusters found by CLIQUE increased rapidly to 31649 whereas for PROCLUS (1015) and SC-Kmeans (207) did not change much. In this step we used the same number (5 for PROCLUS and SUBCLU, 15 for SC-Kmeans) of execution repetition for non-deterministic algorithms due to run-time concern ; however it is to be noted that it should be set higher since we have more dimensions to explore. SUBCLU again did not have any clusterings left after the filtering process.

The issue with CLIQUE regarding the number of cluster per clustering repeats again. They outnumber the rest in term of clustering number and in general are ranked pretty low. We decided hence to remove them from this analysis step which reduced the number of clusterings to 298 (250 of PROCLUS and 48 of SC-Kmeans).

In term of Spearman's rank correlation coefficients 13, we observed that in general the spatial coherence is less correlated to other measures but the adjusted spatial coherence still has a pretty strong score.

When analyzing the top performers, it is interesting to note that this time, 7 out of 10 best clusterings in term of spatial coherence were found by PROCLUS (the other 3 were found by SC-KMeans), in contrast to only 2 in the previous result. We note also that the 2 clustering obtained by SC-Kmeans are actually 2 repetions of the same parameter set wheareas PROCLUS are all of different parameter sets.

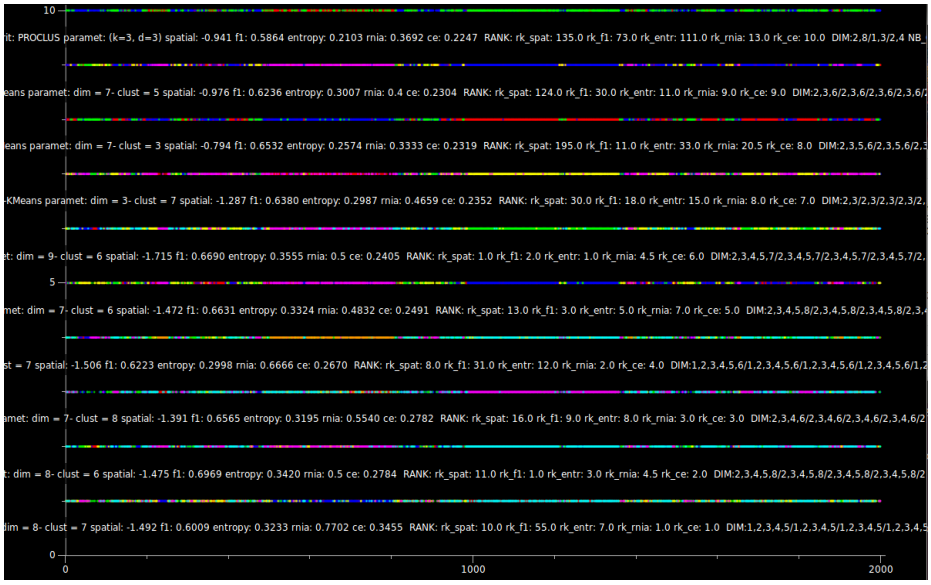


FIGURE 11: Visualization of the top 10 performing clustering w.r.t of spatial coherence with associated information on 3R, 12e6-14e6, 14 dimensions

