Pattern Recognition 45 (2014) 637–648

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A semi-supervised fuzzy clustering algorithm applied to gene expression data

Ioannis A. Maraziotis

Biomedical Research Foundation, Academy of Athens, 4 Soranou Efesiou Street, Athens 11527, Greece

ARTICLE INFO

Article history:
Received 18 February 2012
Received in revised form
27February 2013
Accepted 16 May 2013
Available online 26 May 2014

Keywords:
Semi-supervised clustering
Pair-wise constraints
Fuzzy logic
Gene expression

ABSTRACT

Over the last decade there has been an increasing interest in semi-supervised clustering. Several studies have suggested that even a small amount of supervised information can significantly improve the results of unsupervised learning. One popular method of incorporating partial supervised information is through pair-wise constraints indicating whether a certain pair of patterns should belong to the same (Must-link) or different (Dont-link) clusters. In this study we propose a novel semi-supervised fuzzy clustering algorithm (SSFCA). The supervised information is incorporated via a method quantifying Must-link and/or Dont-link constraints. Additionally, we present an extension of SSFCA that allows the algorithm to automatically detect the number of clusters in the data. We apply SSFCA to the intrinsic problem of gene expression profiles clustering. The advantageous properties of fuzzy logic, inherited to SSFCA, allow genes to belong to more than one group, revealing this way more profound information concerning their multiple functioning roles. Finally, we investigate the incorporation of prior biological knowledge arriving from Gene Ontology in the process of selecting pair-wise constraints. Simulations on artificial and real life datasets proved that the proposed SSFCA significantly outperformed other standard and semi-supervised clustering methods.

1. Introduction

The importance of clustering algorithms and analysis in modern science is signified by the broad spectrum of their applications in various fields ranging from economy to meteorology and from physics to biology. In certain scientific fields like computational biology, clustering has been frequently employed as a useful exploratory technique allowing biologists to identify potentially meaningful relationships among genes or proteins.

Indeed, clustering was, and still remains one of the most popular methods for the analysis of gene expression from microarray experiments, used to provide insight into the structure of the data and to aid at the discovery of biologically related groups of genes [1]. Initial computational efforts employed classical clustering techniques (see Ref [2] for an extensive overview) for grouping genes according to their expression patterns, based on the experimentally validated assumption that genes involved in the same biological process exhibit similar patterns of variation. In most of the cases however, certain peculiarities of the problem at hand, such as the large degree of complexity in the measured entities and the amount of inherent noise present in microarray experiments, prevent standard clustering methods to provide adequate results in terms of pattern similarity and biological correlation.

A way to circumvent these problems is to incorporate additional sources of information. Several studies in the field of functional genomics have shown the advantages of integrating different types of biological data [3]. Nevertheless, in most of the so far developed clustering applications, prior biological knowledge is totally disregarded, while in combination with the similarity of expression profiles could lead to much better results.

An algorithmic family that could utilize prior knowledge on a certain field is the one of semi-supervised algorithms. These methods stand between purely unsupervised and fully supervised methods, benefiting from the advantages of both. Hence, they can be viewed as hybrids that incorporate partial prior information about the patterns, and thus do not suffer from the difficulties of supervised methods that have to be applied on fully annotated datasets. Indeed, while full pattern annotation can be an infeasible feature (at least for the majority of the real-life datasets) most of the times and irrespectively of the application, the user is able to know (e.g. from an outside source) whether some patterns should belong to the same cluster or not, at least for a small fraction of the whole dataset. Towards this direction, an always increasing number of studies that deal with semi-supervision, have shown that even a relatively small amount of supervised information can significantly improve the accuracy of clustering [4–6].

There are various methodologies for incorporating prior or additional knowledge in semi-supervised clustering. Different algorithms have been implemented pending on whether the information is given in terms of partial labelling [5,7] or in the

E-mail addresses: imaraziotis@bioacademy.gr, imaraziotis@gmail.com

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.05.007

form of pair-wise constraints, indicating whether some patterns should be located in the same cluster or not [8].

An additional discrimination of semi-supervised clustering algorithms is based on the number of phases implemented. Indeed there are algorithms that incorporate an intermediate phase, in which a metric is trained using the provided constraints or labels and subsequently used by an existing standard clustering algorithm [9]. In other methods, the need for a two phase clustering process is avoided, by integrating a term, in the main objective function of an algorithm that aims in satisfying the provided constraints [7,10].

Following the second methodology we propose a novel semi-supervised fuzzy clustering algorithm (SSFCA), based on pair-wise constraints. Key concept of SSFCA is the quantification of constraints that are retained and/or violated for every member of a certain cluster. Specifically, in the proposed method, guidance in clustering is provided following an approach where additional information (or prior knowledge) on a specific domain, is given on sets of either Must-link or Dont-link constraints or both. Henceforth, we will suppose E to be the set of Must-link constraints to be given in pairs $(x_i,x_j)\in E$ where patterns x_i and x_j should be assigned to the same cluster. We will also let $\varDelta$ be the set of pair-wise Dont-link constraints $(x_i,x_j)\in \varDelta$ such that x_i, x_j should be assigned to different clusters. At this point we should note that while pattern labelling may be difficult to be known a priori, most of the times it is less complicated to know whether or not some patterns should be members of the same group.

The proposed approach allows us to profit from the major advantages of fuzzy methods [11], over the crisp techniques. One important aspect of fuzzy logic, bearing significant importance in gene expression clustering, is that it facilitates the identification of overlapping clusters. Hence, by allowing genes to be members of various clusters with a variable degree of membership, fuzzy methods can, more suitably account for the complex relations governing gene regulation.

One important topic in semi-supervised clustering concerns the selection of constraints used to guide the clustering process. In the literature there are various techniques for selecting constraints that could be roughly divided in two general categories pending on whether the constraints are selected a priori [7] or actively [12]. In the later case (that could substantially increase the computational complexity of the overall method) the constraints are selected during the online operation of an algorithm. A common characteristic in most of the studies on this subject (using either one of the two categories) is that they do not incorporate external sources of supervised information. The selection is based solely on similarities and dissimilarities concerning the patterns to be clustered (i.e. internal criteria).

Despite that both of the aforementioned methods of selection could be incorporated in the overall proposed methodology, key goal of this study is to focus on the efficiency of SSFCA. Hence we utilize simplified frameworks of a priori selection, and experiment on both internal criteria and external sources of information. Several types of biological data could serve as a source of external information like genetic or protein interactions; Gene Ontology Consortium [13] however currently serves as the dominant approach for machine-legible functional annotation. Gene ontology (GO) data provide detailed information, about a large number of genes, and is considered to be of high quality since it is manually inspected to ensure accuracy.

Another issue that is of high interest is the one of automatically determining the number of clusters in the data. Indeed, in most of the clustering algorithms either purely unsupervised or semi-supervised, the exact number of clusters has to be given as one of the input parameters. This can be a serious drawback especially in the problem we are studying since a clustering algorithm is expected, to discover significant groups among genes that will lead to further biological studies.

We resolve this problem by integrating in the proposed algorithm, the technique of competitive agglomeration [14] suitably implemented under the semi-supervised frame. Via this extension, SSFCA initiates its operation from an overestimation of the number of centroids and through a process of deleting redundant clusters, finally determines the best number for the problem at hand.

Simulations performed on artificial and real-life datasets proved that SSFCA acquires highly similar groups of patterns (expression profiles) by significantly outperforming other unsupervised and semi-supervised algorithms.

2. Algorithms and methods

In this section we will present the methodology we follow to utilize prior information on a specific domain, given in the form of Must-link and Dont-link constraints, via a score quantifying the number of constraints that are retained or violated within a certain cluster for each one of its members. Subsequently we will provide detailed description for the SSFCA algorithm and how the aforementioned score is integrated in the fuzzy clustering process to guide clustering. Finally we will review how the technique of competitive agglomeration is integrated in the basic objective function of SSFCA, via an extension of the standard cardinality to the semi-supervised field, allowing the algorithm to automatically determine the number of clusters present in a dataset.

2.1. Quantifying constraints

As we have already mentioned, in the method we propose, guidance in clustering is provided via sets of either Must-link (set E) or Dont-link (set $\varDelta$) constraints or both. Under this approach, a specific pattern can be associated with more than one pair and/or kind of constraints. We could for example have three constraints that would impose some pattern to be in the same group/class with two patterns, while at the same time belong to different classes with the third pattern. Hence, we will insert a new score metric named pattern's constraints degree (pcd) based on the number of constraints that are retained and/or violated for a specific pattern j within a certain cluster i

$$\beta_{ij}=\frac{R_{ij}-V_{ij}}{T_j} \tag{1}$$

where R_{ij} and V_{ij} are the numbers of constraints that are retained and violated respectively for pattern j in cluster i, while T_j is the total number of constraints implicating pattern j.

We are using T_j instead of T_{ij} (that would have been the total number of constraints for pattern j in cluster i), since we are interested in having clusters whose members satisfy the majority of the overall constraints implicating them.

