Towards a Classification Approach using Meta-Biclustering: Impact of Discretization in the Analysis of Expression Time Series

André V. Carreiro[1,2], Artur J. Ferreira[3,4], Mário A. T. Figueiredo[2,3] and Sara C. Madeira[1,2,*]

[1]KDBIO group, INESC-ID, Lisbon, Portugal

[2]Instituto Superior Técnico, Technical University of Lisbon, Portugal

[3]Instituto de Telecomunicações, Lisbon, Portugal

[4]Instituto Superior de Engenharia de Lisboa, Lisbon, Portugal

Summary

Biclustering has been recognized as a remarkably effective method for discovering local temporal expression patterns and unraveling potential regulatory mechanisms, essential to understanding complex biomedical processes, such as disease progression and drug response. In this work, we propose a classification approach based on meta-biclusters (a set of similar biclusters) applied to prognostic prediction. We use real clinical expression time series to predict the response of patients with multiple sclerosis to treatment with Interferon-β. As compared to previous approaches, the main advantages of this strategy are the interpretability of the results and the reduction of data dimensionality, due to biclustering. This would allow the identification of the genes and time points which are most promising for explaining different types of response profiles, according to clinical knowledge. We assess the impact of different unsupervised and supervised discretization techniques on the classification accuracy. The experimental results show that, in many cases, the use of these discretization methods improves the classification accuracy, as compared to the use of the original features.

1 Introduction

Recent years have witnessed an increase in time course gene expression experiments and analysis. In earlier work, gene expression experiments were limited to static analysis. The inclusion of temporal dynamics of gene expression is now enabling the study of complex biomedical problems, such as disease progression and drug response, from a different perspective. However, studying this type of data is challenging, both from the computational and the biomedical point of view [1]. In this context, recent biclustering algorithms, such as CCC-Biclustering [2], used in this work, have effectively addressed the discovery of local expression patterns. In the specific case of expression time series, the relevant biclusters exhibit contiguous time points.

In this work, we propose a supervised learning approach based on meta-biclusters for prognostic prediction. In this scenario, each patient is characterized by gene expression time series and

[*] To whom correspondence should be addressed. Email: sara.madeira@ist.utl.pt

each meta-bicluster represents a set of similar biclusters. Consequently, these biclusters represent temporal expression profiles which may be involved in the transcriptomic response of a set of patients to a given disease or treatment. The advantage of this approach, when compared to previous ones, is both the interpretability of the results and the data dimensionality reduction. The former is crucial in medical problems and results from the possibility to analyze class-discriminant biclusters and to find promising genes that explain different expression profiles found for different types of treatment response. The latter results from biclustering itself, as it is able to find local temporal patterns shared by a set of genes, which are used as features in the proposed classification method.

Following previous work [3], we present results obtained when analyzing real clinical expression time series with the goal of predicting the response of *multiple sclerosis* (MS) patients to treatment with Interferon (IFN)-β. Given the poor results reported when using discretized versions of the data [3], in this paper we assess the impact of new unsupervised and supervised discretization approaches on this type of data and their effects on the classification accuracy. The results show that discretization is no longer an issue, allowing us to move on towards the improvement of different steps of classification based on meta-biclustering.

The paper is organized as follows. Section 2 discusses related work on the classification of clinical expression time series and provides background on feature discretization (FD). The proposed method is described in detail in Section 3; specifically, we describe the meta-biclusters classifier and its three main steps (biclustering, meta-biclustering, and classification). The results obtained with and without meta-biclustering are presented in Section 4, thus assessing the impact of the discretization process in the classification accuracy with and without meta-biclusters. Finally, Section 5 draws conclusions and discusses future research directions.

2 Background

2.1 Classification of Clinical Expression Time Series

Regarding the case study, there are three main works which focused on it in recent years. Baranzini et al. [4] collected the dataset and proposed a quadratic analysis-based scheme, named *integrated Bayesian inference system* (IBIS). Lin et al. [5] proposed a new classification method based on *hidden Markov models* (HMM) with discriminative learning. Costa et al. [6] introduced the concept of constrained mixture estimation of HMM. A summary of their results can be found in [7].

