

Michele Fratello^{1,2}, Luca Cattelani^{1,2}, Antonio Federico^{1,2}, Alisa Pavel^{1,2}, Giovanni Scala³, Angela Serra^{1,2}, Dario Greco^{1,2,4,5,*}

¹ Faculty of Medicine and Health Technology, Tampere University, Tampere, Finland.

² BioMediTech Institute, Tampere University, Tampere, Finland

³ Department of Biology, University of Naples Federico II, Naples, Italy

⁴ Institute of Biotechnology, University of Helsinki, Helsinki, Finland

⁵ Finnish Center for Alternative Methods (FICAM), Finland.

* Correspondence: dario.greco@tuni.fi;

Unsupervised algorithms for microarray sample stratification

Running head: Unsupervised algorithms for microarray sample stratification

Abstract

The amount of data made available by microarrays gives researchers the opportunity to delve into the complexity of biological systems. However, the noisy and extremely high-dimensional nature of this kind of data poses significant challenges. Microarrays allow the parallel measurement of thousands of molecular objects spanning different layers of interactions. In order to be able to discover hidden patterns, the most disparate analytical techniques have been proposed. Here, we describe the basic methodologies to approach the analysis of microarray datasets that focus on the task of (sub-)group discovery.

Keywords (5-10): microarray; clustering; unsupervised learning; dimensionality reduction; group discovery

Introduction

DNA microarrays are a high throughput technology able to profile tens of thousands of different molecules at the same time (1, 2). However, a classical transcriptomic experiment is usually related to tens of tested samples, making transcriptomics data subject to the curse of dimensionality (2, 3). Moreover, microarray data is also known to be affected by technical and experimental noise (1–3).

Unsupervised learning techniques have been widely applied in the analysis of microarray experiments to uncover hidden structures in the data. Common unsupervised techniques include projection and clustering methodologies. Projection techniques are used for data visualization and dimensionality reduction in order to retain useful information and filter out the experimental noise. Clustering techniques, on the other hand, are used to categorize the experimental samples based on their response across the transcriptomics profile. Indeed, clustering methodologies group together samples that are most similar between themselves and most dissimilar with respect to the rest.

Data projection is a universally employed method in the analysis of big data. It consists of projecting the data in a low-dimensional space to explore the relationships among tested samples for visual inspection and exploratory analyses. In the specific case of biomedical research, data projection is arguably utilized in any study where exploratory data analysis is necessary, while its use for dimensionality reduction purposes plays a crucial role in pattern recognition, classification and clustering.

One of the most widespread applications of unsupervised algorithms is patient stratification for complex human diseases (4). Complex or multifactorial diseases are, by definition, pathological conditions due to both genetic and environmental perturbations (5). Both of these factors contribute to the modification of the genetic and epigenetic makeup of the cells, causing, in turn, the expression of a pathological phenotype. Nowadays, the most studied human multifactorial disease is cancer. Indeed, a cancerous phenotype is the result of genetic and epigenetic alterations leading the transformed cell to the acquisition of novel and non-physiological characteristics. Loss of differentiation, increase of cell cycle speed and low rate of cell apoptosis are the main characteristics of a cancer cell, leading to an overly variable phenotype within the affected population. Such characteristics make cancer one of the most challenging diseases to be treated with well-defined pharmacological strategies. For this reason, the classification of a complex disease, like cancer, into clinically relevant sub-types may help to identify the molecular determinants of the different sub-phenotypes (4) and, therefore, of the different outcomes of a specific therapeutic approach.

Large-scale omics technologies have become cost- and time-effective. This resulted in a growing number of publicly available large-scale (multi-)omics datasets in order to address these kinds of problems (6). These data usually measure different biological variables for the same samples, allowing the integration of complementary information and bringing a deeper insight into the same biological process. For instance, multiple layers of omics data for thousands of cancer patients, along with clinical data, are publicly accessible at The Cancer Genome Atlas and the International Cancer Genome Consortium (7). Such large-scale efforts triggered the development of countless methods and algorithms for patient stratification and the identification of disease

endotypes. These resources shed light on clinically relevant molecular sub-types for a number of cancers, and further resulted in successful tailored pharmacological treatments (8). While sample stratification finds its widest use in the identification of cancer endotypes, it is by no means its only application. For instance, unsupervised clustering can be utilized to identify drugs showing similar mechanisms of action in large experimental profiling (9). Similarly, sample clustering can be applied in toxicity prediction experiments, as reported in (10), where unsupervised hierarchical clustering has been employed in order to prepare the data for QSAR modelling.

The most common clustering methodologies can be categorized in hierarchical, partitive, density-based and fuzzy. Also, specific optimization methodologies, called biclustering, that aim at grouping both the samples and the features of the omics data at the same time, are available.

Multiple issues and challenges are connected to the computational analysis of omics data. First, it should be noted that each unsupervised algorithm relies on different assumptions on data distributions and different input parameters. For example, most of the clustering algorithms take as input the number of groups to detect, that is often arbitrarily selected by the user. To address this issue, some methodologies have been developed to estimate the optimal number of clusters from the data at hand. Moreover, changing the initial parameters setup of an algorithm can change the outcome, leading to potential issues in the stability of the results. This is even more true when different algorithms are applied to the same dataset. Furthermore, since the aim of unsupervised algorithms is to group similar objects, the results are strongly dependent on the type of distance or dissimilarity measure used in the analysis. These issues can be addressed by using consensus clustering methodologies, where the results coming from a pool of different algorithms with different experimental setups are merged in order to obtain a more stable and

reliable result. A further concern is related to the evaluation of the goodness of unsupervised algorithm solutions. Multiple metrics that evaluate the goodness of the cohesion into the groupings or compare the results with some prior knowledge on the distribution of the data are available.

However, there is no one definitive answer on how to choose the best clustering algorithm or distance/similarity measures. Some of them are just more suitable for particular dataset structures (e.g. continuous or binary data). In order to always pick up the most suitable algorithm, it is important to have a deep understanding of their advantages and disadvantages.

In this chapter, we describe the basic concepts of distances and similarity metrics, projective and clustering algorithms, and their evaluation metrics. Moreover, we discuss the multi-omics clustering approaches and provide examples of their application to microarray data analysis.

Methods

Metrics for unsupervised learning

Multiple metrics are available to compare expression patterns from microarray data. However, the choice of which metric to use for group discovery will affect the results to different extents (11, 12).

In general, metrics can be divided into distances or similarities. A distance measures how far apart two data points $\mathbf{P} = (p_1, \dots, p_n)$ and $\mathbf{Q} = (q_1, \dots, q_n)$ lie in a space. A distance is a metric if it is always non-negative as well as i) symmetric, ii) must fulfil the triangle inequality, and iii) must be zero only when measuring the distance of a point from itself.

Popular distance metrics (for numeric data) are the Euclidean distance, the Manhattan distance and the Chebychev distance, all of which are special cases of the parametric Minkowski distance (Table 1). The shape of the clusters detected with any Minkowski distances also varies with the parameter p , *e.g.* the Euclidean distance produces spherical clusters, the Manhattan distance produces diamond-shaped clusters and the Chebychev distance produces squared clusters. The Minkowski distances are influenced by any imbalances in the scale of any component of the data points. As $p \rightarrow \infty$ the corresponding distance is more and more influenced by the components of the data point with the largest scale. To overcome this problem, it is recommended to normalize the scale of each component of the data (**12**). Another distance that is not affected by the scale of the vector components is the Mahalanobis distance (Table 1), a data-driven generalization of the Euclidean distance that uses the inverse covariance matrix Σ^{-1} to “whiten” the data. Whitening has the double advantage of scaling each component to have the same variance as well as suppressing any correlation among pairs of components. Clustering data using the Mahalanobis distance produces ellipsoidal clusters. However, it requires the estimation of the inverse of a, usually large, covariance matrix that is a difficult problem of its own (**13**).

When all the components of the data points \mathbf{P} and \mathbf{Q} are binary, commonly used distances are the Jaccard distance (Table 2). An alternative is the Hamming distance (Table 2), defined as the number of differing bits in the two binary data points.

A distance is a semi-metric if it fulfils conditions i and iii, but not necessarily ii. On the other hand, a quasi-metric needs to be always non-negative and fulfil conditions ii and iii but not

necessarily i. This allows for the distance from P to Q to be different with respect to the distance from Q to P .

In general, a similarity measures the opposite of a distance, it tries to quantify how “similar” two data points are rather than how far apart they are. Some efforts have been made to provide a formal definition of a similarity (**14, 15**). As defined by Chen et al. a valid similarity metric should be i) symmetric, ii) fulfil the triangle inequality, iii) the similarity between the same object needs to be non-negative, iv) if $s(P, P) = s(Q, Q) = s(P, Q)$ holds, then $P = Q$ and v) the similarity between any two different data points cannot be larger than the similarity between a data point and itself.

A simple and often used method to switch between a distance and a similarity (even though it is not guaranteed that the derived distance/similarity satisfies all the requirements to be a proper metric) is to take its reciprocal $s = \frac{1}{(1+d)}$ (**16**) (adding 1 to the denominator avoids numerical issues), or for normalized values in $[0, 1]$, $s = 1 - d$ can be used.

Popular similarity measures are the cosine similarity for real values and the Jaccard index or the Dice’s coefficient for binary data (Table 2).