Concerning the range of value for β_{ij}, it is

$$\beta_{ij}=\begin{cases}1, & \{R_{ij}\to T_j \wedge V_{ij}\to 0\}\\ 0, & \{R_{ij}=V_{ij}\vee T_j=0\}\\ -1, & \{R_{ij}\to 0\wedge V_{ij}\to T_{ij}\}\end{cases} \tag{2}$$

As we can depict from Eqs. (1), (2), the range of pcd for every member of the cluster is within -1 and 1. When β_{ij} is unity or near that value, then most of the constraints regarding the specific pattern are retained within the cluster, while when the score approaches -1, most of the constraints are violated. At this point we should note that there could be cases where the number of constraints retained matches the number of violations or that simply there are no pair-wise constraints regarding a specific pattern in neither of the sets E and $\varDelta$. In cases such as these the value of pcd equals zero and especially in the latter case, we regard the pattern neutral in respect to the provided supervised information.

2.2. SSFCA: combining fuzzy clustering and semi-supervision

Fuzzy clustering is a partition–optimization technique that aims to group data based on pattern similarity in a non-exclusive manner by permitting each sample to belong to more than one cluster. Considering a finite set $\mathbf{X}=\{x_1, x_2, \ldots, x_N\}$ of N vectors in an D-dimensional space, the problem is to perform a partition on this dataset, into C fuzzy sets with respect to a given criterion. One of the most widely used fuzzy clustering methods is Fuzzy c-means (FCM) due to its efficacy and simplicity [15].

SSFCA, resolves a dual goal where on the one hand minimizes the distance among the patterns and the corresponding centroids while on the other, is simultaneously guided by the pairs of constraints towards the determination of more accurate and in certain cases, pending on the source of semi-supervision, more meaningful clusters. Hence, the basic idea of SSFCA is to expand the operation of the well known FCM algorithm into the semi-supervised domain by integrating pcds into its objective function, which is mathematically described as

$$J(U,V;X,\Pi)=\sum_{i=1}^{C}\sum_{j=1}^{N}a_{ij}^{m}d_{ij}^{2} \tag{3}$$

where a_{ij} is defined as

$$a_{ij}=nu_{ij}-(1-n)\beta_{\varphi(i,j)} \tag{4}$$

β_{ij} is the pcd and n is a small positive number ranging from 0 to 1. Objective function in (3) is subjected to the following constraints:

$$\sum_{i=1}^{C}u_{ij}=1,\quad \forall j\in[1,\ldots,N] \tag{5}$$

where u_{ij} represents the degree of membership of pattern j to the centroid/prototype v_i, and m is an exponential weight that controls the fuzzification degree of the membership (partition) matrix (throughout this study we follow that $m=2$). While a large repertoire of distance metrics could be incorporated in Eq. (3), in this study d_{ij} is the square Euclidean distance between the feature vector j and centroid i. Prior proceeding with the analysis of the objective function in Eq. (3) we will discuss about function φ and parameter n described in Eq. (4).

Under the fuzzy logic formalism, each pattern is a member of all existing clusters up to a certain degree indicated by the corresponding membership value. As a result of this, all pairs of pattern constraints should be checked for violations throughout all clusters. We can however consider that a pattern is part of the cluster for which it has the maximum membership value. Hence, we can check for pattern violations, on one cluster per pattern. This is accomplished using function φ, which returns the pcd value of a pattern in a certain cluster only if that pattern has a maximum membership value on that cluster, while in the opposite case a zero value is returned by φ, which is described

$$\beta_{\varphi(i,j)}=\begin{cases}\beta_{ij}, & \text{if } \arg(\max_k u_{kj})\equiv i\\ 0, & \text{else}\end{cases} \tag{6}$$

Similar techniques have been used before [7] and contribute to the faster convergence of the algorithm, reduce the computational complexity, and at the same time do not harm the generality of the solution. For reasons of simplicity henceforth we will refer to $\beta_{\varphi(i,j)}$ as β_{ij}.

A key parameter in SSFCA is n (4), which controls the degree up to which the constraints influence the overall clustering process. Specifically, the smaller the value of n is, the larger the influence of the pcds and hence the supervised information in the overall process.

There are several strategies that can be adapted for setting the value of n. For instance, if we have a large number of constraints and we are certain for the accuracy of the provided semi-supervised information we could consider n as a constant with a small value. In this study however, we follow a framework where n should have a small value during the first iterations, so as to guarantee the initial formulation of structural coherent clusters in terms of retained constraints. After these first steps of clusters initialization, based mainly on the supervised information, there should be a gradual increase of its value so as to allow an analogues degree of importance for patterns similarity and constraints detainment. Finally, after the supervised information has fulfilled its purpose of guiding the clustering process, the value of n should approach a value near its maximum (e.g. a value near unity) to allow the algorithm to converge based mainly (or even purely) on the similarity of patterns within the clusters.

A simple function simulating the behaviour we described regarding parameter n is

$$n(t)=(n_F-n_0)\times\frac{t_{\max}-t}{t_{\max}}+n_F \tag{7}$$

where $t_{\max}$ is the maximum number of iterations we allow the algorithm to run, n_0 is the initial value of parameter n, and n_F is final value of n and t is the index of current iteration. Functions such as the one described in Eq. (7) have been used in similar settings [16].

Following the methodology described in part I of the Appendix, we solve the minimization problem described by Eqs. (3), (4) and derive the update equations of SSFCA. We will, first, review the equation concerning the membership values given by

$$u_{st}=u_{st}^{patt}+u_{st}^{constr} \tag{8}$$

where s and t represent a specific pattern and centroid, respectively. The first term of Eq. (8) is responsible for the standard updating of the membership matrix, by taking into account solely the relative distance of a certain feature point to all prototypes, and follows the standard FCM update equations:

$$u_{st}^{\text{patt}}=\frac{(1/d_{st}^{2})^{1/m-1}}{\sum_{k=1}^{C}[1/d_{kt}^{2}]^{1/m-1}} \tag{9}$$

SSFCA handles the provided semi-supervised information towards a better partition of the initial population, to clusters whose members retain as many as possible of the provided constraints, through the second term of the basic updating formula (8), described as

$$u_{st}^{constr}=\frac{1-n}{2n}\left(\frac{1}{d_{st}^{2}}\right)^{1/(m-1)}\left(\frac{\beta_{st}}{\left(\frac{1}{d_{st}^{2}}\right)^{1/(m-1)}}-\overline{B_t}\right) \tag{10}$$

where $\overline{B_t}$ defined as

$$\overline{B_t}=\frac{\sum_{k=1}^{C}\beta_{kt}}{\sum_{k=1}^{C}(1/d_{kt}^{2})^{1/(m-1)}} \tag{11}$$

is the weighted average of pcds for pattern t, and evaluates the constraints (retained or violated) regarding pattern t throughout all clusters. As we can depict from Eqs. (10) and (11) both β_{st} and all pcds calculated in $\overline{B_t}$ are weighted by the distance of t from centroids enabling this way to consider their pattern similarity.

The constraint term, augments or suppresses memberships values of a specific pattern to a certain cluster, according to whether or not, the pattern is best suited in the cluster (in terms of retained constraints), when compared to all other clusters. SSFCA compares the weighted pcd value of pattern t in cluster s with the weighted average pcd of the same pattern. If the former is larger than $\overline{B}_t$, then the membership value of pattern t in cluster s is increased, while in the opposite case is decreased. At this point we should clarify that (see also Section 2.1) if there are no constraints for some pattern, then $u_{st}^{Constr}=0$ and its membership value is influenced only by the proximity of that pattern towards the corresponding sth prototype.

Regarding the updating of the centroids (details in part I of the Appendix) we have

$$v_i = \frac{1}{\sum_{j=1}^{N} a_{ij}^m} \sum_{j=1}^{N} a_{ij}^m x_j \tag{12}$$

where i, and j correspond to the ith cluster and jth patter, respectively.

An important final comment on SSFCA, is that by providing different functions for a_{ij} and/or β_{ij} than those described in Eqs. (1) and (4), we end up with different versions of SSFCA, and hence with a family of semi-supervised fuzzy clustering algorithms.

2.3. SSFCA and agglomeration process

In this section we will present an extension of SSFCA that aims in allowing the algorithm to automatically determine the number of clusters in a dataset. The proposed extension integrates the technique of competitive agglomeration process in SSFCA by including a regularization term to the objective function described in Eq. (3), which becomes

$$J'(U,V;X,\Pi) = \sum_{i=1}^{C}\sum_{j=1}^{N} a_{ij}^2 d_{ij}^2 - \gamma \sum_{i=1}^{C}\left[\sum_{j=1}^{N}(nu_{ij}+(1-n)\beta_{ij})\right]^2 \tag{13}$$

subjected to the constraints in Eq. (5). As we can depict from Eq. (13) m has now a fixed value of $m=2$. Parameter γ is a factor that controls the agglomeration process and we shall deal with the range of its value later in this section.

In standard fuzzy clustering, the cardinality of a cluster is the sum of the membership values for a specific cluster against all patterns

$$C_s = \sum_{j=1}^{N} u_{sj} \tag{14}$$

Cardinality is a measure of how dense a certain cluster is. However, in our integrated environment, the coherency of a cluster does not depend solely on the distance of the patterns from the centroid, but also from the number of retained constraints among the members of the cluster. Hence we insert the term Composite Cardinality (CC) as an extension of standard cardinality, in the sense that the membership of every pattern in a specific cluster is enriched by its corresponding pcd value

$$K_s = \sum_{j=1}^{N}\left[nu_{sj}+(1-n)\beta_{\varphi(s,j)}\right] \tag{15}$$

where by setting $n=1$, we get the definition of cardinality (14). Inspection of Eq. (15) makes clear the fact that our goal is to have clusters with large values of CC.