Following those works, Carreiro et al. [7] have recently introduced biclustering-based classification in gene expression time series. The authors proposed different strategies revealing important potentialities, especially regarding discretized data. The developed methods included a biclustering-based *k-nearest neighbor* (kNN) algorithm, based on different similarity measures, namely: between biclusters, expression profiles, or between whole discretized expression matrices (per patient), and also a *meta-profiles* strategy, where they searched for the biclusters with similar expression profiles, computed the respective class proportions, using these as a classifying threshold. In the work reported in this paper, compared with [7] we note that the main advantages of meta-biclusters is the easier interpretation of the results, as we get, from the most class-discriminant meta-biclusters, the most promising sets of genes and time points (biclusters) involved in patient classification. In the meta-profiles method [7], we first have to compute the biclusters which represent the respective expression profiles.

Hanczar and Nadif [8] adapted bagging to biclustering problems. The idea is to compute biclusters from bootstrapped datasets and aggregate the results. The authors perform hierarchical clustering upon the collection of computed biclusters and select K clusters of biclusters, defined as *meta-clusters*. Finally, they compute the probability that a given element *(Example, Gene)* belongs to each meta-cluster by assigning the element to the most probable one. The sets of *Example* and *Genes* associated to each meta-cluster define the final biclusters. This technique has shown to reduce the biclustering error and the *mean squared residue* (MSR) in both simulated and real datasets. However, gene expression time series or classification problems, as we introduce in this paper, were not considered in this previous approach.

2.2 Feature Discretization

In this Section, we review FD methods, addressing unsupervised and supervised techniques. FD can be performed in supervised or unsupervised modes, *i.e.*, using or not the class labels, and aims at reducing the amount of memory as well as improving classification accuracy [9]. A good discretization method should be able to find an adequate and more compact (using less memory) representation of the data for learning purposes. Regardless of the type of classifier considered, FD techniques aim at finding a representation of each feature that contains enough information for the learning task at hand, while ignoring minor fluctuations that may be irrelevant for that task. As a consequence, FD usually leads to a set of features yielding both better accuracy and lower training time, as compared to the use of the original features.

The supervised mode may lead, in principle, to better classifiers. However, it has been found that unsupervised FD methods perform well on different types of data (see for instance [10, 11]). The unsupervised and supervised FD methods can also be classified as dynamic or static [12, 9]; while static methods treat each feature independently, dynamic methods try to quantize all features simultaneously, thus taking into account feature interdependencies. FD methods can also be categorized as local (discretization of some features based on a decision mechanism such as learning a tree) or global (discretize all the features); as a final categorization, the methods can work in a top-down or a bottom-up approach.

2.2.1 Unsupervised Methods

In this Subsection we review some unsupervised FD methods. In the context of unsupervised scalar FD [9], the most common static techniques are:

EIB *(equal-interval binning)* performs uniform quantization with a given number of bits per feature;

EFB *(equal-frequency binning)* [13] obtains a non-uniform quantizer with intervals such that, for each feature, the number of occurrences in each interval is the same; this technique is also known as *maximum entropy quantization*.

PkID *(proportional k-interval discretization)* [11] adjusts the number and size of the discretization intervals to the number of training instances, thus seeking a trade-off between bias and variance of the class probability estimate of a *naïve Bayes* (NB) classifier [14].

EIB is simple and easy to implement, but it is very sensitive to outliers, and thus may lead to inadequate discrete representations. In EFB, the quantization intervals are smaller in regions where there are more occurrences of the values of each feature; EFB is less sensitive to outliers, as compared to EIB.

In the EIB and EFB methods, one can choose exactly the number of discretization bins, by an input parameter. In contrast, PkID computes the adequate number of bins, as function of the number of training instances. The PkID method computes the number and size of discretized intervals proportional to the number of training instances, seeking an appropriate trade-off between the granularity of the intervals and the expected accuracy of probability estimation. For a given numeric attribute for which the number of instances that have a known value is v, it is discretized into $\sqrt{v}$ intervals, with $\sqrt{v}$ instances in each interval. As v increases, both the number and size of discretized intervals increase.

It has been found that unsupervised FD performs well in conjunction with several classifiers; in particular, EFB in conjunction with NB classification produces very good results [9]. It has also been found that applying FD with either EIB and EFB to microarray data, in conjunction with *support vector machine* (SVM) classifiers, yields good results [15]. The experimental results in [11] suggest that, in comparison to EIB and EFB, PKID boosts NB classifiers to a competitive classification performance with lower dimensional datasets, and better classification performance for larger dimensional datasets.