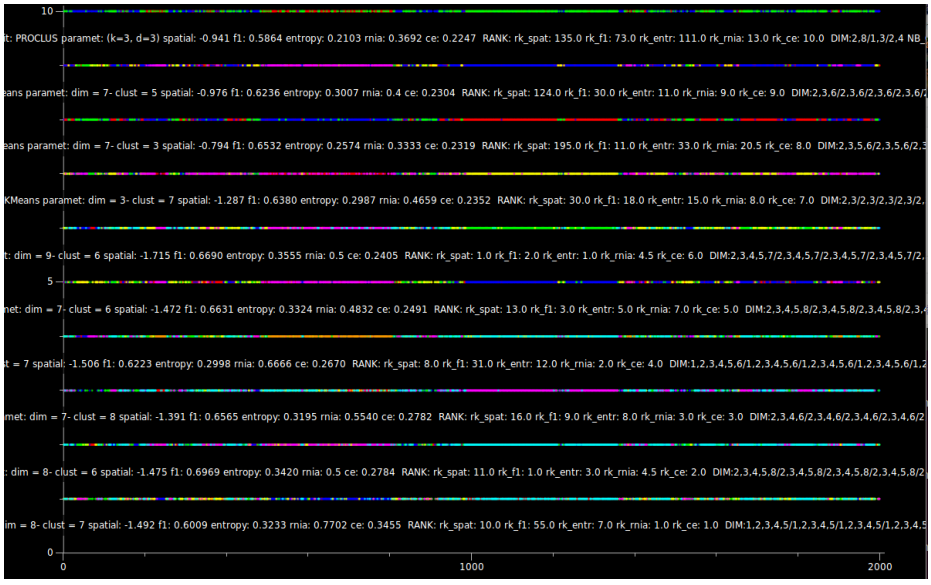
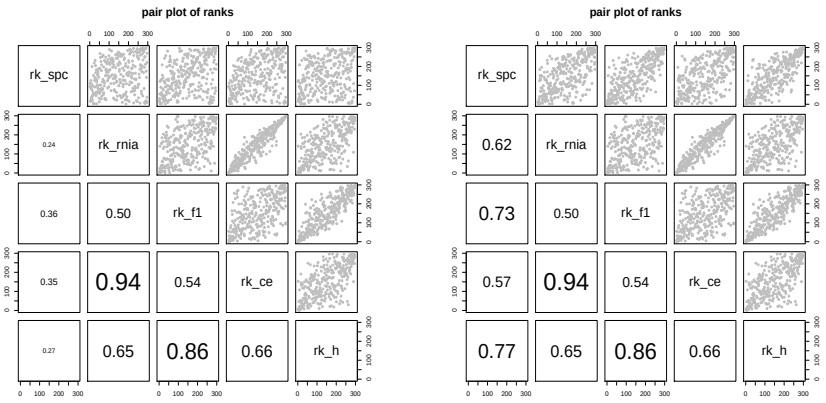


FIGURE 12: Visualization of the top 10 performing clustering w.r.t of CE score on 3R, 12e6-14e6, 14 dimensions



(a) Not adjusted spatial coherence (b) Adjusted spatial coherence

FIGURE 13: Spearman's rank correlation coefficients between measures on 3R, 12e6-14e6, 14 dimensions

On the other hand, even though some of the best ranked by spatial coherence performed well in other measures, for CE most of the clusterings were not very well evaluated.

Spatial	RNIA	F1	CE	Entropy
1	4.5	2	6	1
2	61.0	16	108	9
3	75.0	41	81	44
4	15.0	100	28	29
5	182.0	50	181	76
6	27.0	21	27	32
7	185.0	15	136	24
8	2.0	31	4	12
9	181.0	88	173	137
10	1.0	55	1	7

TABLE 2: Top 10 clusterings out of 298 total clusterings ranked by adjusted spatial coherence on 3R, 12e6-14e6, 14 dimensions

The obtained results indicate that the measures de spatial coherence (notably the adjusted version) allows to discover clusterings of good quality as they have high scores of CE and F1 in comparison to clusterings found by Sexton et al. [37]. In addition, our proposed method also helps to find automatically 4 relevant

dimensions that were selected by domain experts. These results were obtained without using the *Hi-C* data (as well as the so-called physical domains).

7 Conclusion and perspectives

In our work, we studied cluster analysis of high-dimensional genomic data using techniques from subspace clustering, a paradigm that aims to deal with such high-dimensional data. We presented a framework for comparing and evaluating subspace clustering results found by different algorithms and studied their application on real genomic dataset. We discussed the problem of clustering *meaningfulness* and proposed a process that takes into account multiple aspects : redundancy, spatial coherence and similarities to reference clusters. To address redundancy, we proposed a post-processing redundancy removal procedure that help to "clean" the obtained results. We also developed measures for spatial coherence evaluation. Concerning reference clustering comparison, we studied and implemented a set of potentially suitable external evaluation measures.

We performed experiments on two different datasets and reported the results. Our results indicate that subspace clustering seemed promising for our needs. Besides, we showed that it might be possible to get meaningful clusters without relying on the knowledge of the physical domains (from Hi-C data) which are not always available. In addition, we reported evidences that the spatial coherence measures proposed in this work is effectively helpful for selecting meaningful clusterings.

Regarding future work, we would like to continue to carry out this cluster analysis on other regions of the genome at different spatial scales and with a larger pool of algorithms. We have also intention of implementing other evaluation measures such as *V-Measure* or the *p-value* of the spatial coherence.

Références

[1] Charu C Aggarwal et al. "Fast algorithms for projected clustering". Dans : *ACM SIGMOD Record* 28.2 (1999), p. 61–72.