We can now proceed to the presentation of the solution regarding the minimization of the objective function J'. As detailed described in part II of the Appendix, the updating equations for the membership values of the reformulated SSFCA described in Eq. (13) now becomes

$$u_{st} = u_{st}^{patt} + u_{st}^{constr} + u_{st}^{clust} \tag{16}$$

where the first and second term are described via Eqs. (9) and (10), (11), respectively if we set $m=2$.

The analytical expression of the final term of Eq. (16) is given by

$$u_{st}^{clust} = \frac{\gamma}{nd_{st}^2}(K_s - \overline{K_t}) \tag{17}$$

where $\overline{K_t}$ is described by

$$\overline{K_t} = \frac{\sum_{k=1}^{C}[1/d_{kt}^2]K_k}{\sum_{k=1}^{C}[1/d_{kt}^2]} \tag{18}$$

and describes the weighted average of all clusters composite cardinalities. Weights reflect the proximity of feature point $\mathbf{x}_t$ in question from the centroids and it is an additional factor to consider pattern similarity. Hence, the last term of the main updating rule is responsible for comparing the CC, of a certain cluster with the average CC of all clusters. As we can depict from Eq. (17), the outcome of this term, can be either positive, in the case where the CC of a certain cluster, is higher than the average CC value, or negative in the opposite case. This technique allows the algorithm to reduce the CC of spurious clusters, by reducing the membership values of its members. Later on in SSFCA operation and if the CC of a cluster, drops below a certain threshold, this cluster will be deleted.

We have already mentioned, that γ in Eq. (13) is a parameter controlling the agglomeration process. Hence, its value should be slow in the beginning to encourage the formation of small coherent clusters and gradually should be increased to promote agglomeration. Finally, when the number of clusters approach an appropriate value, parameter γ, should slowly decay to allow the algorithm to converge. This ranging in the values of γ can be simulated by a function of the form

$$\gamma(t) = \exp\left(\frac{-t}{\tau}\right)\frac{\sum_{i=1}^{C}\sum_{j=1}^{N} a_{ij}^2 d_{ij}^2}{\sum_{i=1}^{C}\sum_{j=1}^{N} a_{ij}^2} \tag{19}$$

where t represents the current iteration and τ is a constant. Functions like (19) controlling the agglomeration process have been used in similar schemes [8,14] and the value of t was set equal to a small integer value (e.g. 10).

Since there is not a distance parameter in the regularization term we have added in SSFCA the equation concerning centroids update, is the same with the one described in Eq. (12), if we substitute $m=2$.

Next, we provide a description of the proposed algorithm in the form of pseudo-code:

Semi-supervised fuzzy clustering algorithm—SSFCA

```
Step 1: Set an overestimation value of $C_{oves}$ ($2 <= C_{oves} <= N$). Set the iteration index $t=0$, set $t_0=0$, randomly initialize the fuzzy $C_{oves}$ partition matrix $U^{(t)}$, the value of $CC_{Thres}$, the value of $t_{del}$, a termination criterion $\varepsilon$ and the maximum number of iterations $Max\_iters$. Calculate $n(0)$, $\gamma(0)$.
Step 2: Set $t=t+1$. Calculate the centers of the fuzzy clusters $\{\mathbf{v}_i^{(t)}|i=1, 2, \ldots, C_{oves}\}$ based on $U^{(t-1)}$ and Eq. (12) with $m=2$.
Step 3: Calculate the new partition matrix $U^{(t)}$ using $\{\mathbf{v}_i^{(t)}|i=1, 2, \ldots, C\}$ and eq (16).
Step 4: $Del=\emptyset$. For $i=1\ldots C_{oves}$
Step 4.1: Calculate CC value for current cluster $i$.
Step 4.2: If $CC_i < CC_{Thres}$ $Del = Del \cup \{CC_i\}$
Step 5: Sort Del. If $|Del|==1$ or $t_0 > t_{del}$ discard cluster $\mathbf{v}_j$ where $j=Del\{1\}$ and update the number of clusters to $C_{oves} = C_{oves}-1$ and set $t_0=0$, Else $t_o+=1$.
Step 6: Calculate $\Delta=\|U^{(t+1)}-U^{(t)}\|=\max_{i,j}|u_{ij}^{(t+1)}-u_{ij}^{(t)}|$. If $\Delta<\varepsilon$ or $t==$Max_Iters, then Stop, Else calculate $n(t)$, $\gamma(t)$ and go to Step 2.
```

Key goal of the proposed extension in SSFCA, is to allow the algorithm, to detect the correct number of centroids after initiating from an overestimation C_{oves}. This goal is resolved through a process of deleting redundant clusters based on the corresponding CC values. In applications where agglomeration has been incorporated for similar reasons [8,14], the cardinalities of the centroids are checked per iteration and clusters with values smaller than a pre-specified threshold, are deleted. However, as the complexity of the data increases and especially in our case where we expand the standard cardinality to CC, a scheme such as this may prove to be inefficient. Hence we adapt a slightly different method where, if there are more than one clusters whose CC values is less than the pre-specified threshold CC_{Thres}, SSFCA concludes that there was not enough time for convergence in the current state and does not proceed to any deletions. Nevertheless, if after a predetermined number of iterations t_{Del}, the scheme we described still stands, the cluster with the lowest CC value is deleted.

3. Results

In this part of the paper we describe the datasets, we have used, both real and artificial, as well as the measures we employed to test the validity of the proposed algorithm. We have compared SSFCA with two standard clustering algorithms FCM, and K-means [17], as well as CA [14] that we used to compare the results of cluster number determination. Additionally we used, HMRF-KMeans [18], a method based on hidden Markov models [19] and COP-Kmeans (CPK) [20] to check our algorithm with other semi-supervised methods. CPK however, provided better apodosis than the other two methods and hence, for reasons of clarity and simplicity, in the presented results we only use CPK algorithm.

Concerning the selection of the constraints, we provide evidence that even when simplified frameworks are employed, and irrespectively of whether we use both types of constraints or solely Must-link, SSFCA attained improved clustering results. In all of the simulations performed we have provided a number of constraints ranging from a minimum of zero to a maximum of 30% of the total number of patterns in a dataset. We performed 30 replications for each one of the algorithms employed in this study, and the average of the validity measures were used in all of the results presented below.

The maximum iterations number for SSFCA was set to 100 and for all simulations we present in this study the algorithm converged before reaching this limit while a decrease in the value of the termination criterion from its value setting we used ($1e-03$) did not present significantly improved results. Additionally, unless otherwise stated, the minimum value of n, n_0 was set to 0.3 and its maximum n_F to 0.9. Lower values of n_0 and larger values of n_F could some time lead to an increase of the iterations needed for SSFCA to converge and terminate its operation (especially with a large number of provided constraints, larger than 20% of the total patterns), with no special efficiency improvement.

3.1. *Artificial dataset (AD)*

This dataset has been artificially created initiating from real data as described in Ref. [21]. It consists of 400 patterns across 10 different experimental conditions. The dataset has 10 clusters.

3.2. *Y5 (RD1)*

In this data set first published by Cho et al. [22] we have the expression levels of more than 6000 genes, measured in 17 time points during two cell cycles of Yeast. As in Ref. [22], we have used a specific subset, abbreviated as Y5, consisting of 384 genes [19] visually identified to peak at five distinct time points, each one representing a certain phase of the cell cycle (Early G1, Late G1, S, G2, and M). All of the Y5 set member genes are annotated. The expression levels of each gene were normalized.

3.3. *Yeast sporulation (RD2)*

This data set [23] consists of more than 6400 genes measured across seven time points (0, 0.5, 2, 5, 7, 9, and 11 h) during the sporulation of the budding yeast.

All genes that had more than 20% of their expression values missing were excluded from further analysis. Applying a threshold of 1.6 for the root mean squares of the log2-transformed values, we came up with 474 genes [24] whose expression patterns significantly changed.

3.4. *Evaluation measures*

We have employed several metrics to test the apodosis of SSFCA, pending on whether the labelling of the patterns in a dataset was known or not.

Normalized mutual information (NMI) [25] is a measure on the efficiency of a clustering algorithm to reconstruct an underlying label distribution in the data, and is defined as

$$NMI = \frac{H(C)-H(C|C')}{(H(C)+H(C'))/2} \tag{20}$$

where C and C' are random variables representing the cluster assignments of the patterns and the known class labels of the patterns, respectively. The nominator of Eq. (20) is the mutual information between C and C', $H(C|C')$ is the conditional entropy of C given C' while $H(C)$ and $H(C')$ is the Shannon entropy of the corresponding variables.

Additionally we have used standard information retrieval measures like sensitivity and specificity as well as the adjusted rand index (ARI) [26]

$$ARI = \frac{2(ad-bc)}{(a+b)(b+d)+(a+c)(c+d)} \tag{21}$$

via variable a we represent the number of pairs that are members of the same cluster in C and same class in C', with b pairs with the same cluster but different class, c different class and cluster, and finally d different cluster but same class.

To evaluate the results of the algorithms on RD2 dataset for which no information about the labeling of the patterns was known we used Silhouette index (SI) [27]. SI is a cluster validity index that is used to judge the quality of any clustering solution C. It is a measure on the compactness and separation of clusters and its value varies from -1 to 1, where higher values indicate better clustering result.