Correlation measures can also be used, which are popular in the analysis of microarray data (**17**). Multiple correlation scores exist that can be categorized into parametric and non-parametric (Table 2). In both cases, their values lie in the range $[-1, 1]$, with -1 meaning negative proportionality and $+1$ positive proportionality. A correlation value of 0 means that no linear statistical relation between the two data points is present. Parametric scores make an assumption on the distribution of the data (*e.g.* the assumption underlying the Pearson correlation coefficient

is that the data are normally distributed), whereas non-parametric scores are less stringent. When the assumptions hold, parametric scores have more statistical power than non-parametric ones. In the case of gene expression microarray data, the log-ratio measurements are approximately normally distributed, thus the assumptions of the Pearson correlation are satisfied. In the case of data that are not normally distributed other non-parametric correlation scores, such as the Spearman and Kendall correlation, can be used.

Multiple tests are available to determine the goodness-of-fit of data to a normal distribution, such as the Shapiro-Wilk test (*18*) or the D'Agostino-Pearson omnibus test (*19*). Moreover, a good practice is to visually inspect the empirical distribution (*e.g.* histograms/density plots, quantile-quantile plots) of the data to determine whether it deviates significantly from distributional assumptions made.

Dimensionality Reduction

Microarray technology allows the expression of tens of thousands of biological entities, such as mRNA, miRNA or copy numbers to be profiled in parallel. However, it is unfeasible for any microarray experiment to collect a comparable number of samples and therefore, this data may be subject to the curse of dimensionality. Also, not all the measured genes are relevant for all the experiments: it is common for most genes to be unaffected by the experimental conditions under study. In addition, most genes are organized in overlapped functional groups, implying a degree of correlation among them.

As opposed to supervised learning tasks like the classification of samples or regression, in the unsupervised setting of (sub-)group discovery, there is no information that can guide the

detection of relevant genes or the learning of predictive representations. Thus, it is helpful to reduce the dimensionality of the data (*i.e.* discard irrelevant and/or redundant content) without losing much information. This preprocessing will favour downstream clustering analyses not only reducing the chance of overfitting but also reducing the computational cost **(20)**.

In the following, we assume that the microarray dataset is represented by the matrix \mathbf{X} . \mathbf{X} has dimensions $n \times p$, with $p \gg n$, where n is the total number of samples and p is the number of measured entities across the samples, we further assume that \mathbf{X} is centred, *i.e.* the columns of \mathbf{X} have zero mean.

Principal Components Analysis (PCA)

PCA is a dimensionality reduction methodology that estimates a linear transformation \mathbf{W} such to preserve as much variance of the projected data $\mathbf{X} = \mathbf{X}\mathbf{W}$ as possible. The complete transformation matrix \mathbf{W} can have dimensions at most $k \times k$, where $k \leq \min(n, p)$. Each column of \mathbf{W} corresponds to a principal component (PC) which is estimated as a weighted combination of the contributions of all the original features. To reduce the impact of noise and compress the data, it is common to only consider the first $q < k$ components.

It can be shown that PCs correspond to the eigenvectors of the covariance matrix of the data $\Sigma = \mathbf{X}^T\mathbf{X}$ sorted according to the proportion of variance of the original data, indexed by their respective eigenvalues **(21)**.

Of particular relevance are the first two PCs, because they usually represent the bulk of variance in the data and can be easily plotted and explored interactively. An example of such explorative analysis is (22), where gene expression samples from different tissues have been projected onto the first two PCs and showed meaningful groupings associated with tissue-specific gene signatures.

A similar approach combined with hierarchical clustering was used in (23) to consistently identify four sub-groups of neuroblastoma which were highly correlated with prognostic factors across several datasets of gene expression profiles. As a matter of fact, three of the found clusters overlapped largely with known molecular sub-types, however, the fourth cluster was considerably different from the others, suggesting a potential unknown new sub-type. Finally, an inspection of the PC weights helped to identify a relevant set of 6 discriminant genes.

PCA can be used as a preprocessing step to reduce the dimensionality of the dataset, as in (24), where it was used to compress the dimensionality of several copy number profiles in cancer tissues in order to identify stable sub-populations.

However, it is not true in general that the information necessary to separate different experimental conditions gets captured among the first PCs. In (25) the authors showed with a series of synthetic and real data that clustering information can be encoded in non-leading PCs, which represent far less variance than the first ones. In some cases, to be able to extract the grouping information, the first and the last PCs were necessary. This effect was further studied in (26) and attributed mainly to the effect size of the signal of interest and the number of samples in which it is present. Small variations within a reduced set of genes are hardly represented in the

first PCs, whereas, different environmental conditions or tissue types that heavily influence hundreds of genes are easier to be captured in the leading PCs.

Non-negative Matrix Factorization (NMF)

NMF is a factorization algorithm that approximates the data matrix \mathbf{X} with the product of two lower rank matrices \mathbf{W} and \mathbf{H} , *i.e.*, $\mathbf{X} \approx \mathbf{H}\mathbf{W}$, where \mathbf{H} is $n \times r$ and \mathbf{W} is $r \times p$. The rank of the approximation r is chosen so that $(n + p)r < np$. Each row of \mathbf{X} is approximated as a linear combination of the rows of \mathbf{W} weighted by the corresponding row in \mathbf{H} . The non-negativity induces the matrix \mathbf{W} to learn a representation space whose components correspond to localized and interpretable features. In contrast to PCA where each feature contributes to all the components, NMF learns a parts-based decomposition of the data (27). This effect is due to the non-negativity constraint that allows only the additive composition of features.

Like PCA, NMF has been employed to reduce the dimensionality of the data and to discover sub-groups in several datasets. In (28) and (29) NMF was applied to a leukaemia dataset (30) and three other cancer datasets of gene expression data. The discovered sub-groups learned with hierarchical clustering over the representation learned with NMF were correlated to relevant phenotypes. In (31) also the copy number data of three cancer types has been analyzed with NMF where the groups discovered were highly associated with molecular sub-types with different prognosis. In the case of breast cancer, the sub-groups were correlated to the levels of the estrogen receptor, a relevant biomarker.

In both analyses, (28) and (29), it is highlighted that the features learned by NMF have the advantage of being easier to interpret due to their parts-based, compositional nature (27).

Compared to the holistic features learned from PCA, in NMF, each learned feature is influenced by just a subset of the original features. In the case of microarray gene expression data, each feature, represented as a row in the matrix W , can be considered as a “metagene” which summarizes the common pattern of a subset of genes (32). As a consequence, the values in the matrix H can be considered as the expression value of each sample for each metagene.

Despite its higher computational cost compared to PCA, it is acknowledged that NMF can be employed as a substitute for dimensionality reduction and visualization of microarray data. When used in conjunction with clustering algorithms for sub-class discovery the resulting group separations are consistently sharper than those obtained with PCA (32, 33).

Isometric Mapping

High dimensional gene expression data is regulated by complex non-linear interactions that can be missed by linear dimensionality reduction models like PCA and NMF (34). However, due to the regulatory patterns, the distribution of the data may not cover the entirety of the high-dimensional space but lie on a manifold (*i.e.* a surface embedded in higher dimensions) characterized by a lower number of variables compared to the number of actual dimensions. Thus learning the manifold structure can help in untangling the hidden factors of variation, while reducing the dimensionality of the data.

Moreover, a direct consequence of data distributed on a manifold is that computing the Euclidean distance between any two points may be a bad approximation of their geodesic

distance (*i.e.* the length of the path between the two points on the surface of the manifold), leading to a misinterpretation of similarity between data points.

Isometric Mapping (Isomap) **(34)** has been proposed as a non-linear method to learn an approximation of the geodesic distance between points on a manifold in order to embed the data in a linear low-dimensional space that can be explored visually and further analyzed.

The first step of Isomap is to learn a weighted neighborhood graph $G = (V, E, d)$ among the data points in the original space, where V is the set of nodes, E is the set of edges and d is a distance function among pairs of nodes. A weighted edge $e = (i, j)$ belongs to the graph G if x_i and x_j are neighbors; the weight of the edge corresponds to their distance in the high-dimensional space. Neighborhood relations can be determined in two ways: either connect one point to all other points within a fixed distance threshold ϵ_d , or to all its K nearest neighbors. In the second step, the geodesic distance between any pair of points is approximated with the length of its shortest path in graph G .

The third step is to apply classical multidimensional scaling (MDS) **(35)** to embed the data points in a Euclidean lower-dimensional space that preserves as much as possible the geodesic distances of the data points. MDS is an embedding methodology that arranges data points in a low dimensional space such that the distances between data points in the original space and the embedded space are as similar as possible. Isomap is theoretically guaranteed to converge to the real structure of the data, however, enough data has to be provided **(34)**.

An assessment of Isomap was performed in **(36)**, where the model was applied to two rat datasets consisting of a multiple tissue gene expression dataset and a collection of samples from a study on spinal cord injury and a high-throughput drug screening dataset. In all instances

Isomap performed remarkably well, highlighting hidden structures in the datasets. In the multiple tissue dataset, different regions of the low-dimensional space were populated with consistent clusters of samples originated from the same tissues, whereas in the spinal cord injury dataset the embedded samples shown strong relationships with experimental variables such as the distance of the sample from the location of the injury, the time after which the sample was collected and the severity of the injury. In the drug screening dataset, the authors were able to identify two directions in the embedded space which correlated with the differentiation patterns of leukaemia cells exposed to different chemicals.

Another benchmark was performed in (37) where five different cancer datasets were analysed. For all datasets, the authors report well-separated groups of samples corresponding to known categories and sub-groups in both two and three dimensions. They also evaluated the performance of clustering algorithms applied to the reduced datasets and compared the clustering with the known groupings, showing in every case an improvement with respect to performing the same clustering on the raw data.