[2] Rakesh Agrawal et al. "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications". Dans : *SIGMOD Conference*. 1998, p. 94–105.

[3] Ira Assent et al. "INSCY : Indexing subspace clusters with in-process-removal of redundancy". Dans : *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE. 2008, p. 719–724.

[4] Kevin S. Beyer et al. "When Is "Nearest Neighbor" Meaningful ?" Dans : *Proceedings of the 7th International Conference on Database Theory*. ICDT '99. London, UK, UK : Springer-Verlag, 1999, p. 217–235. isbn : 3-540-65452-6.

[5] Christian Bohm et al. "Density connected clustering with local subspace preferences". Dans : *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*. IEEE. 2004, p. 27–34.

[6] Sergey Brin, Rajeev Motwani et Craig Silverstein. "Beyond market baskets : generalizing association rules to correlations". Dans : *ACM SIGMOD Record*. T. 26. 2. ACM. 1997, p. 265–276.

[7] Chun-Hung Cheng, Ada Waichee Fu et Yi Zhang. "Entropy-based subspace clustering for mining numerical data". Dans : *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 1999, p. 84–93.

[8] David L Donoho. "High-dimensional data analysis : The curses and blessings of dimensionality". Dans : *AMS Math Challenges Lecture* (2000), p. 1–32.

[9] Martin Ester et al. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". Dans : *KDD*. 1996, p. 226–231.

[10] Usama Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth et al. "Knowledge discovery and data mining : Towards a unifying framework". Dans : *Knowledge Discovery and Data Mining* (1996), p. 82–88.

[11] Guillaume J Filion et al. "Systematic Protein Location Mapping Reveals Five Principal Chromatin Types in Drosophila Cells". Dans : *Cell* 143.2 (2010), p. 212–224.

[12] Benjamin CM Fung, Ke Wang et Martin Ester. "Hierarchical document clustering using frequent itemsets". Dans : *Proceedings of the SIAM international conference on data mining*. T. 30. 5. 2003, p. 59–70.

[13] Sanjay Goil, Harsha Nagesh et Alok Choudhary. "MAFIA : Efficient and scalable subspace clustering for very large data sets". Dans : *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1999, p. 443–452.

[14] Stephan Günnemann, Brigitte Boden et Thomas Seidl. "DB-CSC : a density-based approach for subspace clustering in graphs with feature vectors". Dans : *Machine Learning and Knowledge Discovery in Databases*. Springer, 2011, p. 565–580.

[15] Stephan GÜNNEMANN et al. "A Subspace Clustering Extension for the KNIME Data Mining Framework". Dans : *12th IEEE International Conference on Data Mining Workshops - Demonstration Session, ICDM Workshops, Brussels, Belgium*. 2012, p. 886–889.

[16] Stephan GÜNNEMANN et al. "External evaluation measures for subspace clustering". Dans : *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM. 2011, p. 1363–1372.

[17] Stephan GÜNNEMANN et al. "GAMer : a synthesis of subspace clustering and dense subgraph mining". Dans : *Knowledge and Information Systems* (2013), p. 1–36.

[18] Maria HALKIDI, Yannis BATISTAKIS et Michalis VAZIRGIANNIS. "On clustering validation techniques". Dans : *Journal of Intelligent Information Systems* 17.2-3 (2001), p. 107–145.

[19] Jiawei HAN, Micheline KAMBER et Jian PEI. *Data mining : concepts and techniques*. Morgan Kaufmann, 2011.

[20] Karin KAILING, Hans-Peter KRIEGEL et Peer KRÖGER. "Density-connected subspace clustering for high-dimensional data". Dans : *Proceedings of the Fourth SIAM International Conference on Data Mining*. T. 4. 2004.

[21] Peter V KHARCHENKO et al. "Comprehensive analysis of the chromatin landscape in Drosophila melanogaster". Dans : *Nature* 471.7339 (2010), p. 480–485.

[22] Hans-Peter KRIEGEL, Peer KRÖGER et Arthur ZIMEK. "Clustering high-dimensional data : A survey on subspace clustering, pattern-based clustering, and correlation clustering". Dans : *ACM Transactions on Knowledge Discovery from Data* 3.1 (2009).

[23] Hans-Peter KRIEGEL et al. "A Generic Framework for Efficient Subspace Clustering of High-Dimensional Data". Dans : *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005)*. 2005, p. 250–257.