3.5. *Determining the number of clusters*

Initially we tested the apodosis of SSFCA concerning the correct determination of the number of clusters. We have used datasets AD and RD1, for which the labelling of the patterns was known and thus we tested the algorithm in a more controlled setup for both real and artificial data.

To select constraints for AD and RD1 we created all pairs of patterns that share the same label, and called the resulting set Must. Additionally we created another set, formed of all pairs that were not members of the same class and called it Dont set. The total number of constraints N requested was provided as input. This number corresponds to a specific percentage of the total number of patterns in a dataset. Finally we created the Must-link constraints by random selection of $N_C=N/2$ pairs from the Must set. Analogously we created the set of Dont-link constraints by randomly selecting N_C pairs from the Dont set. In schemes like the one we described, there could be cases where more than one pair of constraints is created for a specific pattern. Cases such as this cause no problem to the operation of SSFCA since they are handled from the pcd of the pattern (Section 2.1).

As we have described in detail in Section 2.3, the proposed algorithm initiates its operation from an overestimation of clusters C_{OVES}. In every iteration of SSFCA, the CC values of all clusters are calculated and under certain conditions, the cluster with the worst CC (lowest value) is deleted. This iterative procedure is repeated until the algorithm reaches a number of centroids that correctly fit the data.

In the simulations performed, we have set the initial value of C_{OVES} to be 2, 3 and 4 times larger than the actual number of clusters (C_{act}) in the datasets. In Fig. 1 we can depict that SSFCA (in all range of provided supervision), determined the correct number of clusters for both datasets (i.e. 5 for RD1 and 10 for AD). The CA algorithm we used for comparison failed to detect the correct number.

Before we further continue the analysis of the results we should mention at this point that we have performed simulations with larger and smaller values of C_{OVES}. Specifically we have increased C_{OVES} up to 15% of the number of patterns in the datasets and in 70% of the simulations in all datasets (data not shown) the algorithm converged to the correct number of clusters with an increase however in the number of iterations (i.e. up to 150) needed for convergence. For values of C_{OVES} smaller than C_{act} the algorithm in all simulations did not perform any cluster deletion.

In general, the larger the dimensionality and complexity of a specific dataset, the more difficult becomes the clustering problem and hence the number of iterations needed for the algorithm to converge to a solution is increased. This conclusion can also be derived by inspection of Fig. 1. Specifically, we can conclude that the iterations needed by SSFCA to detect C_{act}, increase as the initial C_{OVES} value becomes larger. Furthermore, the increased complexity of real dataset RD1, accounts for the larger number of iterations spent on a certain number of clusters (prior deletion) in comparison to the artificial AD. However when C_{OVES} was initially set to 20, for both AD and RD1 (as we can depict from the illustrated cases of the same figure), the corresponding iterations were more in the artificial dataset than in RD1. This observation is explained, if we consider that the initial value of C_{oves} (i.e. 20) was closer to the actual number present in AD rather than RD1. Another example of this property is illustrated in Fig. 1(B), corresponding to RD1. The algorithm needs 7 iterations prior deciding on deleting a certain cluster, when $C_{OVES}=10$ and only 4 when $C_{OVES}=20$. Similar scheme stands for AD dataset.

Hence it becomes obvious, that SSFCA needs more time to delete a certain cluster, when initial C_{OVES} value approaches C_{act}. Indeed, when the process initiates from a large overestimation of the actual number, more clusters will have low values of CC and thus, it is easier to select the next one to be deleted. It is interesting, however, that in this case, SSFCA does not increase the iterations prior a cluster deletion when it approaches C_{act}. This property is elucidated by considering that SSFCA has spent much time adjusting the centers of the clusters (e.g. SSFCA spent 34 iterations to reach a number of clusters equal to 10 when initiated from 20 in RD1) thus most of the centroids fit the data much better and hence the algorithm needs less time in deciding the next cluster for deletion.

During SSFCA process, when a certain cluster is deleted, the patterns that were members of the corresponding centroid will become members of one or more of the remaining centroids. Hence, the locations of those centroids in the search space will be influenced, during the first iterations following the deletion, before converging to a new well established location. Based on the definition we gave for composite cardinality, we deduce that we could monitor how SSFCA adjusts the centroids to better fit the data, by tracking the CC values of the remaining (after every deletion) centroids.

Fig. 2 presents a detailed recording of CC values for the centroids of RD1 dataset. Specifically we can depict the course of CC variations

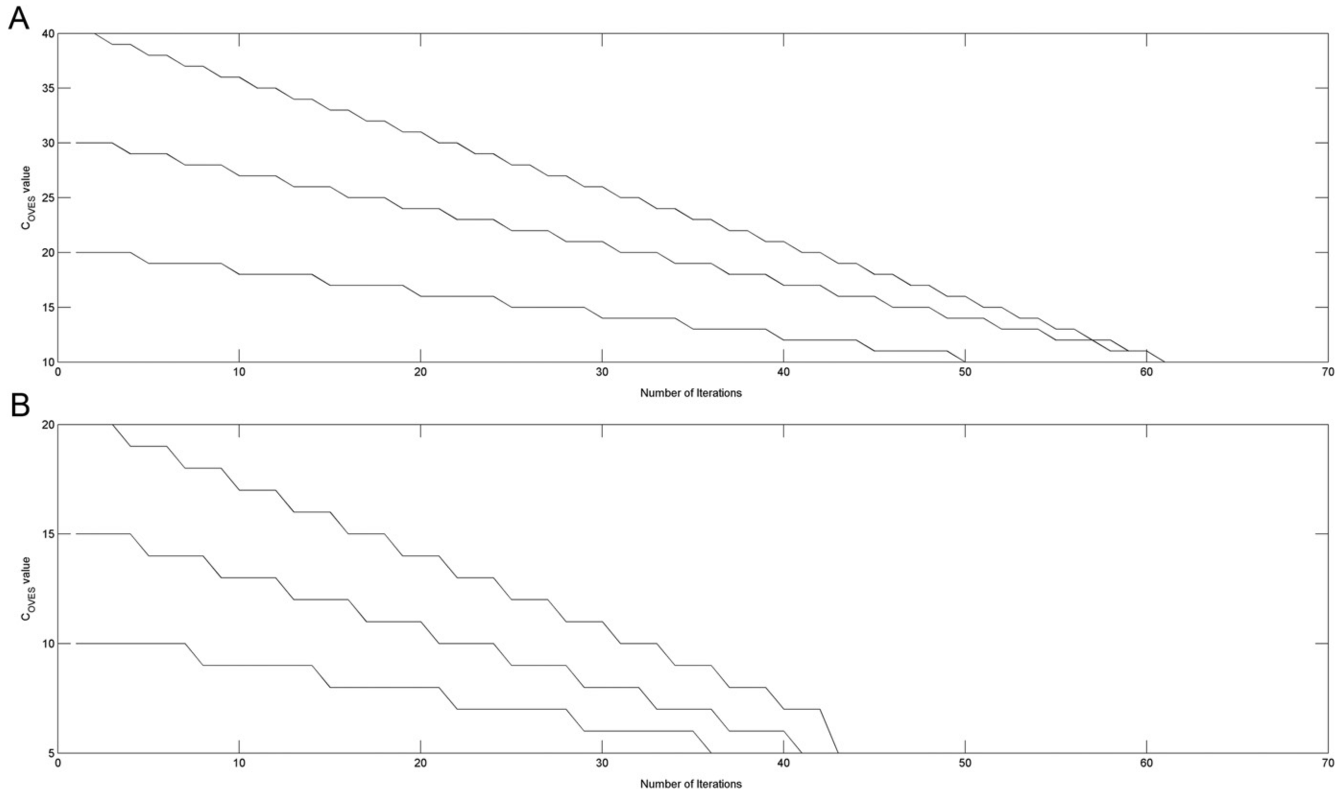


Fig. 1. In this figure we can depict the cluster deletion process of SSFCA through which the correct number of centroids is determined, for (A) AD and (B) RD1 datasets. In both datasets we have set the overestimated number of clusters to be 2, 3, and 4 times larger than the actual number.