Both studies also compared the embedding learned by Isomap with the one provided by PCA. In both analyses, Isomap provided better visualizations and clustering performances over PCA, however, there were some cases in which they provided comparable results (using a higher number of PCs, thus, diminishing the advantage of visual inspection).

Clustering

The problem of clustering is about assigning objects to groups, in a way that maximizes (minimizes) the similarities (distances) among the members of the clusters. This general

description leaves space for interpretation, and in fact different clustering algorithms and clustering evaluation metrics focus on different views of the problem, as there is, in general, no algorithm that performs significantly better on every problem instance.

There is a long history of applications of clustering to gene expression data (38–41). The most direct approaches consist in clustering the genes with respect to their expression in a set of high throughput samples, or, analogously, to cluster samples with respect to their expression in the considered genes. The first option is useful to understand which clusters of genes maintain a similar expression in different conditions, while the second allows to cluster together similar conditions, for example to identify sub-types of a disease, or drugs having a similar effect.

Many clustering algorithms have been introduced (42–45), they are extremely variegated and can be classified along a number of categorizations. One of these is the type of cluster membership. The output of the algorithm may be a partitioning of the input objects so that each object is assigned to exactly one cluster. This is also called hard-clustering. On the opposite when membership is not a sharp yes/no value we have soft-clustering, by assigning to each object a probability distribution of membership to the groups, or a fuzzy assignment. Among the hard-clustering algorithms that are most often applied to microarray data, we find k -means (46), Affinity propagation (47), Density-based clustering (48), Spectral clustering (49), and Hierarchical Clustering (50).

Mixture models instead, are soft-clustering techniques that assign to each object a probability distribution of belonging to the clusters (51–53). For a list of the main characteristics of these algorithms see (3). Hard-clustering is a special case of both probabilistic clustering and fuzzy

clustering, where the belonging of an object is to just one cluster with probability/degree of membership equal to 1.

Generally, it is better to avoid introducing unneeded complexity, hence, it is preferable to use a hard-clustering algorithm if it is enough to get the most likely cluster assignment. On the other hand, using probabilistic clustering is recommended when assignments that are not the most likely are of interest, or fuzzy clustering if the elements are expected to possess a degree of membership, perhaps to more than one cluster.

Based on the hypothesis and the experimental data at hand, some similarity measures between objects may be more suited than others to be used in the clustering. Some algorithms may work on any kind of similarity measures, and are thus more general, while others impose the use of proper distances, satisfying the property of triangle inequality, or even of a specific distance. The choice of measure is important, and the most effective measure depends on the data and the choice of clustering algorithm and evaluation metric (see next section) (54). In choosing the similarity measure, it is possible to leverage prior knowledge, if available. When in doubt between more than one choice, it is best to perform objective comparisons using an evaluation metric.

Choosing the number of clusters, k , is clearly important. The optimal number depends on the underlying structure of the data, but also on the user preferences. For example, in certain cases a high k fits the data effectively, but may not be practical if intended for visual inspection. Many clustering algorithms require the user to provide k (46, 49). Other clustering algorithms do not require k as input, but still k depends indirectly from user choices, like the definition of a similarity or a neighborhood relation (47, 48, 50).

The fuzzy paradigm is a generalization of set theory where the degree of belonging to a set is not crisp, *i.e.* true or false, but is fuzzy, *i.e.* a value between 0 and 1. For example type-2 diabetes is a disease with a continuous degree of seriousness and the degree to which a person is affected from diabetes can be seen as a fuzzy quantity. The fuzzy paradigm can be applied to clustering so that the belonging of an object to a specific cluster is a value between 0 and 1, and for each object a value is specified about its belonging to each of the clusters (55). Fuzzy clustering has been widely applied to microarray data (56, 57). The choice between hard-clustering and fuzzy clustering depends on the problem at hand. If the objects can be seen as having degrees of belonging near the borders of clusters, and/or the objects can be seen to belong possibly to more than one cluster, the use of fuzzy clustering is suggested (58).

In clustering microarray data it is common to face the presence of outliers, objects far from the rest of the samples that cannot be easily categorized in any of the clusters. They can be caused by measurement errors, but this cannot be taken for granted. Various options are available, like removing the outliers from the set of objects, substituting them with nearer objects, or assigning them a lower weight while running the clustering algorithm (59, 60).

Consensus Clustering

In order to improve the stability of an algorithm having a random component, *e.g.* the initial choice of centroids in k -means, consensus clustering can be used to aggregate the results from a number of runs. It can also be used in case an ensemble approach is desired (61–63). In ensemble algorithms different clustering algorithms are used then their results are aggregated to produce a final single partition of the data. Cluster ensembles are often motivated by the fact that different

data is structured in different ways, and different clustering algorithms are more suited to identify different kinds of structures. For example, k -means is particularly well suited to identify spherical structures. If the user does not possess prior information on the kind of structures that characterize the data, an ensemble approach may allow to reach a better result than a single algorithm because of the higher probability that a component of the ensemble may recognize the structure in the data (64). A drawback is that it must be taken into account that by applying more algorithms there is an increased chance that spurious structures, *i.e.* structures produced by noise, are mistaken as real structures. Consensus clustering has found many applications in analyzing gene expression data, like in cancer (65), toxicogenomics (66), or identifying gene subsets (67). Specific implementations designed for applying consensus clustering to microarray data exist, like clusterCons (68) that is written in the R language, and the web-application ArrayMining (69).

k -means is probably the most used clustering algorithm, but it has been noticed that it can also be used as a basis to compute a consensus between clustering results, so a consensus approach based on k -means has been proposed, called k -means-based consensus clustering (KCC). Wu et al. (70) provides a unified framework for KCC and suggestions for real-world practical applications.

Among consensus clustering methods an important family is the one of the co-association matrix-based methods, where the information from the single partitions is summarized in an adjacency matrix with weights that are proportional to the number of times two elements are found together in the single partitions. Using the co-association matrix it is possible to express

the consensus problem as a community detection problem on a weighted graph (71). While brute force approaches starting from the co-association matrix are generally intractable, there are a number of approximate methods. Liu et al. (72) propose a spectral clustering approach starting from the co-expression matrix.

In many applications of microarray data, there is no clear boundary between types of readings, being them from patients, drugs, or toxic substances, and fuzzy clustering is used to represent these types with soft boundaries. As in the case of crisp clustering, ensemble methods may be applied also to compute a consensus from fuzzy clusterings. One such methodology is presented by Avogadri et al. (73), where first multiple random projections on a lower dimensional space of the gene expression data are created, then fuzzy clustering is applied to each of them. Afterwards, a consensus is computed on the fuzzy clusterings to obtain a single final result. They compared this approach with other algorithms on four microarray datasets. While the comparison cannot be considered conclusive, the experiments show that ensemble fuzzy clustering can produce good results when applied to microarray data.

Subspace clustering

Even though with microarrays it is possible to measure thousands of variables at the same time for different experimental conditions, it is likely that only a small subset of the genes are relevant, and considering all available genes (*i.e.* the whole data space) could lead to noisy or misleading results. Therefore, clustering the samples only on a subset of relevant genes (*i.e.* a subspace) can lead to a more accurate clustering with respect to the posed task.

There are two main groups of subspace clustering algorithms: i) top-down and ii) bottom-up. Subspace clustering algorithms following a top-down approach commonly start by estimating clusters in the whole data space and then iteratively evaluate every subspace of the identified clusters (74, 75). Some examples for top-down subspace clustering algorithms are ORCLUS (76) and FINDIT (77). On the other hand, bottom-up subspace clustering algorithms first find clusters in a low dimensional space and then iteratively merge them into higher dimensional clusters. Some examples of algorithms following a bottom-up subspace clustering approach are CLIQUE (78), ISC (75) and CLTree (79). CLIQUE tries to find dense regions, by dividing each dimension into a “search grid” where a dense area is defined as a region where there are more data points contained than a set threshold. Dimensions are then merged and dense regions are identified based on the same principle. The mentioned concepts and algorithms can be grouped together as what has been described as hard subspace clustering algorithms (80), which aim at characterizing exact subspaces.

Parsons et al. (74) suggest that in high dimensional data bottom-up algorithms should perform better with respect to computational time, while top-down algorithms should scale well to very large datasets. The authors also performed a direct comparison between MAFIA (81), a bottom-up algorithm and FINDIT (77), a top-down algorithm. On synthetic data they showed that for certain numbers of instances, MAFIA tended to identify all clusters but often was not able to determine all significant dimensions. On the other hand, FINDIT seemed to be more accurate in the dimensions it identified for determined clusters but struggled to identify all clusters, especially with higher instance size (74).

In contrast to the hard subspace clustering methodology there is a more recently described group of algorithms - known in literature as soft-subspace clustering, which tries to identify how much a dimension contributes to a particular cluster (80). Examples of soft-subspace clustering algorithms are WFCM (82) and SCAD (83). For further reading, a detailed classification of soft-subspace clustering algorithms in sub-categories has been published by Deng et al. (80). The authors further suggest that soft-subspace algorithms have achieved good performance, especially for high dimensional data.

Evaluation metrics

Various validation methods can be used to assess clustering quality and can help in choosing between algorithms and their parameters, such as the number of clusters. We can identify four kinds of evaluation: manual, internal, and external.

In manual evaluation, the clustering is validated by human domain experts.

The methods for internal validation assign a score to the quality of the clustering that is based solely on the clustering itself and the intrinsic properties of data on which the clustering was performed. These measures do not need any external data, but even with high scores, it is not guaranteed that the obtained grouping is correlated to any phenotypic variable. Examples of internal measures are the Davies-Bouldin index, the Dunn index, and the silhouette index (84, 85).