2.2.2 Supervised Methods

This Subsection is devoted to the description of supervised FD methods. The *information entropy minimization* (IEM [16]) method based on the *minimum description length* (MDL) principle [17] is one of the oldest and most applied methods for the task of supervised FD. The key idea of using the MDL principle is that the most informative features to discretize are the most compressible features. The IEM method is based on the use of the entropy minimization heuristic for discretization of a continuous value into multiple intervals as well as on the idea of constructing small decision trees. It works in a recursive approach computing the discretization cut-points in a way such that it minimizes the amount of bits to represent the data. It follows a top-down approach in the sense that it starts with one interval and split intervals in the process of discretization.

The method *IEM variant* (IEMV) proposed in [18] is also based on the MDL principle, using an entropy minimization heuristic to choose the discretization intervals. In fact, the authors propose a function based on the MDL principle, such that its value decreases as the number of different values for a feature increases. Experimental results show that these methods lead to better decision trees than previous methods.

The supervised static *class-attribute interdependence maximization* (CAIM) [19] algorithm aims to maximize the class-attribute interdependence and to generate a (possibly) minimal number of discrete intervals. The algorithm does not require the user to predefine the number of intervals, as opposed to some other discretization algorithms. The experimental results in [19] show the comparison of CAIM with six other state-of-the-art discretization algorithms. The discrete attributes generated by the CAIM algorithm almost always have the lowest number of intervals and the highest class-attribute interdependency. The highest classification accuracy was achieved with the CAIM discretization, as compared with the other six algorithms.

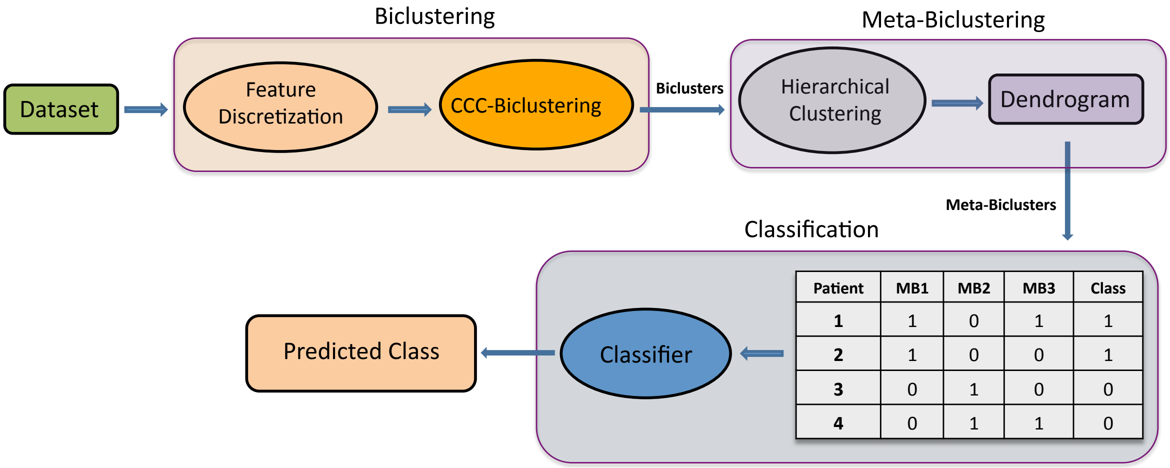
The *class-attribute contingency coefficient* (CACC) [20] is a static, global, incremental, supervised, and top-down discretization algorithm. Empirical evaluation of seven discretization algorithms on real and artificial datasets showed that CACC generates a better set of discrete features, improving the accuracy of classification. It shows promising results regarding execution time, the number of generated rules, and the training time of the classifiers.

A recent supervised discretization algorithm based on *correlation maximization* (CM) uses *multiple correspondence analysis* (MCA) to capture correlations between multiple features [21]. For each numeric feature, the correlation information obtained from MCA is used to build the discretization algorithm that maximizes the correlations between feature intervals and classes.

A detailed description of FD methods can be found in [12, 22, 23] and the many references therein. An unified view of several discretization methods is provided in [24].