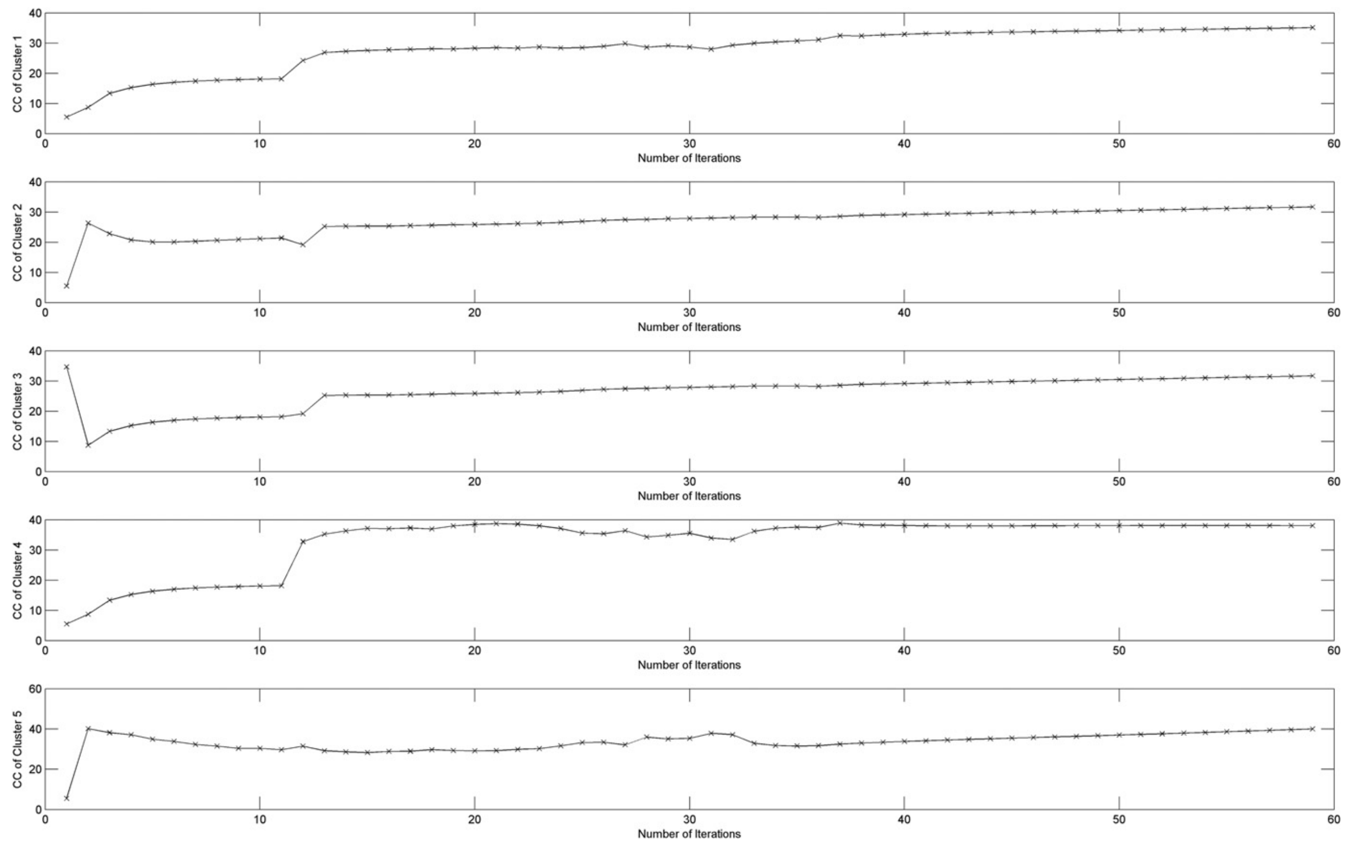


Fig. 2. Variation of composite cardinality (CC) values for the 5 centroids of RD1 dataset that were present throughout the cluster deletion process after initiating from $C_{oves}=10$.

of the 5 final centroids (represented as C1–C5, respectively) that were present throughout the process of consecutives deletions after initiating from an overestimation of 10 clusters.

In the 1st iteration, all clusters have various values of CC, ranging from 5 to 35 due to initialization. After only two iterations all clusters stabilize around a temporary threshold that is maintained for the next 10 iterations where a centroid deletion is performed. After the tenth iteration we can depict a change in the value of the clusters C1 to C4 while the CC value of C5 remains intact. We observe the next significant variation of CCs from iteration 29 to 35 where the number of clusters drops from 7 to the actual number of 5. After iteration step 40, the CC values of all clusters, stabilize around their final value, indicating that the algorithm has concluded on the number of clusters and continues its operation towards convergence.

As a final comment we have to mention that the threshold values for deleting a cluster CC_{Thres} and the iterations necessary prior deletion t_{Del} were experimentally determined from a wide range of different settings and were finally both set to 10 for all performed simulations. The algorithm presents highly similar behavior if we increase or decrease around 30% of these thresholds. If we decrease or increase further those values, the algorithm tends to produce an analogously larger or smaller number of clusters, respectively. In other words these values provide a means for a more detailed or coarse final clustering.

3.6. Clustering results

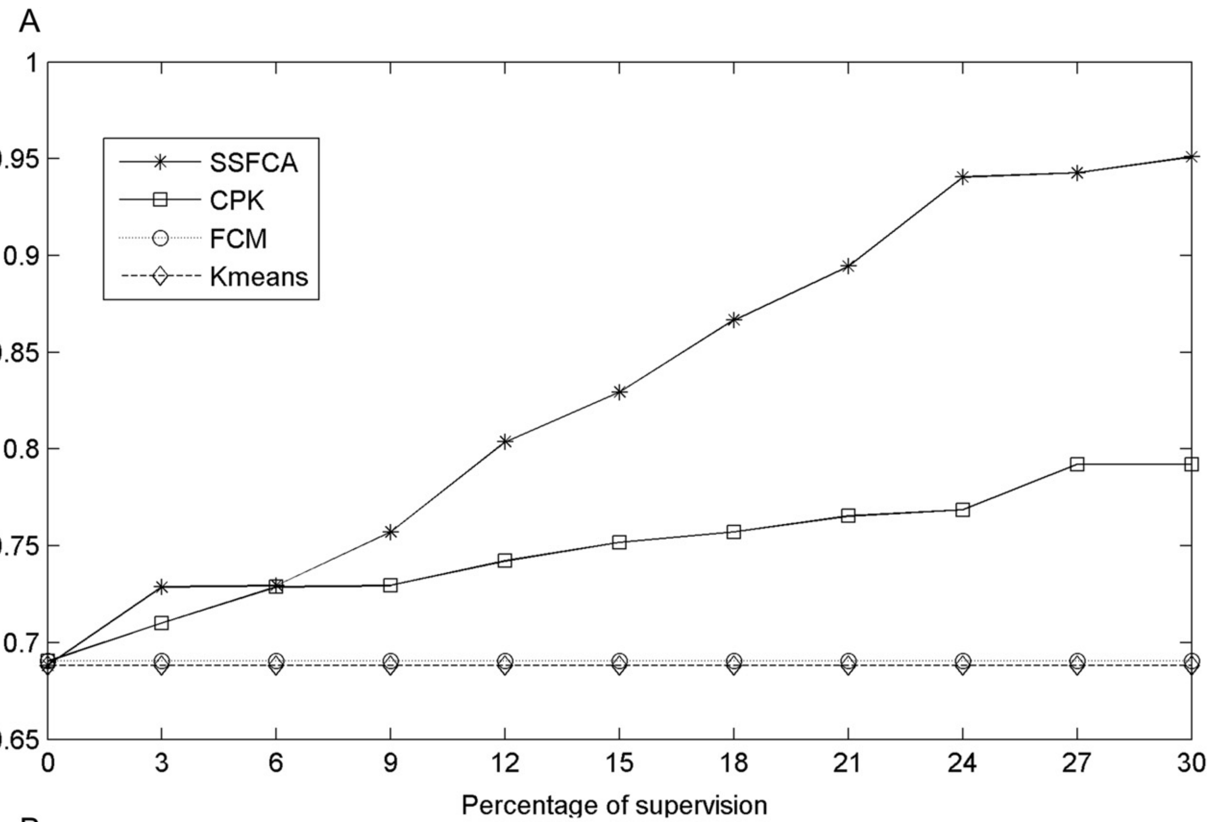
We will now review in detail the clustering performance of SSFCA in comparison to the rest of the algorithms employed. The selection of the constraints regarding RD1 and AD was the same with the one we described earlier.

As we can depict in Fig. 3(A) and (B) corresponding to datasets AD and RD1 respectively, SSFCA obtained better results than the other algorithms in terms of the NMI metric. Results in Fig. 3 are also confirmed from Table 1 where the proposed algorithm systematically had better performance in terms of sensitivity, specificity and ARI.

Inspection of Table 1 leads to the conclusion that SSFCA had very good apodosis concerning specificity in all ranges of supervised information for both datasets AD and RD1. In comparison to specificity, sensitivity had smaller values for small percentages of provided constraints. However, as the provided supervision was increased, the value of specificity was also systematically increased, and after 20% we had large values (i.e. more than 0.8) for both metrics and datasets.

In Fig. 3(A), we can see that between 0% and 6% of the provided constraints all algorithms displayed similar results of effectiveness. After that percentage, SSFCA systematically improved its performance up to a value of NMI equal to 0.95. On the real dataset, Fig. 3(B), all algorithms produced results with lower values of NMI, a fact indicating the increased complexity of RD1 in comparison to AD. However, it is evident that even the smallest amount of supervision can provide a significant improvement in the outcome of SSFCA.

At this point we have to stress on the fact that the way AD is constructed and RD1 is formulated they consist of groups of genes with similar expression, such that each one of its members has a peek in a certain time point (or points) throughout its expression pattern and hence a large value of the NMI metric reflects to a better clustering of the dataset. These results not only demonstrate the improved behaviour of the proposed algorithm but also indicate the benefits of semi-supervised clustering, especially when applied in datasets of increased complexity.