[24] Tilman LANGE et al. "Stability-based validation of clustering solutions". Dans : *Neural computation* 16.6 (2004), p. 1299–1323.

[25] Michael LEVANDOWSKY et David WINTER. "Distance between sets". Dans : *Nature* 234.5323 (1971), p. 34–35.

[26] Sara C MADEIRA et Arlindo L OLIVEIRA. "Biclustering algorithms for biological data analysis : a survey". Dans : *Computational Biology and Bioinformatics, IEEE/ACM Transactions on* 1.1 (2004), p. 24–45.

[27] Marina MEILĂ. "Comparing clusterings an information based distance". Dans : *Journal of Multivariate Analysis* 98.5 (2007), p. 873–895.

[28] Emmanuel MULLER et al. "Discovering multiple clustering solutions : Grouping objects in different views of the data". Dans : *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*. IEEE. 2012, p. 1207–1210.

[29] Emmanuel MÜLLER et al. "Evaluating clustering in subspace projections of high dimensional data". Dans : *Proceedings of the VLDB Endowment* 2.1 (2009), p. 1270–1281.

[30] Emmanuel MÜLLER et al. "OpenSubspace : An open source framework for evaluation and exploration of subspace clustering algorithms in WEKA". Dans : (2009).

[31] Lance PARSONS, Ehtesham HAQUE et Huan LIU. "Subspace clustering for high dimensional data : a review". Dans : *ACM SIGKDD Explorations Newsletter* 6.1 (2004), p. 90–105.

[32] Anne PATRIKAINEN et Marina MEILA. "Comparing subspace clusterings". Dans : *Knowledge and Data Engineering, IEEE Transactions on* 18.7 (2006), p. 902–916.

[33] Cecilia M PROCOPIUC et al. "A Monte Carlo algorithm for fast projective clustering". Dans : *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*. ACM. 2002, p. 418–427.

[34] Anand RAJARAMAN et Jeffrey David ULLMAN. *Mining of massive datasets*. Cambridge University Press, 2011.

[35] Andrew ROSENBERG et Julia HIRSCHBERG. "V-measure : A conditional entropy-based external cluster evaluation measure". Dans : *Proceedings of the 2007 EMNLP-CoNLL)*. T. 410. 2007, p. 420.

[36] Karlton SEQUEIRA et Mohammed ZAKI. "SCHISM : A new approach for interesting subspace mining". Dans : *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*. IEEE. 2004, p. 186–193.

[37] T SEXTON et al. "Three-dimensional folding and functional organization principles of the Drosophila genome." Dans : *Cell* 148.3 (2012), p. 458.

[38] Kelvin SIM et al. "A survey on enhanced subspace clustering". Dans : *Data Mining and Knowledge Discovery* 26.2 (2013), p. 332–397.

[39] René VIDAL. "A tutorial on subspace clustering". Dans : *IEEE Signal Processing Magazine* 28.2 (2010), p. 52–68.

[40] Man Lung YIU et Nikos MAMOULIS. "Frequent-pattern based iterative projected clustering". Dans : *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE. 2003, p. 689–692.

[41] Ying ZHAO et George KARYPIS. "Criterion functions for document clustering : Experiments and analysis". Dans : *Machine Learning* (2001).

[42] Arthur ZIMEK. "Correlation Clustering". PhD dissertation. Ludwig-Maximilians-Universität München, 2008.

8 Acknowledgement

First and foremost, I would like to express my sincerest gratitude to my principal supervisor Dr. Christophe Rigotti for offering me this research internship opportunity and for his continuous support, unequivocal patience and excellent knowledge. His invaluable guidance helped me in all the time of research and writing of this thesis. One could not simply ask for a better mentor, on both an academic and a personal level,

My sincere thanks also go to my two co-supervisors, Dr. Marion Leleu and Dr. Jacques Rougemont for inviting me to come visit their research group and for providing me with their gennomic datasets, insightful comments and hard questions. This thesis would not have been finished without their helpful advice and their immerse knowledge.

I would like also to thank Dr. Stephan Günnemann (at Carnegie Mellon University) for providing me the implementations of subspace clustering algorithms and for his willingness to answer all of my questions.

I am also grateful to the Beagle Team and its staff for providing me with an excellent atmosphere and all ressources necessary for doing this research.

I would like thank my fellow labmates for the stimulating discussions with them and for their occasional free cookies.

Finally, I would like to acknowledge my parents for their uncondional support for me throughout my life.