3 Methods

In this Section we present the proposed supervised learning approach based on meta-biclusters, outlined in Figure 1 with its three main steps: *1) Biclustering; 2) Meta-Biclustering; 3) Classification.* The first step is the biclustering of the multiple expression time series after feature discretization. In the second step, a distance matrix is built for all the computed biclusters, on which a hierarchical clustering is performed. Cutting the resulting dendrogram at a given level returns a set of meta-biclusters. A **meta-bicluster** is thus a cluster of biclusters returned by a cut in a dendrogram of biclusters, that is, a set of similar biclusters. The third step starts by building a binary matrix representing, for each patient, which meta-biclusters contain biclusters from that patient. An example of such a matrix is also represented in Figure 1. Finally, in order to classify the instances, this binary matrix is used as input to a classifier.



Patient	MB1	MB2	MB3	Class
1	1	0	1	1
2	1	0	0	1
3	0	1	0	0
4	0	1	1	0

Figure 1: Workflow of a classifier based on meta-biclusters.

3.1 Biclustering

The biclustering step comprises feature discretization followed by CCC-Biclustering [2]. The resulting biclusters are given as the input to the meta-biclustering step.

3.1.1 Feature discretization

In this Subsection the used discretization approaches and techniques are explained in detail, as this is an important part of the analysis. We note that, as shown in Figure 1, the feature discretization process is part of the biclustering step, and thus, it is not directly related to the final classification matrix.

Unsupervised Methods Recently, two scalar unsupervised FD methods, based on the Linde-Buzo-Gray (LBG) algorithm [25], have been proposed [26]. The first method, named U-LBG1, applies the LBG algorithm individually to each feature, and stops when the *mean square error* (MSE) distortion falls below a threshold Δ or when a maximum number, q, of bits per feature is reached. Thus, for U-LBG1 a pair of input parameters (Δ, q) needs to be specified; using Δ equal to 5% of the range of each feature and $q \in \{4, \ldots, 10\}$ has been shown to be adequate [26] for different types of data. Naturally, U-LBG1 may discretize features using a variable number of bits. The second method, named U-LBG2, results from a minor modification of U-LBG1 by using a fixed number of bits per feature, q.

Both these FD methods exploit the same key idea that a discretization with a low MSE will provide an accurate representation of each feature, being suited for learning purposes. Previous work [26] has shown that this discretization method leads to better classification results than EFB on different kinds of (sparse and dense) data. Algorithm 1 presents the U-LBG1 procedure.

Algorithm 1 U-LBG1 - Unsupervised Linde-Buzo-Gray Discretization 1

Input: X, $n \times d$ matrix training set (d features, n patterns).
Δ: maximum expected distortion.
q: the maximum number of bits per feature.

Output: $\widetilde{X}$: $n \times d$ matrix, discretized version on X.
$Q^1, ..., Q^d$: set of d quantizers (one per feature).

```
for $i = 1$ to $d$ do
    for $b = 1$ to $q$ do
        Apply the LBG algorithm to the $i$-th feature to obtain a $b$-bit quantizer $Q_b(\cdot)$;
        Compute $\text{MSE}_i = \frac{1}{n} \sum_{j=1}^{n} (X_{ij} - Q_b(X_{ij}))^2$;
        if ($\text{MSE}_i \leq \Delta$ or $b = q$) then
            $Q^i(\cdot) = Q_b(\cdot)$;                    {/* Store the quantizer. */}
            $\widetilde{X}_i = Q^i(X_i)$;                  {/* Quantize feature. */}
            break;                                 {/* Proceed to the next feature. */}
        end if
    end for
end for
```

Supervised Methods For supervised FD we propose to use a discretization method based on *mutual information* (MI) [27], that is, we discretize each feature using a variable number of bits, maximizing its *normalized MI* (NMI) with the vector holding the class labels of each pattern, say y, given by

$$NMI(X_i, y) = MI(X_i, y)/\min(\log_2(|X_i|), \log_2(C)), \tag{1}$$

where $|X_i|$ is the number of values of (quantized) feature X_i and C is the number of classes. Notice that $0 \leq NMI(X_i, y) \leq 1$.

Our supervised *MI discretization* (MID) method follows these key ideas:

- the class label information is useful to guide the discretization process;
- begin by discretizing each feature with 1 bit, with a discretization cut-point such that it maximizes the NMI between the discrete binary feature and the class label;
- in a progressive approach, allocate more bits to the feature by recursively splitting each discretization interval into two new sub-intervals;
- the two new cut-points that break each of the existing discretization intervals are chosen in a way such that they maximize the NMI with the class label of the training data;
- for each feature, discretization stops whenever we reach the maximum number of bits or when there is a small (or no) increase on the NMI on the successive discretization stages.