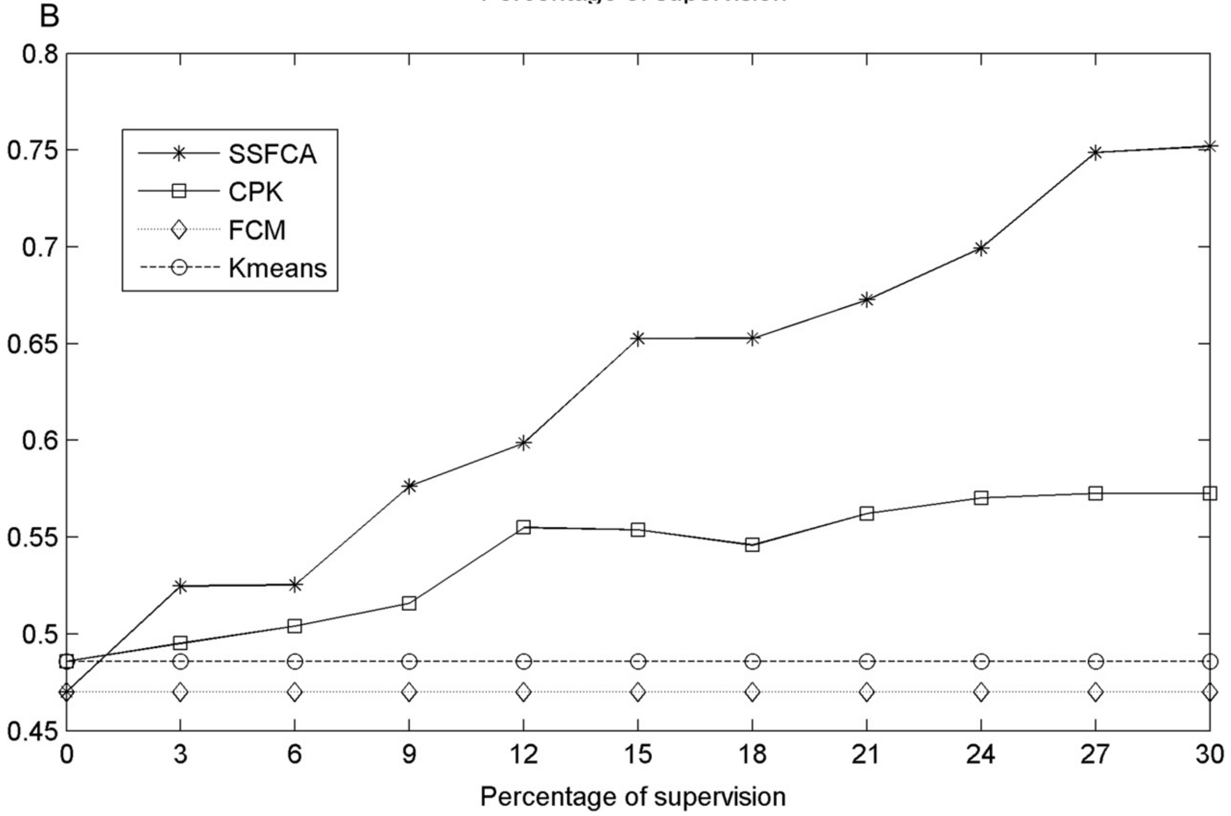


Fig. 3. Comparison of clustering results, on the considered algorithms, for different degrees of supervision on (A) AD and (B) RD1 datasets.

An important parameter of SSFCA that controls the degree up to which the provided constraints influence the clustering process, is n. In particular the smaller the value of n, the larger confidence we assign to the provided constraints (detailed discussion in Section 2.2). Hence we have conducted an experiment, based on RD1 dataset, where we have tested the effect of n in the overall process. Originating from the same value of $n_0=0.3$ we have tested SSFCA with different values n_F and specifically 0.7, 0.8, and 0.9 (which is the standard value we have used for all other simulations of this study).

Table 1
This table, reports detailed clustering results on sensitivity (Sens), specificity (Spes) and adjusted rand index (ARI) obtained by different algorithms, on AD and RD1 datasets. Concerning the proposed SSFCA, we provide results throughout all range of supervision, while for CPK only the best values that were acquired with the maximum percentage.

		RD1/AD		
Algorithms	Constraints percent	ARI	Sens	Spes
SSFCA	0	0.354/0.430	0.441/0.489	0.890/0.943
	3	0.505/0.519	0.610/0.654	0.885/0.931
	6	0.591/0.508	0.669/0.574	0.884/0.948
	9	0.595/0.562	0.673/0.633	0.892/0.951
	12	0.601/0.636	0.690/0.728	0.902/0.953
	15	0.603/0.705	0.731/0.751	0.910/0.968
	18	0.635/0.739	0.792/0.833	0.911/0.963
	21	0.679/0.833	0.811/0.852	0.913/0.983
	24	0.658/0.929	0.845/0.937	0.925/0.993
	27	0.725/0.870	0.876/0.975	0.931/0.975
	30	0.738/0.936	0.895/0.984	0.945/0.985
CPK	30	0.517/0.580	0.619/0.641	0.894/0.955
K-means	–	0.362/0.460	0.466/0.563	0.879/0.934
FCM	–	0.354/0.430	0.441/0.489	0.890/0.943

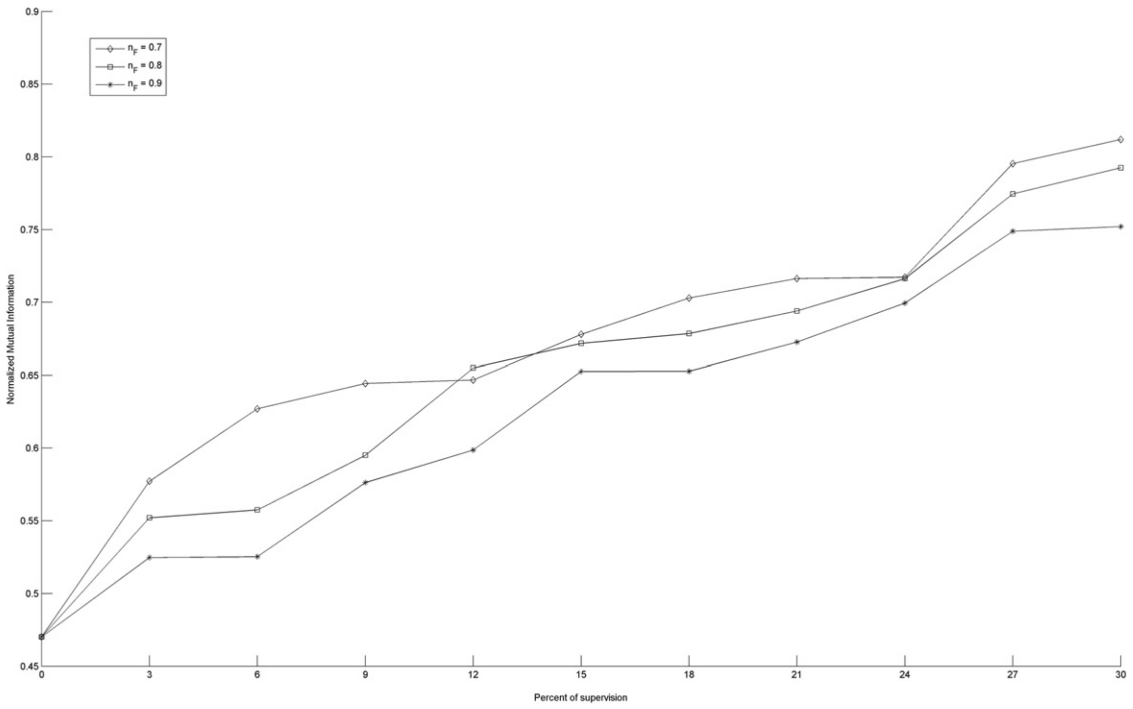


Fig. 4. Clustering results for different values of parameter n_F under the same sets of constraints for RD1 dataset.

In Fig. 4 we can see that, when we keep a low value for n_F we have better results of NMI, in almost all of the range of provided supervision and especially for small percentages. Hence, we could conclude that, in cases where we can be certain of the accuracy of the constraints supplied to SSFCA, setting n_F to a low value could further improve the clustering results in comparison to larger values.

At this point we should note that in all of the simulations we performed the number of Must-link constraints that were retained was smaller than the number of Dont-link. This observation implies that it could be more effective to provide a larger number of Must-link constraints, since in certain cases they are more difficult to be retained in comparison to Dont-link.

Based on this conclusion and the fact that in many applications (like the one considered in this study) it can be easier to assemble Must-link than Dont-link constraints, we have performed a final simulation where we employed only Must-link constraints. In this final experimental setup we checked the apodosis of SSFCA on dataset RD2 for which we had no information regarding neither the number of clusters nor the labelling of the patterns.