External quality measures rely on an external, expected grouping and on a similarity measure between clusterings to measure the concordance between the output clustering and the optimal

one. Approaches to this problem have a long history (86, 87), and examples of similarity measures used for external evaluation are the Rand index, the Jaccard index, the Normalized Mutual Information, and the *F*-measure (88–90).

Different clustering algorithms can produce different results starting from the same input data, and even a single algorithm may produce two different results in two different runs if featured with a random component. The same similarity measures used for external evaluation may be applied to two arbitrary clusterings to quantify their similarity.

Blindly applying numerous clustering algorithms and then subjectively choosing one result based on expectations is a dangerous practice that can lead to cherry picking (91). In order to test more than one clustering algorithm and/or combination of parameters, it is possible to use an objective validation measure to choose which solution to use. The suggested procedure consists in a) identifying one or more clustering algorithms considered particularly well suited for the specific domain of application, based on a priori knowledge, in an analogous way b) identifying the set up parameters, like *e.g.* the number of clusters, and c) a validation measure, d) running the clustering algorithms, and finally e) selecting the best clustering with respect to the scores of the validation measure.

Liu et al. (92) compared 11 internal validation measures using 5 different kinds of synthetically generated datasets, and found that the *S_Dbw* validity index was the only one to perform well in all cases. The *S_Dbw* index is based on the concepts of cluster compactness (intra-cluster low variance) and density between clusters (inter-cluster density) (87). Wiwie et al. (93) compared 13 clustering methods on 24 biological datasets. Based on external evaluation with the *FI*-score, the

best performers were transitivity clustering, hierarchical clustering, Clusterdp, and partitioning around medoids. They found that internal and external quality measures do not correlate well in general, and the best correlations were between the silhouette internal score and the *F1*-score, *F2*-score, *FM*-index, Jaccard index, and *V*-measure external scores. This suggests that the silhouette score might in most cases be the best internal score to use for choosing the method and the model parameters when applying clustering to biological datasets. When using the silhouette score to choose the parameters, the best models were transitivity clustering, hierarchical clustering, and partitioning around medoids.

Biclustering

Biclustering has become a common technique for the study of omics data to identify molecular features with similar alteration patterns in different experimental conditions (94). Biclustering methodologies aim at identifying sub-groups of measurements that have a similar profile in a subset of the samples. The main assumption of the biclustering methodology is that genes are not necessarily related to every sample, but they might be important only for a subset of them. Moreover, one gene can be assigned to a cluster under certain samples and in another cluster for different samples, depending on their involvement in different biological processes. Thus, it is important to simultaneously group both the samples and features (*e.g.*, genes). This problem has been shown to belong to the class of *NP*-hard problems (95) and, consequently, proposed approaches for biclustering are based on optimization procedures based on heuristic search algorithms.

Biclustering has been studied for a long time and proposed solutions for this problem cover almost all of the spectra of combinatorial optimization algorithm families (94). Biclustering approaches can be mainly divided into two categories based on the usage of an evaluation measure guiding the construction of biclusters as proposed in (94).

Approaches belonging to the first class make use of one or more functions evaluating the quality of the solution being built (represented by a possible set of biclusters) and driving the construction process. They mainly differ on the strategy used to build and improve the solution and are represented by greedy search approaches, stochastic iterative greedy search, nature-inspired meta-heuristics and clustering-based approaches. A list of algorithms implementing these strategies is provided in Table 3.

Greedy search algorithms work by selecting a local optimum solution at each step in order to move closer to a globally optimal solution (124). These algorithms generally work by constructing the biclusters starting from the small constituent parts (genes/samples). They do not guarantee global optimum solutions but provide good approximations in a reasonable time.

Nature-inspired meta-heuristics resembling schemas of efficient behaviors derived from the natural world have been shown to have good performances in addressing biclustering optimization problems (109–119). They are characterized by the use of operators that simulate natural phenomena (like natural selection) to improve the solution.

Clustering-based approaches are based on the usage of a standard one-way clustering approach in addition to a strategy to derive the desired 2-dimensional clustering (121–123).

The other class of approaches for biclustering is constituted by algorithms where the construction of the solution is not directly driven by an evaluation measure. This class includes graph-based approaches, one-way clustering-based approaches, probabilistic models, linear algebra, and optimal reordering of rows and columns (*94*).

Biclustering is useful whenever there is a biological reason for expecting sub-structures of genes or samples in the data. Cancer is a classical example where biclustering algorithms applied to microarray experiments succeeded at identifying several cancer sub-types (*125, 126*) that one way-clustering couldn't detect (*127*).

The choice of the particular approach to be used usually depends on the problem being addressed and/or on the trade-off between the available computational time/resources and the quality of the solution.

The quality of a bicluster can be measured by means of technical and/or biological metrics. Technical metrics are used by many algorithms to drive their search, such as the variance of the bicluster elements and the Mean Squared Residue (MSR) that measure the coherence of the genes and samples in each bicluster. The size of the bicluster is also important, with bigger biclusters usually being preferred. Biological metrics are tightly linked to the problem of bicluster interpretation: usually, a good bicluster is one where genes do resemble a particular molecular function (*e.g.* some KEGG pathway is enriched for them) and/or samples do constitute a biologically meaningful group (*e.g.* a distinct cancer sub-type). When doing this, it is important to take into account that not all genes/samples should always fall in a particular bicluster, therefore it must not be expected for all biclusters to be meaningful objects.

Multi-omics clustering

Biological systems are regulated by a number of interrelated, but different, regulatory layers. Genomic, transcriptomic, epigenomic, proteomic and metabolomic are a non-exhaustive list of regulatory entities governing molecular processes (*128*).

High-throughput technologies, including DNA microarrays, allow a snapshot of such molecular layers (also called “omes”) in one or more biological conditions to be obtained. The integrated analysis of such layers allows a thorough characterization of the system under study, in respect with the separate analysis of the single layers. In fact, by combining different omics, it is possible to pinpoint not only relevant information in single layers, but also complementary and integrated inter-omics relationships (*128*). This strategy empowers enormously the ability of the researchers to extract knowledge about the biological condition of interest. Such methodological advancements give the possibility to achieve a wide landscape of the molecular profiles, even from a single sample (*7*). In fact, longitudinal studies, often led by big consortia, such as The Cancer Genome Atlas (TCGA), International Cancer Genome Consortium (ICGC), Therapeutically Applicable Research to Generate Effective Treatments (TARGET) among others, aimed at profiling the omes in order to identify molecular determinants of certain phenotypic traits. While multi-omics approaches offer an unprecedented opportunity to acquire considerable knowledge about complex biological processes, they also increase the complexity of the analytical methods. In fact, in order to maximize the knowledge achievable from each of the considered omics, integrative analyses of multiple omics layers is required. Indeed, the analysis of multiple datasets produced by employing different platforms and experimental

procedures is a challenging task, since systematic biases exist due to technological platforms, laboratories and analysis methods (7). Multi-omics integrated approaches combine individual molecular layers in order to understand the interplay among molecules (129). They help in assessing the flow of information from one omics level to the other and thus help in bridging the gap from genotype to phenotype.

Clustering multi-omics data techniques primarily differ on the adopted strategy to integrate data from different layers. These strategies are adopted to solve the problem of integrating different types of information about the same sample and/or the same feature characterized by different subjects, different representations, different cardinality and different relationships.

Integration strategies can be broadly categorized in three groups (130): i) early, ii) intermediate and iii) late integration strategies.

In early integration strategies multiple data types from the different omics are fused together by concatenating data matrices into a single comprehensive matrix on which clustering is performed. The advantage of this approach is the relative ease of applying statistical methods to any final data matrix. However, this method does suffer from problems related to the differences in scaling of the blocks of concatenated data, the inherent biases of each data type, the fact that it ignores the different distributions of values in different omics and, finally, from the fact that the integrated dataset might have too many (often dependent) variables to be used in the clustering phase. Normalization techniques can be used to ensure that data of different orders of magnitude are scaled to be in the same range. Also, data dimensionality can be reduced by using supervised/unsupervised feature selection and reduction. Among the tools for early integration of different omics we have TW- k -means. It is a variant of the k -means clustering, which uses a

double weighting scheme to assess the relevance of each single feature as well as the relevance of a whole data layer *(131)*.

In intermediate integration strategies each dataset is first converted into a common intermediate form (*e.g.* a “graph” or a “kernel matrix”), then integration is performed at the level of transformed data and clustering is finally performed onto the integrated model. This approach has the advantage of preserving and gives the possibility to virtually combine any data structure applying the appropriate transformations. The main challenge of intermediate integration resides in the difficulty to interpret the relationship between the original features and obtained clusters. A widely used implementation of this approach is Similarity Network Fusion (SNF) *(6)* and its derivatives such as Affinity Network Fusion (ANF) *(132)* and Integrative Network Fusion (INF) *(133)*.

In late integration strategies separate clusters are generated for each omics layer and successively integrated to produce a final data-driven clustering. The advantage is that with this approach it is possible to combine models coming from various groups of samples for which different data sources have been analyzed. As a downside, this approach can miss important interactions existing among different data types. MVDA *(134)* is an example of late integration that has been successfully applied in identifying disease sub-groups.

Although more and more data integration algorithms yielding better performances are being developed, they have to still face some unsolved challenges. Often, the main reason is that data integration approaches were designed to deal with single data layers, and only later they have been adapted to work with multiple types of data. For instance, network-based methods may fail to derive information about the connectivity among different networks representing different

data layers, and thus, failing to capture their biological significance (*135*). Further improvements are still needed to overcome this and many other challenges posed by data integration.

Conclusions

Unsupervised learning is a challenging task because, for the most part, both practitioners and algorithms cannot exploit any collateral information other than the dataset itself. In the case of (sub-)group discovery, where no predefined class labeling is available, additional care must be taken to discard results due to noise from the relevant patterns hidden in the data. On top of that, the peculiar characteristics of microarray data, namely, the high number of variables compared to the number of samples, their complex regulatory interactions, and the mixture of biological and technical noise signals, pose remarkable challenges to the development of reliable analytic models.

In this chapter we explored the basic building blocks of an unsupervised analysis focused on group discovery. In particular, we described different measures to evaluate the degree of similarity of pairs of samples, methodologies to reduce the effects of the curse of dimensionality and compressing the data to a handful of informative features (down to 2 or 3 dimensions for visual exploration). Finally, we showed several methodologies for discovering groups and different criteria to evaluate the findings both numerically and based on domain knowledge. Moreover, we also mentioned the case when multiple -omics measurements are available for the same samples, the challenges this setting poses and some successful approaches.

Where possible, we tried to focus our discussions on relative advantages and disadvantages of each methodology as well as reporting comparisons, results and case studies from the literature,

hoping that this will help practitioners to make more informed decisions even though there are no hard rules about when to choose an approach over another.

References

1. Kinaret PAS, Serra A, Federico A, et al (2020) Transcriptomics in Toxicogenomics, Part I: Experimental Design, Technologies, Publicly Available Data, and Regulatory Aspects. *Nanomater Basel Switz* 10:. <https://doi.org/10.3390/nano10040750>
2. Federico A, Serra A, Ha MK, et al (2020) Transcriptomics in Toxicogenomics, Part II: Preprocessing and Differential Expression Analysis for High Quality Data. *Nanomater Basel Switz* 10:. <https://doi.org/10.3390/nano10050903>
3. Serra A, Fratello M, Cattelani L, et al (2020) Transcriptomics in Toxicogenomics, Part III: Data Modelling for Risk Assessment. *Nanomater Basel Switz* 10:. <https://doi.org/10.3390/nano10040708>
4. Sun J, Bi J, Kranzler HR (2014) Multi-view singular value decomposition for disease subtyping and genetic associations. *BMC Genet* 15:73. <https://doi.org/10.1186/1471-2156-15-73>
5. Antonarakis SE, Chakravarti A, Cohen JC, Hardy J (2010) Mendelian disorders and multifactorial traits: the big divide or one for all? *Nat Rev Genet* 11:380–384. <https://doi.org/10.1038/nrg2793>
6. Wang B, Mezlini AM, Demir F, et al (2014) Similarity network fusion for aggregating data types on a genomic scale. *Nat Methods* 11:333–337. <https://doi.org/10.1038/nmeth.2810>
7. Casamassimi A, Federico A, Rienzo M, et al (2017) Transcriptome Profiling in Human Diseases: New Advances and Perspectives. *Int J Mol Sci* 18:. <https://doi.org/10.3390/ijms18081652>
8. Koboldt DC, Fulton RS, McLellan MD, et al (2012) Comprehensive molecular portraits of human breast tumours. *Nature* 490:61–70. <https://doi.org/10.1038/nature11412>
9. Garside H, Marcoe KF, Chesnut-Speelman J, et al (2014) Evaluation of the use of imaging parameters for the detection of compound-induced hepatotoxicity in 384-well cultures of HepG2 cells and cryopreserved primary human hepatocytes. *Toxicol Vitro Int J Publ Assoc BIBRA* 28:171–181. <https://doi.org/10.1016/j.tiv.2013.10.015>
10. Martin TM, Lilavois CR, Barron MG (2017) Prediction of pesticide acute toxicity using two-dimensional chemical descriptors and target species classification. *SAR QSAR Environ Res* 28:525–539. <https://doi.org/10.1080/1062936X.2017.1343204>
11. Serra A, Greco D, Tagliaferri R (2015) Impact of different metrics on multi-view clustering. In: 2015 International Joint Conference on Neural Networks (IJCNN). pp 1–8
12. Shirchorshidi AS, Aghabozorgi S, Wah TY (2015) A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data. *PLOS ONE* 10:e0144059. <https://doi.org/10.1371/journal.pone.0144059>
13. Fan J, Liao Y, Liu H (2016) An overview of the estimation of large covariance and precision matrices. *Econom J* 19:C1–C32. <https://doi.org/10.1111/ectj.12061>

14. Chen S, Ma B, Zhang K (2009) On the similarity metric and the distance metric. *Theor Comput Sci* 410:2365–2376. <https://doi.org/10.1016/j.tcs.2009.02.023>
15. Chen S, Ma B, Zhang K (2007) The Normalized Similarity Metric and Its Applications. In: 2007 IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2007). pp 172–180
16. Ontañón S (2020) An overview of distance and similarity functions for structured data. *Artif Intell Rev* 53:5309–5351. <https://doi.org/10.1007/s10462-020-09821-w>
17. Jaskowiak PA, Campello RJ, Costa IG (2014) On the selection of appropriate distances for gene expression data clustering. *BMC Bioinformatics* 15:S2. <https://doi.org/10.1186/1471-2105-15-S2-S2>
18. Shapiro SS, Wilk MB (1965) An Analysis of Variance Test for Normality (Complete Samples). *Biometrika* 52:591–611. <https://doi.org/10.2307/2333709>
19. D’agostino RB, Belanger A, Jr RBD (1990) A Suggestion for Using Powerful and Informative Tests of Normality. *Am Stat* 44:316–321. <https://doi.org/10.1080/00031305.1990.10475751>
20. Araújo D, Neto AD, Martins A, Melo J (2011) Comparative study on dimension reduction techniques for cluster analysis of microarray data. In: The 2011 International Joint Conference on Neural Networks. pp 1835–1842
21. Bishop CM (2006) *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg
22. Misra J, Schmitt W, Hwang D, et al (2002) Interactive Exploration of Microarray Gene Expression Patterns in a Reduced Dimensional Space. *Genome Res* 12:1112–1120. <https://doi.org/10.1101/gr.225302>
23. Abel F, Dalevi D, Nethander M, et al (2011) A 6-gene signature identifies four molecular subgroups of neuroblastoma. *Cancer Cell Int* 11:9. <https://doi.org/10.1186/1475-2867-11-9>
24. Brito I, Hupé P, Neuvial P, Barillot E (2013) Stability-Based Comparison of Class Discovery Methods for DNA Copy Number Profiles. *PLOS ONE* 8:e81458. <https://doi.org/10.1371/journal.pone.0081458>
25. Yeung KY, Ruzzo WL (2001) Principal component analysis for clustering gene expression data. *Bioinformatics* 17:763–774. <https://doi.org/10.1093/bioinformatics/17.9.763>
26. Lenz M, Müller F-J, Zenke M, Schuppert A (2016) Principal components analysis and the reported low intrinsic dimensionality of gene expression microarray data. *Sci Rep* 6:25696. <https://doi.org/10.1038/srep25696>
27. Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. *Nature* 401:788–791. <https://doi.org/10.1038/44565>
28. Brunet J-P, Tamayo P, Golub TR, Mesirov JP (2004) Metagenes and molecular pattern discovery using matrix factorization. *Proc Natl Acad Sci* 101:4164–4169. <https://doi.org/10.1073/pnas.0308531101>
29. Devarajan K, Ebrahimi N (2008) Class Discovery via Nonnegative Matrix Factorization. *Am J Math Manag Sci* 28:457–467. <https://doi.org/10.1080/01966324.2008.10737738>
30. Slonim DK, Tamayo P, Mesirov JP, et al (2000) Class prediction and discovery using gene expression data. In: Proceedings of the fourth annual international conference on Computational molecular biology. Association for Computing Machinery, New York, NY, USA, pp 263–272
31. Campos CP de, Rancoita PMV, Kwee I, et al (2013) Discovering Subgroups of Patients from