To select constraints on RD2, we applied a framework, based on a measure concerning gene pairs, utilizing both gene expression data and GO as a source of prior biological knowledge. GO has a directed acyclic graph structure (DAG) and describes genes in tree orthogonal taxonomies: biological process (BP), molecular function, and cellular component. In this study we have used BP information. The leaves of DAG are genes and the internal nodes are terms (annotations). The closer a node is to the root the more general is its biological class. The number of genes associated with a GO annotation term indicates how specific that term is, therefore based on this criterion we could discriminate between general and more specific terms. Genes sharing a more specific term are more likely to interact than genes that share a general term. While, there are many GO measures [28] in the literature

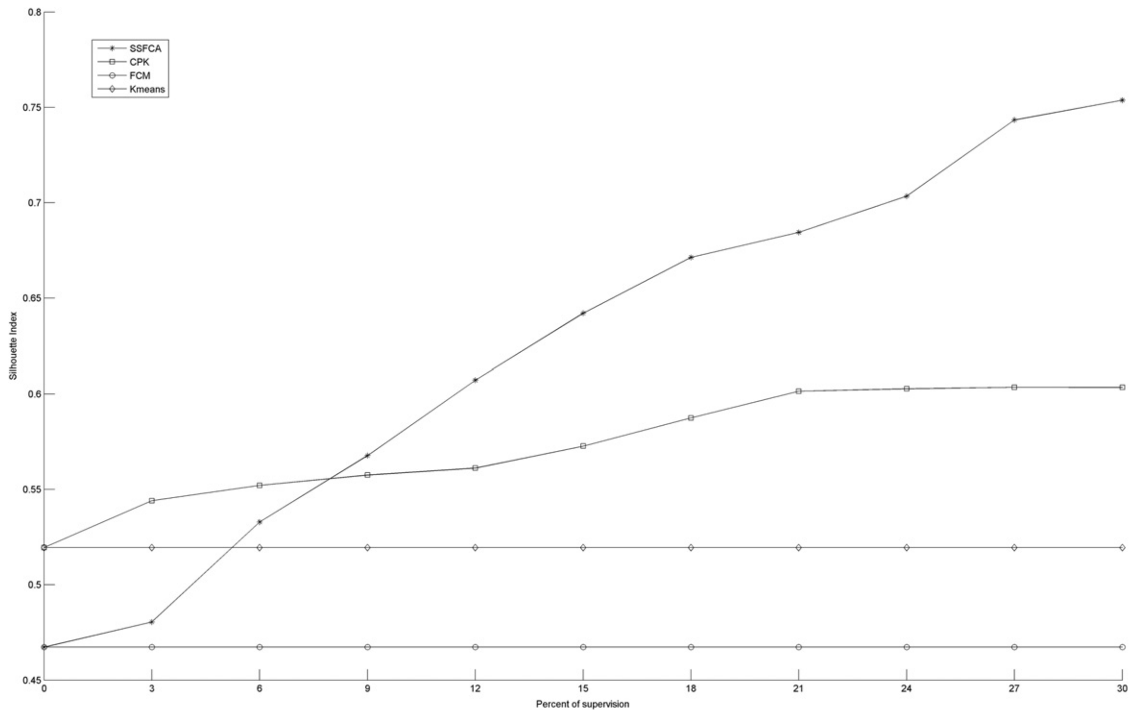


Fig. 5. Comparison of clustering results, on the considered algorithms, for different degrees of supervision on RD2 dataset.

that provide a quantitative degree of similarity between two specific genes in respect to their GO terms, Resnik's [29], is one of the most widely used.

We will now define a simple measure integrating similarity of expression profile patterns and gene annotations that will be used to guide the process of constraints selection

$$G_{ij} = \frac{d_{ij}^2/s_{ij}}{\sum_{t=1}^{N_G}\sum_{k=1}^{N_G} d_{tk}^2/s_{tk}}, \quad \forall j \in [1\ldots N] \tag{22}$$

where i and j correspond to a certain pair of genes, d_{ij} is the Euclidean distance between the expression profiles of i and j, s_{ij} is Resnik's similarity measure, and N_G is the number of pairs for which we have GO information. During the preprocessing step we calculated s_{ij} for all possible gene pairs and the pairs that had a zero value (i.e. no GO correlation), were excluded from further processing. While s_{ij} ranges from 0 to a maximum value, having 0 as worst case, the opposite occurs for Euclidean distance. Hence we are using the inverse of Resnik's measure to weight the similarity of profiles for every pair of genes. We finally normalize the score of a certain pair by the total sum of all pairs in a dataset. The range of the proposed measure ranges from 0 to a maximum value, having 0 as the best case.

In order to extract the necessary constraints from a given dataset, we adapt a methodology where every one of the genes present, is cross-checked against all others (provided there is corresponding GO information). Each one of these pairs is given a similarity degree based on Eq. (22). After the process is concluded for all genes, we sort the values of the corresponding pairs and determine the mean values of these scores. The final step of the proposed methodology is to randomly select a specific fraction of the pairs that have achieved a minimum score (smaller than the mean value) and use them to form the Must-link constraints.

Running the algorithm with values of C_{oves} ranging from 10 to 40 always the algorithm provided 6, as the best number of clusters. This result can be also confirmed by other similar studies on the same dataset [24].

We have checked the apodosis of the algorithms, based on SI, which is a metric used to check the results of a certain data partitioning and as we have already mentioned ranges from -1 to 1. The larger its value the better is the clustering.

In contrast to datasets AD and RD1, in the case of RD2, FCM algorithm presented worst performance than k-means and hence SSFCA originated its operation from the same value. However, again in this dataset, SSFCA managed to improve its performance from the initial stages of supervised information, as we can depict in Fig. 5. After around 10% and till the maximum amount of provided supervision, it consistently outperformed both the unsupervised and semi-supervised methods we have considered. Specifically for the maximum percent of supervision, achieved a value SI of 0.75 when the other semi-supervised method for the same amount of constraints had a value 0.59 and the best of the unsupervised algorithms (k-means in this dataset) had a value of 0.52.

The last performed simulation on dataset RD2 not only emphasize on the efficiency of the of the proposed SSFCA algorithm that was also evident in datasets AD and RD1 (Fig. 1A and B), but also on the advantages originating from using an external source of information integrated with the internal characteristics of the data on which clustering is applied (22).

4. Conclusions

A novel algorithm for semi-supervised clustering named SSFCA was presented in this paper. The proposed algorithm incorporates supervised information using pair-wise constraints indicating whether a pair of patterns should be part of the same cluster or not. A key parameter in the algorithm controls the degree up to which the constraints will influence the overall clustering process. While the user could have control over this parameter, in the standard operation of SSFCA is automatically handled. We have also presented an extension of the algorithm by integrating in the main objective function of SSFCA, a regularization term allowing this way the automatic determination of the number of clusters.

While the algorithm can have a broad range of applications in various scientific fields, in this study we have applied SSFCA to the intrinsic problem of gene expression profiles clustering. Grouping genes based on similarities of expression levels, can be a valuable process since it can determine the cooperation of certain genes under specific experimental set-ups or provide insights for the functioning of unknown genes. Additionally there is an extremely large amount of information concerning genes (e.g. GO, protein–DNA interactions, etc.) that could contribute greatly to the selection of constraints. Indeed the specific application allowed us to study the advantages originating from the integration of external sources of information in the clustering process. Specifically, we have used GO data as primary source of supervised information enriched with internal characteristics of the dataset to be clustered (i.e. pattern similarities). The improved results we acquired from this process suggest that SSFCA could be applied to other sources of biological information like protein–protein interactions as possible sources of supervised information.

Finally an interesting future extension of SSFCA, would be to reformulate its basic objective function, so that it could be integrated in the operation of an evolutionary algorithm in the form of a fitness function. Similar approaches have been used for standard FCM [30]. Under this framework SSFCA could take advantage of the ability of EAs to avoid local minima entrapment, which can be a problem for calculus based algorithms. An approach such as this could further elucidate the performance of SSFCA which as we have seen was considerably better than other well established semi-supervised and standard clustering techniques, both crisp as well as fuzzy.

Acknowledgments

I would like to thank Maria Tsekoura (MD) and Andrei Dragomir (PhD) for reading the manuscript and making suggestions, as well as the anonymous reviewers for their comments.

Appendix

In the following sections we will describe the derivation of the update equations for the objective functions of SSFCA.

Part I: semi-supervised fuzzy clustering algorithm

To minimize the objective function of SSFCA with respect to U, we apply the method of Lagrange multipliers to (3)

$$J(U,V;X,\Pi)=\sum_{i=1}^{C}\sum_{j=1}^{N}\left(nu_{ij}-(1-n)\beta_{ij}\right)^{m}d_{ij}^{2}-\sum_{j=1}^{N}\lambda_j\left[\sum_{i=1}^{C}u_{ij}-1\right] \quad (23)$$

Fixing the centroids

$$\frac{\partial J}{\partial u_{st}}=nm\left(nu_{st}-(1-n)\beta_{st}\right)^{m-1}d_{st}^{2}-\lambda_t \quad (24)$$

Setting Eq. (24) equal to zero and solving for u_{st} we get

$$u_{st}=\frac{1}{n}\left(\frac{\lambda_t}{nmd_{st}^{2}}\right)^{1/(m-1)}+\frac{1-n}{n}\beta_{st} \quad (25)$$

Using Eq. (25) and the constraints in Eq. (5), we obtain

$$\sum_{k=1}^{C}\left[\frac{1}{n}\left(\frac{\lambda_t}{nmd_{kt}^{2}}\right)^{1/(m-1)}+\frac{1-n}{n}\beta_{kt}\right]=1 \quad (26)$$

simplifying Eq. (26) we get

$$\frac{1}{n}\left(\frac{\lambda_t}{nm}\right)^{1/(m-1)}\sum_{k=1}^{C}\left(\frac{1}{d_{kt}^{2}}\right)^{1/(m-1)}=1-\frac{1-n}{n}\sum_{k=1}^{C}\beta_{kt} \quad (27)$$

solving Eq. (27) in respect to λ_t we have

$$\lambda_t^{1/(m-1)}=\frac{1-\frac{1-n}{n}\sum_{k=1}^{C}\beta_{kt}}{\frac{1}{n}\left(\frac{1}{nm}\right)^{1/(m-1)}\sum_{k=1}^{C}\left(\frac{1}{d_{kt}^{2}}\right)^{1/(m-1)}} \quad (28)$$

Substituting Eq. (28) to Eq. (25) and rearranging we get

$$u_{st}=\frac{\left(\frac{1}{d_{st}^{2}}\right)^{1/(m-1)}}{\sum_{k=1}^{C}\left(\frac{1}{d_{kt}^{2}}\right)^{1/(m-1)}}+\frac{1-n}{n}\beta_{st}-\frac{1-n}{n}\left(\frac{1}{d_{st}^{2}}\right)^{1/(m-1)}\frac{\sum_{k=1}^{C}\beta_{kt}}{\sum_{k=1}^{C}\left(\frac{1}{d_{kt}^{2}}\right)^{1/(m-1)}} \quad (29)$$

From Eq. (29) we derive the update equation (6) as described in Section 2.1.