- DNA Copy Number Data Using NMF on Compacted Matrices. PLOS ONE 8:e79720. <https://doi.org/10.1371/journal.pone.0079720>
32. Devarajan K (2008) Nonnegative Matrix Factorization: An Analytical and Interpretive Tool in Computational Biology. PLOS Comput Biol 4:e1000029. <https://doi.org/10.1371/journal.pcbi.1000029>
 33. Liu W, Yuan K, Ye D (2008) Reducing microarray data via nonnegative matrix factorization for visualization and clustering analysis. J Biomed Inform 41:602–606. <https://doi.org/10.1016/j.jbi.2007.12.003>
 34. Tenenbaum JB, Silva V de, Langford JC (2000) A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science 290:2319–2323. <https://doi.org/10.1126/science.290.5500.2319>
 35. Cox MAA, Cox TF (2008) Multidimensional Scaling. In: Chen C, Härdle W, Unwin A (eds) Handbook of Data Visualization. Springer, Berlin, Heidelberg, pp 315–347
 36. Dawson K, Rodriguez RL, Malyj W (2005) Sample phenotype clusters in high-density oligonucleotide microarray data sets are revealed using Isomap, a nonlinear algorithm. BMC Bioinformatics 6:195. <https://doi.org/10.1186/1471-2105-6-195>
 37. Shi J, Luo Z (2010) Nonlinear dimensionality reduction of gene expression data for visualization and clustering analysis of cancer tissue samples. Comput Biol Med 40:723–732. <https://doi.org/10.1016/j.compbiomed.2010.06.007>
 38. Ben-Dor A, Shamir R, Yakhini Z (1999) Clustering Gene Expression Patterns. J Comput Biol 6:281–297. <https://doi.org/10.1089/106652799318274>
 39. Kerr G, Ruskin HJ, Crane M, Doolan P (2008) Techniques for clustering gene expression data. Comput Biol Med 38:283–293. <https://doi.org/10.1016/j.compbiomed.2007.11.001>
 40. Andreopoulos B, An A, Wang X, Schroeder M (2009) A roadmap of clustering algorithms: finding a match for a biomedical application. Brief Bioinform 10:297–314. <https://doi.org/10.1093/bib/bbn058>
 41. Pirim H, Ekşioğlu B, Perkins AD, Yüceer Ç (2012) Clustering of high throughput gene expression data. Comput Oper Res 39:3046–3061. <https://doi.org/10.1016/j.cor.2012.03.008>
 42. Jain AK (2010) Data clustering: 50 years beyond K-means. Pattern Recognit Lett 31:651–666. <https://doi.org/10.1016/j.patrec.2009.09.011>
 43. Xu D, Tian Y (2015) A Comprehensive Survey of Clustering Algorithms. Ann Data Sci 2:165–193. <https://doi.org/10.1007/s40745-015-0040-1>
 44. Saxena A, Prasad M, Gupta A, et al (2017) A review of clustering techniques and developments. Neurocomputing 267:664–681. <https://doi.org/10.1016/j.neucom.2017.06.053>
 45. Serra A, Tagliaferri R (2019) Unsupervised Learning: Clustering. In: Encyclopedia of Bioinformatics and Computational Biology
 46. Celebi ME, Kingravi HA, Vela PA (2013) A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Syst Appl 40:200–210. <https://doi.org/10.1016/j.eswa.2012.07.021>
 47. Frey BJ, Dueck D (2007) Clustering by Passing Messages Between Data Points. Science 315:972–976. <https://doi.org/10.1126/science.1136800>
 48. Kriegel H-P, Kröger P, Sander J, Zimek A (2011) Density-based clustering. WIREs Data Min Knowl Discov 1:231–240. <https://doi.org/10.1002/widm.30>
 49. von Luxburg U (2007) A tutorial on spectral clustering. Stat Comput 17:395–416. <https://doi.org/10.1007/s11222-007-9033-z>

50. Hasan MN, Malek MB, Begum AA, et al (2019) Assessment of Drugs Toxicity and Associated Biomarker Genes Using Hierarchical Clustering. *Medicina (Mex)* 55:451. <https://doi.org/10.3390/medicina55080451>
51. Melnykov V, Maitra R (2010) Finite mixture models and model-based clustering. *Stat Surv* 4:80–116. <https://doi.org/10.1214/09-SS053>
52. McNicholas PD, Murphy TB (2010) Model-based clustering of microarray expression data via latent Gaussian mixture models. *Bioinformatics* 26:2705–2712. <https://doi.org/10.1093/bioinformatics/btq498>
53. Bouveyron C, Brunet-Saumard C (2014) Model-based clustering of high-dimensional data: A review. *Comput Stat Data Anal* 71:52–78. <https://doi.org/10.1016/j.csda.2012.12.008>
54. Kumar V, Chhabra JK, Kumar D (2014) Performance Evaluation of Distance Metrics in the Clustering Algorithms. *INFOCOMP J Comput Sci* 13:38–52
55. Ali AM, Karmakar GC, Dooley LS (2008) Review on Fuzzy Clustering Algorithms. *J Adv Comput* 2:169–181
56. Dombéle D, Kastner P (2003) Fuzzy C-means method for clustering microarray data. *Bioinformatics* 19:973–980. <https://doi.org/10.1093/bioinformatics/btg119>
57. Gasparoviča-Asīte M, Aleksejeva L, Nazaruks V (2013) Using Fuzzy Clustering with Bioinformatics Data. In: publication.editionName. pp 62–70
58. Bora DJ, Gupta DAK (2014) A Comparative study Between Fuzzy Clustering Algorithm and Hard Clustering Algorithm. *Int J Comput Trends Technol* 10:108–113. <https://doi.org/10.14445/22312803/IJCTT-V10P119>
59. Aggarwal CC (2015) Outlier Analysis. In: Aggarwal CC (ed) *Data Mining: The Textbook*. Springer International Publishing, Cham, pp 237–263
60. Campos GO, Zimek A, Sander J, et al (2016) On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Min Knowl Discov* 30:891–927. <https://doi.org/10.1007/s10618-015-0444-8>
61. Swift S, Tucker A, Vinciotti V, et al (2004) Consensus clustering and functional interpretation of gene-expression data. *Genome Biol* 5:R94. <https://doi.org/10.1186/gb-2004-5-11-r94>
62. Vega-Pons S, Ruiz-Shulcloper J (2011) A survey of clustering ensemble algorithms. *Int J Pattern Recognit Artif Intell* 25:337–372. <https://doi.org/10.1142/S0218001411008683>
63. Ghosh J, Acharya A (2011) Cluster ensembles. *WIREs Data Min Knowl Discov* 1:305–315. <https://doi.org/10.1002/widm.32>
64. Boongoen T, Iam-On N (2018) Cluster ensembles: A survey of approaches with recent extensions and applications. *Comput Sci Rev* 28:1–25. <https://doi.org/10.1016/j.cosrev.2018.01.003>
65. Brannon AR, Reddy A, Seiler M, et al (2010) Molecular Stratification of Clear Cell Renal Cell Carcinoma by Consensus Clustering Reveals Distinct Subtypes and Survival Patterns. *Genes Cancer* 1:152–163. <https://doi.org/10.1177/1947601909359929>
66. Gao C, Weisman D, Gou N, et al (2012) Analyzing High Dimensional Toxicogenomic Data Using Consensus Clustering. *Environ Sci Technol* 46:8413–8421. <https://doi.org/10.1021/es3000454>
67. Nguyen TT, Nowakowski RS, Androulakis IP (2009) Unsupervised Selection of Highly Coexpressed and Noncoexpressed Genes Using a Consensus Clustering Approach. *OMICS J Integr Biol* 13:219–237. <https://doi.org/10.1089/omi.2008.0074>

68. Simpson TI, Armstrong JD, Jarman AP (2010) Merged consensus clustering to assess and improve class discovery with microarray data. *BMC Bioinformatics* 11:590. <https://doi.org/10.1186/1471-2105-11-590>
69. Glaab E, Garibaldi JM, Krasnogor N (2009) ArrayMining: a modular web-application for microarray analysis combining ensemble and consensus methods with cross-study normalization. *BMC Bioinformatics* 10:358. <https://doi.org/10.1186/1471-2105-10-358>
70. Wu J, Liu H, Xiong H, et al (2015) K-Means-Based Consensus Clustering: A Unified View. *IEEE Trans Knowl Data Eng* 27:155–169. <https://doi.org/10.1109/TKDE.2014.2316512>
71. Fortunato S (2010) Community detection in graphs. *Phys Rep* 486:75–174. <https://doi.org/10.1016/j.physrep.2009.11.002>
72. Liu H, Liu T, Wu J, et al (2015) Spectral Ensemble Clustering. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, New York, NY, USA, pp 715–724
73. Avogadri R, Valentini G (2009) Fuzzy ensemble clustering based on random projections for DNA microarray data analysis. *Artif Intell Med* 45:173–183. <https://doi.org/10.1016/j.artmed.2008.07.014>
74. Parsons L, Haque E, Liu H (2004) Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explor Newsl* 6:90–105. <https://doi.org/10.1145/1007730.1007731>
75. Lakshmi BJ, Shashi M, Madhuri KB (2020) A rough set based subspace clustering technique for high dimensional data. *J King Saud Univ - Comput Inf Sci* 32:329–334. <https://doi.org/10.1016/j.jksuci.2017.09.003>
76. Aggarwal CC, Yu PS (2000) Finding generalized projected clusters in high dimensional spaces. In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. Association for Computing Machinery, New York, NY, USA, pp 70–81
77. Woo K-G, Lee J-H, Kim M-H, Lee Y-J (2004) FINDIT: a fast and intelligent subspace clustering algorithm using dimension voting. *Inf Softw Technol* 46:255–271. <https://doi.org/10.1016/j.infsof.2003.07.003>
78. Agrawal R, Gehrke J, Gunopulos D, Raghavan P (1998) Automatic subspace clustering of high dimensional data for data mining applications. In: *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*. Association for Computing Machinery, New York, NY, USA, pp 94–105
79. Liu B, Xia Y, Yu PS (2000) Clustering through decision tree construction. In: *Proceedings of the ninth international conference on Information and knowledge management*. Association for Computing Machinery, New York, NY, USA, pp 20–29
80. Deng Z, Choi K-S, Jiang Y, et al (2016) A survey on soft subspace clustering. *Inf Sci* 348:84–106. <https://doi.org/10.1016/j.ins.2016.01.101>
81. Goil S, Nagesh H, Choudhary A (1999) MAFIA: Efficient and Scalable Subspace Clustering for Very Large Data Sets
82. Wang X, Wang Y, Wang L (2004) Improving fuzzy c-means clustering based on feature-weight learning. *Pattern Recognit Lett* 25:1123–1132. <https://doi.org/10.1016/j.patrec.2004.03.008>
83. Frigui H, Nasraoui O (2004) Unsupervised learning of prototypes and attribute weights. *Pattern Recognit* 37:567–581. <https://doi.org/10.1016/j.patcog.2003.08.002>
84. Arbelaitz O, Gurrutxaga I, Muguerza J, et al (2013) An extensive comparative study of

- cluster validity indices. *Pattern Recognit* 46:243–256.
<https://doi.org/10.1016/j.patcog.2012.07.021>
85. Bolshakova N, Azuaje F (2003) Cluster validation techniques for genome expression data. *Signal Process* 83:825–833. [https://doi.org/10.1016/S0165-1684\(02\)00475-9](https://doi.org/10.1016/S0165-1684(02)00475-9)
 86. Hubert L, Arabie P (1985) Comparing partitions. *J Classif* 2:193–218.
<https://doi.org/10.1007/BF01908075>
 87. Halkidi M, Batistakis Y, Vazirgiannis M (2001) On Clustering Validation Techniques. *J Intell Inf Syst* 17:107–145. <https://doi.org/10.1023/A:1012801612483>
 88. Pfitzner D, Leibbrandt R, Powers D (2008) Characterization and evaluation of similarity measures for pairs of clusterings. *Knowl Inf Syst* 19:361.
<https://doi.org/10.1007/s10115-008-0150-6>
 89. Rodriguez MZ, Comin CH, Casanova D, et al (2019) Clustering algorithms: A comparative approach. *PLOS ONE* 14:e0210236. <https://doi.org/10.1371/journal.pone.0210236>
 90. Vinh NX, Epps J, Bailey J (2010) Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *J Mach Learn Res* 11:2837–2854
 91. Handl J, Knowles J, Kell DB (2005) Computational cluster validation in post-genomic data analysis. *Bioinformatics* 21:3201–3212. <https://doi.org/10.1093/bioinformatics/bti517>
 92. Liu Y, Li Z, Xiong H, et al (2010) Understanding of Internal Clustering Validation Measures. In: 2010 IEEE International Conference on Data Mining. pp 911–916
 93. Wiwie C, Baumbach J, Röttger R (2015) Comparing the performance of biomedical clustering methods. *Nat Methods* 12:1033–1038. <https://doi.org/10.1038/nmeth.3583>
 94. Pontes B, Giráldez R, Aguilar-Ruiz JS (2015) Biclustering on expression data: A review. *J Biomed Inform* 57:163–180. <https://doi.org/10.1016/j.jbi.2015.06.028>
 95. Tanay A, Sharan R, Shamir R (2002) Discovering statistically significant biclusters in gene expression data. *Bioinformatics* 18:S136–S144.
https://doi.org/10.1093/bioinformatics/18.suppl_1.S136
 96. Hartigan JA (1972) Direct Clustering of a Data Matrix. *J Am Stat Assoc* 67:123–129.
<https://doi.org/10.1080/01621459.1972.10481214>
 97. Cheng Y, Church GM (2000) Biclustering of expression data. *Proc Int Conf Intell Syst Mol Biol* 8:93–103
 98. Mukhopadhyay A, Maulik U, Bandyopadhyay S (2009) A novel coherence measure for discovering scaling biclusters from gene expression data. *J Bioinform Comput Biol* 07:853–868. <https://doi.org/10.1142/S0219720009004370>
 99. Yip KY, Cheung DW, Ng MK (2004) HARP: a practical projected clustering algorithm. *IEEE Trans Knowl Data Eng* 16:1387–1397. <https://doi.org/10.1109/TKDE.2004.74>
 100. Liu X, Wang L (2007) Computing the maximum similarity bi-clusters of gene expression data. *Bioinformatics* 23:50–56. <https://doi.org/10.1093/bioinformatics/btl560>
 101. Chen L-C, Yu PS, Tseng VS (2011) WF-MSB: A weighted fuzzy-based biclustering method for gene expression data. *Int J Data Min Bioinforma* 5:89–109.
<https://doi.org/10.1504/IJDMB.2011.038579>
 102. Teng L, Chan L (2008) Discovering Biclusters by Iteratively Sorting with Weighted Correlation Coefficient in Gene Expression Data. *J Signal Process Syst* 50:267–280.
<https://doi.org/10.1007/s11265-007-0121-2>
 103. Yun T, Yi G-S (2013) Biclustering for the comprehensive search of correlated gene

- expression patterns using clustered seed expansion. *BMC Genomics* 14:144.
<https://doi.org/10.1186/1471-2164-14-144>
104. Ahmed HA, Mahanta P, Bhattacharyya DK, Kalita JK (2014) Shifting-and-Scaling Correlation Based Biclustering Algorithm. *IEEE/ACM Trans Comput Biol Bioinform* 11:1239–1252. <https://doi.org/10.1109/TCBB.2014.2323054>
 105. Yang J, Wang H, Wang W, Yu PS (2005) An improved biclustering method for analyzing gene expression profiles. *Int J Artif Intell Tools* 14:771–789.
<https://doi.org/10.1142/S0218213005002387>
 106. Angiulli F, Cesario E, Pizzuti C (2008) Random walk biclustering for microarray data. *Inf Sci* 178:1479–1497. <https://doi.org/10.1016/j.ins.2007.11.007>
 107. Dharan S, Nair AS (2009) Biclustering of gene expression data using reactive greedy randomized adaptive search procedure. *BMC Bioinformatics* 10:S27.
<https://doi.org/10.1186/1471-2105-10-S1-S27>
 108. Ayadi W, Elloumi M, Hao J-K (2012) Pattern-driven neighborhood search for biclustering of microarray data. *BMC Bioinformatics* 13:S11.
<https://doi.org/10.1186/1471-2105-13-S7-S11>
 109. Bryan K, Cunningham P, Bolshakova N (2006) Application of Simulated Annealing to the Biclustering of Gene Expression Data. *IEEE Trans Inf Technol Biomed* 10:519–525.
<https://doi.org/10.1109/TITB.2006.872073>
 110. Liu J, Li Z, Hu X, Chen Y (2009) Biclustering of microarray data with MOSPO based on crowding distance. *BMC Bioinformatics* 10:S9. <https://doi.org/10.1186/1471-2105-10-S4-S9>
 111. Coelho GP, de França FO, Von Zuben FJ (2009) Multi-Objective Biclustering: When Non-dominated Solutions are not Enough. *J Math Model Algorithms* 8:175–202.
<https://doi.org/10.1007/s10852-009-9102-8>
 112. Bleuler S, Prelic A, Zitzler E (2004) An EA framework for biclustering of gene expression data. In: *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*. pp 166-173 Vol.1
 113. Divina F, Aguilar-Ruiz JS (2006) Biclustering of expression data with evolutionary computation. *IEEE Trans Knowl Data Eng* 18:590–602.
<https://doi.org/10.1109/TKDE.2006.74>
 114. Gallo CA, Carballido JA, Ponzoni I (2009) BiHEA: A Hybrid Evolutionary Approach for Microarray Biclustering. In: Guimarães KS, Panchenko A, Przytycka TM (eds) *Advances in Bioinformatics and Computational Biology*. Springer, Berlin, Heidelberg, pp 36–47
 115. Huang Q, Tao D, Li X, Liew A (2012) Parallelized Evolutionary Learning for Detection of Biclusters in Gene Expression Data. *IEEE/ACM Trans Comput Biol Bioinform* 9:560–570. <https://doi.org/10.1109/TCBB.2011.53>
 116. Pontes B, Giráldez R, Aguilar-Ruiz JS (2013) Configurable pattern-based evolutionary biclustering of gene expression data. *Algorithms Mol Biol* 8:4.
<https://doi.org/10.1186/1748-7188-8-4>
 117. Mitra S, Banka H (2006) Multi-objective evolutionary biclustering of gene expression data. *Pattern Recognit* 39:2464–2477. <https://doi.org/10.1016/j.patcog.2006.03.003>
 118. Maulik U, Mukhopadhyay A, Bandyopadhyay S (2009) Finding Multiple Coherent Biclusters in Microarray Data Using Variable String Length Multiobjective Genetic Algorithm. *IEEE Trans Inf Technol Biomed* 13:969–975.
<https://doi.org/10.1109/TITB.2009.2017527>