Finally by differentiating Eq. (3) in respect to $\mathbf{v}_i$ and setting equal to zero we conclude to the updating equation regarding the centroids of the clusters as depicted in Eq. (12).

Part II: extension of semi-supervised fuzzy clustering algorithm

For the derivation of the update equations of J' (13) we will apply again the method of Lagrange multipliers, thus

$$J_A=\sum_{i=1}^{C}\sum_{j=1}^{N}a_{ij}^{2}d_{ij}^{2}-\gamma\sum_{i=1}^{C}\left[\sum_{j=1}^{N}\left(nu_{ij}+(1-n)\beta_{ij}\right)\right]^{2}-\sum_{j=1}^{N}\lambda_j\left[\sum_{i=1}^{C}u_{ij}-1\right] \quad (30)$$

fixing matrix $\mathbf{V}$ of the centroid vectors, differentiating on u_{st} and setting equal to zero we get

$$\frac{\partial J_A}{\partial u_{st}}=2na_{st}d_{st}^{2}-2\gamma nK_s-\lambda_t=0 \quad (31)$$

from the above, we get for a_{st}

$$a_{st}=\frac{\lambda_t+2\gamma nK_s}{2nd_{st}^{2}} \quad (32)$$

substituting a_{st} with Eq. (32), and replacing to Eq. (31) we have for u_{st}

$$u_{st}=\frac{\lambda_t+2\gamma nK_s}{2n^{2}d_{st}^{2}}+\frac{1-n}{n}\beta_{st} \quad (33)$$

Using the constraints of (5) and (33) we get

$$\sum_{k=1}^{C}u_{kt}=1\Rightarrow\sum_{k=1}^{C}\left(\frac{\lambda_t+2\gamma nK_k}{2n^{2}d_{kt}^{2}}+\frac{1-n}{n}\beta_{kt}\right)=1 \quad (34)$$

after rearranging and solving in respect to λ_t we get

$$\lambda_t=\frac{1-\frac{\gamma}{n}\sum_{k=1}^{C}\frac{K_k}{d_{kt}^{2}}-\frac{1-n}{n}\sum_{k=1}^{C}\beta_{kt}}{\frac{1}{2n^{2}}\sum_{k=1}^{C}\frac{1}{d_{kt}^{2}}} \quad (35)$$

Replacing Eq. (35) to Eq. (33) and rearranging we get for u_{st}

$$u_{st}=\frac{\frac{1}{d_{st}^{2}}}{\sum_{k=1}^{C}\frac{1}{d_{kt}^{2}}}+\frac{1-n}{n}\frac{1}{d_{st}^{2}}\left(\frac{\beta_{st}}{\frac{1}{d_{st}^{2}}}-\frac{\sum_{k=1}^{C}\beta_{kt}}{\sum_{k=1}^{C}\frac{1}{d_{kt}^{2}}}\right)+\frac{\gamma}{n}\frac{1}{d_{st}^{2}}\left(K_s-\sum_{k=1}^{C}\frac{K_k}{d_{kt}^{2}}\frac{1}{\sum_{k=1}^{C}\frac{1}{d_{kt}^{2}}}\right) \quad (36)$$

from where we deduce Eq. (16).

References

[1] M.B. Eisen, P.T. Spellman, P.O. Brown, D. Botstein, Cluster analysis and display of genome-wide expression patterns, Proc. Nat. Acad. Sci. 95 (1998) 14863–14868.

[2] A.W-C. Liew, H. Yan, M. Yang, Pattern recognition techniques for the emerging field of bioinformatics, Pattern Recognition 38 (2005) 2055–2073.

[3] I.A. Maraziotis, K. Dimitrakopoulou, A. Bezerianos, An in silico method for detecting overlapping functional modules from composite biological networks, BMC Syst. Biol. 2 (2008) 93.

[4] A.M. Bensaid, L.O. Hall, J.C. Bezdek, L.P. Clarke, Partially supervised clustering for image segmentation, Pattern Recognition 29 (1996) 859–871.

[5] W. Pedrycz, J. Waletzky, Fuzzy clustering with partial supervision, IEEE Trans. Syst. Man Cybern. 27 (1997) 787–795.

[6] H. Liu, S.-T. Huang, Evolutionary fuzzy clustering, Pattern Recognition Lett. 24 (2003) 3105–3113.

[7] W. Pedrycz, G. Vukovich, Fuzzy clustering with supervision, Pattern Recognition 37 (2004) 1339–1349.

[8] N. Grira, M. Crucianu, M. Boujemaa, Active semi-supervised fuzzy clustering, Pattern Recognition 41 (2008) 1834–1844.

[9] E.P. Xing, A.Y. Ng, M.I. Jordan, S. Russell, Distance metric learning with application to clustering with side-information, in: S.T.S. Becker, K. Obermayer (Eds.), Advances in Neural Information Processing Systems, vol. 15, MIT Press, Cambridge, MA, 2003, pp. 505–512.

[10] I.A. Maraziotis, A. Dragomir, A. Bezerianos, Semi-supervised fuzzy clustering networks for constrained analysis of time-series gene expression data, Lect. Notes Comput. Sci. 4132 (2006) 818–826.

[11] D. Dembele, P. Kastner, Fuzzy C-means method for clustering microarray data, Bioinformatics 19 (2003) 973–980.

[12] S. Basu, A. Banerjee, R.J. Mooney, Active semi-supervision for pairwise constrained clustering, in: Proceedings of the 2004 SIAM International Conference on Data Mining, 2004.

[13] Gene Ontology Consortium, Gene ontology: tool for the unification of biology, Nat. Genet. 25 (2000) 25–29.

[14] H. Frigui, R. Krishnapuram, Clustering by competitive agglomeration, Pattern Recognition 30 (1997) 1223–1232.

[15] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.

[16] E.C.-K. Tsao, J.C. Bezdek, N.R. Pal, Fuzzy Kohonen clustering networks, Pattern Recognition 27 (1994) 757–764.

[17] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[18] S. Basu, M. Bilenko, R.J. Mooney, A Probabilistic Framework for SemiSupervised Clustering, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004 pp. 59–68.

[19] A. Schliep, I.G. Costa, C. Steinhoff, A. Schonhuth, Analyzing gene expression time-courses, IEEE/ACM Trans. Comput. Biol. Bioinformatics 2 (2005) 179–193.

[20] K. Wagstaff, C. Cardie, S. Rogers, S. Schroedl, Constrained K-Means clustering with background knowledge, in: Proceedings of 18th International Conference on Machine Learning, 2001 pp. 577–584.

[21] K.Y. Yeung, D.R. Haynor, W.L. Ruzzo, Validating clustering for gene expression data, Bioinformatics 17 (2001) 309–318.

[22] R.J. Cho, M.J. Campbell, E.A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T.G. Wolfsberg, A.E. Gabrielian, D. Landsman, D.J. Lockhart, R.W. Davis, A genome-wide transcriptional analysis of the mitotic cell cycle, Mol. Cell 2 (1998) 65–73.

[23] S. Chu, J. DeRisi, M. Eisen, J. Mulholland, D. Botstein, P.O. Brown, I. Herskowitz, The transcriptional program of sporulation in budding yeast, Science 282 (1998) 699–705.

[24] S. Bandyopadhyay, A. Mukhopadhyay, U. Maulik, An improved algorithm for clustering gene expression data, Bioinformatics 23 (2007) 2859–2865.

[25] A. Strehl, J. Ghosh, R. Mooney. Impact of similarity measures on web-page clustering, in: Workshop on Artificial Intelligence forWeb Search, 2000, pp. 58–64.

[26] K.Y. Yeung, W.L. Ruzzo, An empirical study on principal component analysis for clustering gene expression data, Bioinformatics 17 (2001) 763–774.

[27] P. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65.

[28] F.M. Couto, M.J. Silva, P.M. Coutinho, Measuring semantic similarity between Gene Ontology terms, Data Knowl. Eng. 61 (2007) 137–152.

[29] P. Resnik, Using information content to evaluate semantic similarity in taxonomy, in: Proceedings of the International Joint Conference on Artificial Intelligence, 1995, pp. 448–453.

[30] L.O. Hall, I.B. Ozyurt, J.C. Bezdek, Clustering with a genetically optimized approach, IEEE Trans on Evolutionary Computation 3 (1999) 103–112.

Ioannis A. Maraziotis received his B.S. degree in Electrical Engineering (1999) from ATEI Kozani, B.S. degree in Informatics (2002) from Aristotle University of Thessaloniki, M.S. (2004) and Ph.D. (2007) in the field of Bioinformatics from the University of Patras. His research interests include recurrent structures, neural networks, fuzzy logic, graph theory, probabilistic modelling and their application in problems of bioinformatics and systems biology.