119. Maulik U, Mukhopadhyay A, Bandyopadhyay S, et al (2008) Multiobjective fuzzy biclustering in microarray data: Method and a new performance measure. In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). pp 1536–1543
120. Divina F, Pontes B, Giráldez R, Aguilar-Ruiz JS (2012) An effective measure for assessing the quality of biclusters. *Comput Biol Med* 42:245–256. <https://doi.org/10.1016/j.combiomed.2011.11.015>
121. Yan D, Wang J (2013) Biclustering of gene expression data based on related genes and conditions extraction. *Pattern Recognit* 46:1170–1182. <https://doi.org/10.1016/j.patcog.2012.09.028>
122. Cano C, Adarve L, López J, Blanco A (2007) Possibilistic approach for biclustering microarray data. *Comput Biol Med* 37:1426–1436. <https://doi.org/10.1016/j.combiomed.2007.01.005>
123. Yang W, Dai D, Yan H (2011) Finding Correlated Biclusters from Gene Expression Data. *IEEE Trans Knowl Data Eng* 23:568–584. <https://doi.org/10.1109/TKDE.2010.150>
124. Biswal BS, Mishra P, Mohapatra A, Vipsita S (2016) A Survey on Greedy Based Algorithms for Biclustering of Gene Expression Microarray Data. In: 2016 International Conference on Information Technology (ICIT). pp 124–128
125. Sørlie T, Perou CM, Tibshirani R, et al (2001) Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proc Natl Acad Sci U S A* 98:10869–10874. <https://doi.org/10.1073/pnas.191367098>
126. Yu G, Yu X, Wang J (2017) Network-aided Bi-Clustering for discovering cancer subtypes. *Sci Rep* 7:1046. <https://doi.org/10.1038/s41598-017-01064-0>
127. Ben-Dor A, Chor B, Karp R, Yakhini Z (2003) Discovering local structure in gene expression data: the order-preserving submatrix problem. *J Comput Biol J Comput Mol Cell Biol* 10:373–384. <https://doi.org/10.1089/10665270360688075>
128. Tarazona S, Balzano-Nogueira L, Conesa A (2018) Chapter Eighteen - Multiomics Data Integration in Time Series Experiments. In: Jaumot J, Bedia C, Tauler R (eds) *Comprehensive Analytical Chemistry*. Elsevier, pp 505–532
129. Bersanelli M, Mosca E, Remondini D, et al (2016) Methods for the integration of multi-omics data: mathematical aspects. *BMC Bioinformatics* 17:S15. <https://doi.org/10.1186/s12859-015-0857-9>
130. Pavlidis P, Weston J, Cai J, Noble WS (2002) Learning Gene Functional Classifications from Multiple Data Types. *J Comput Biol* 9:401–411. <https://doi.org/10.1089/10665270252935539>
131. Chen X, Xu X, Huang JZ, Ye Y (2013) TW-k-means: Automated two-level variable weighting clustering algorithm for multiview data. *IEEE Trans Knowl Data Eng* 25:932–944. <https://doi.org/10.1109/TKDE.2011.262>
132. Ma T, Zhang A (2017) Integrate Multi-omic Data Using Affinity Network Fusion (ANF) for Cancer Patient Clustering. *ArXiv170807136 Q-Bio*
133. Chierici M, Bussola N, Marcolini A, et al (2020) Integrative Network Fusion: A Multi-Omics Approach in Molecular Profiling. *Front Oncol* 10:1065. <https://doi.org/10.3389/fonc.2020.01065>
134. Serra A, Fratello M, Fortino V, et al (2015) MVDA: a multi-view genomic data integration methodology. *BMC Bioinformatics* 16:261.

<https://doi.org/10.1186/s12859-015-0680-3>

135. Gligorijević V, Pržulj N (2015) Methods for biological data integration: perspectives and challenges. *J R Soc Interface* 12:.. <https://doi.org/10.1098/rsif.2015.0571>

Table Captions

Table 1: Distances between two data points $\mathbf{P} = (p_1, \dots, p_n)$ and $\mathbf{Q} = (q_1, \dots, q_n)$. \mathbb{R}^n is the n -dimensional vector space over real numbers. $\{0, 1\}^n$ is the n -dimensional vector space over binary numbers.

Table 2: Similarities between two data points $\mathbf{P} = (p_1, \dots, p_n)$ and $\mathbf{Q} = (q_1, \dots, q_n)$. \mathbb{R}^n is the n -dimensional vector space over real numbers. $\{0, 1\}^n$ is the n -dimensional vector space over binary numbers.

Table 3: Collection of Biclustering algorithms grouped according to the class of algorithm used.

Tables

Distance	Formula	Domain	Notes
Minkowski	$d_p(\mathbf{p}, \mathbf{q}) = \left(\sum_{k=1}^n p_k - q_k ^p \right)^{\frac{1}{p}}$	\mathbb{R}^n	Parametric distance; When $p \geq 1$, $d_p(\mathbf{p}, \mathbf{q})$ is a metric; when $p < 1$ $d_p(\mathbf{p}, \mathbf{q})$ is a semi-metric
Euclidean	$d_E(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{k=1}^n (p_k - q_k)^2}$	\mathbb{R}^n	Equal to Minkowski d_p when $p = 2$
Manhattan	$d_M(\mathbf{p}, \mathbf{q}) = \sum_{k=1}^n p_k - q_k $	\mathbb{R}^n	Equal to Minkowski d_p when $p = 1$
Chebychev	$d_C(\mathbf{p}, \mathbf{q}) = \max_{k=1}^n p_k - q_k $	\mathbb{R}^n	Equal to Minkowski d_p in the limit $p \rightarrow \infty$
Mahalanobis	$d_\Sigma(\mathbf{p}, \mathbf{q}) = \sqrt{(\mathbf{p} - \mathbf{q})^T \Sigma^{-1} (\mathbf{p} - \mathbf{q})}$	\mathbb{R}^n	
Jaccard	$d_J(\mathbf{p}, \mathbf{q}) = 1 - \frac{\mathbf{1}_{\mathbf{p}\mathbf{q}}}{\mathbf{1}_{\mathbf{p}} + \mathbf{1}_{\mathbf{q}} + \mathbf{1}_{\mathbf{p}\mathbf{q}}},$	$\{0, 1\}^n$	$\mathbf{1}_{\mathbf{p}\mathbf{q}}$ is the number of components equal to 1 in both data points, $\mathbf{1}_{\mathbf{p}}$ and $\mathbf{1}_{\mathbf{q}}$ are the number

			of components that are equal to 1 in \mathbf{P} and \mathbf{Q} respectively
Hamming	$d_H(\mathbf{p}, \mathbf{q}) = 1 - \mathbf{1}_{\mathbf{p}\mathbf{q}}$	$\{0, 1\}^n$	

Similarity	Formula	Domain	Note
Cosine similarity	$s_{\cos}(\mathbf{p}, \mathbf{q}) = \frac{\mathbf{p} \cdot \mathbf{q}}{\ \mathbf{p}\ \cdot \ \mathbf{q}\ }$	\mathbb{R}^n	
Dice Coefficient	$s_D(\mathbf{p}, \mathbf{q}) = \frac{2 \cdot \mathbf{1}_{\mathbf{p}\mathbf{q}}}{\mathbf{1}_{\mathbf{p}} + \mathbf{1}_{\mathbf{q}} + 2 \cdot \mathbf{1}_{\mathbf{p}\mathbf{q}}}$	$\{0, 1\}^n$	
Jaccard Index	$s_J = 1 - d_J(\mathbf{p}, \mathbf{q})$	$\{0, 1\}^n$	
Pearson Correlation	$s_\rho(\mathbf{p}, \mathbf{q}) = \frac{\text{Cov}(\mathbf{p}, \mathbf{q})}{\sigma_{\mathbf{p}} \cdot \sigma_{\mathbf{q}}}$	\mathbb{R}^n	Parametric, assumes normal distribution, Cov is the covariance, and $\sigma_{\mathbf{p}}$ is the standard deviation of \mathbf{p}
Spearman Correlation	$s_r(\mathbf{p}, \mathbf{q}) = \frac{\text{Cov}(r(\mathbf{p}), r(\mathbf{q}))}{\sigma_{r(\mathbf{p})} \cdot \sigma_{r(\mathbf{q})}}$	\mathbb{R}^n	Non-parametric, rank based, assumes a monotonic dependency

			between variables. $r(\mathbf{p})$ are the ranks of variable \mathbf{P}
Kendall Correlation	$s_{\tau}(\mathbf{p}, \mathbf{q}) = \frac{\sum_{i < j} \text{sgn}(p_i - p_j) \cdot \text{sgn}(q_i - q_j)}{\binom{n}{2}}$	\mathbb{R}^n	Non-parametric, rank based, assumes a monotonic dependency between variables. $\text{sgn}(x)$ is the sign of x . $\binom{n}{2}$ is the number of subsets of size 2 of a set of n elements.

Class	Algorithm	Acronym	Reference
Iterative greedy search	Direct Clustering	DC	(96)
	Cheng and Church	CC	(97)
	SMSR-based Biclustering	SMSR-CC	(98)
	HARP Algorithm	HARP	(99)
	Maximum Similarity Bicluster Algorithm	MSB	(100)
	Weighted Fuzzy-Based Maximum Similarity Bicluster Algorithm	WF-MSB	(101)
	Biclustering by Iteratively Sorting with Weighted	BISWC	(102)

	Coefficients		
	Bic. by Correlated and Large number of Individual Clustered seeds	BICLIC	(103)
	Intensive Correlation Search	ICS	(104)
Stochastic iterative greedy search	FLexible Overlapped biClustering	FLOC	(105)
	Random Walk Biclustering	RWB	(106)
	Reactive GRASP Biclustering	RGRASP-B	(107)
	Pattern-Driven Neighborhood Search	PDNS	(108)
Nature-inspired meta-heuristics	Simulated Annealing Biclustering	SA-B	(109)
	Crowding distance based Multi-Objective PSO Biclustering	CMOPSOB	(110)
	Multi-Objective Multi-Population Artificial Immune Network	MOM-aiNet	(111)
	Bleuler Alg.	Bleuler-B	(112)
	SEquential Evolutionary Biclustering	SEBI	(113)
	Biclustering via a Hybrid Evolutionary Algorithm	BiHEA	(114)
	Condition-Based Evolutionary Biclustering	CBEB	(115)
	EVOlutionary Biclustering based in EXpression PATterns	EvoBexpa	(116)
	Mitra & Banka Alg.	M&B	(117)
	Multi-Objective GA-based Biclustering	MOGAB	(118)
	Multi-Objective Fuzzy Biclustering	MOFB	(119)

	Sequential Multi-Objective Biclustering	SMOB	<i>(120)</i>
Clustering-based approaches	Biclustering based on related genes and conditions extraction	RGCE-B	<i>(121)</i>
	Possibilistic Spectral Biclustering	PSB	<i>(122)</i>
	Biclustering with SVD and Hierarchical Clustering	SVD&HC-B	<i>(123)</i>