MID, which is a supervised progressive discretization technique, is detailed in Algorithm 2. As in U-LBG1, this algorithm performs its actions solely on the training set portion of the data; it does not require the existence of a separate hold-out test set. In Algorithm 2, $@quant_{MI}$ denotes the discretization procedure mentioned above, in which each feature is discretized to reach maximum NMI. η represents the minimum expected increase on the NMI when we add one more bit to represent the discrete feature; we have found $\eta = 0.05$ adequate in our tests.

3.1.2 CCC-Biclustering

The goal of biclustering algorithms is to identify a set of biclusters $B_k = (I_k, J_k)$, where each bicluster is defined by a subset of genes and a subset of conditions, such that each bicluster satisfies specific characteristics of homogeneity [28]. For time series gene expression data analysis, Madeira et al. [2] defined the concept of CCC-Bicluster as follows: A *contiguous column coherent bicluster* (CCC-Bicluster) A_{IJ} is a subset of rows $I = \{i_1, \ldots, i_k\}$ and a subset of contiguous columns $J = \{r, r+1, \ldots, s-1, s\}$ such that $A_{ij} = A_{lj}$, $\forall i, l \in I$ and $\forall j \in J$. A CCC-Bicluster defines a string (a symbolic pattern) common to every gene in I for the time points in J. A CCC-Bicluster A_{IJ} is maximal if no other CCC-Bicluster exists that properly contains A_{IJ}: for all other CCC-Biclusters A_{LM}, $I \subseteq L \wedge J \subseteq M \Rightarrow I = L \wedge J = M$.

In this work, each patient is represented by a matrix with N_G (number of genes) rows and N_T (number of time points) columns. Since we are using gene expression time series, the biclustering algorithm used is CCC-Biclustering [2]. We end up with a set of CCC-Biclusters (named biclusters for simplicity) for each patient. Two examples of biclusters are represented in Figure 2. The values of an expression matrix A can be discretized to a set of symbols of interest, Σ, that represent distinctive activation levels. After discretization, matrix A' is transformed into matrix A, where $A_{ij} \in \Sigma$ represents the discretized value of the expression

Algorithm 2 MID - Mutual Information Discretization (Supervised)

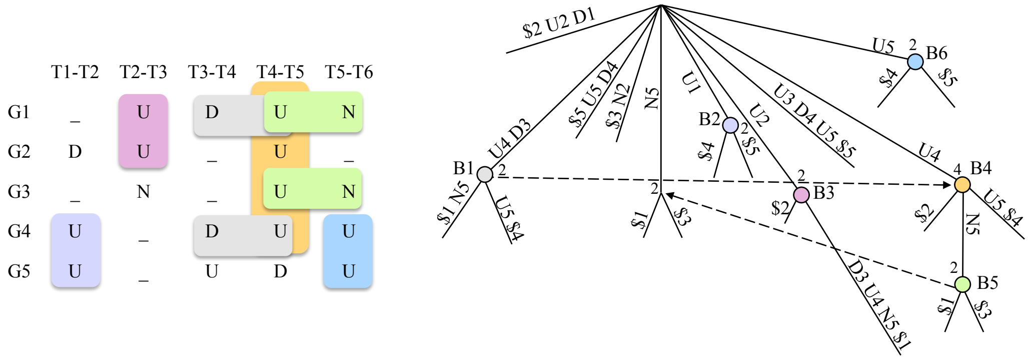
Input: X, $n \times d$ matrix training set (d features, n patterns).
y: n-length vector with class labels.
q: the maximum number of bits per feature.
η: the minimum expected increase on the NMI values.
Output: $\widetilde{X}$: $n \times d$ matrix, discretized version on X.
$Q^1, \ldots, Q^d$: set of d quantizers (one per feature).

```
for $i = 1$ to $d$ do
    $\widetilde{X}_{i1} = @quant_{MI}(X_i, 1)$;                      {/* Discretize with 1 bit. */}
    r$_{i1} \leftarrow @NMI(\widetilde{X}_{i1}, y)$;                  {/* Compute NMI with 1 bit. */}
    featureDone[i] = false;
end for
for $b = 2$ to $q$ do
    for $i = 1$ to $d$ do
        if featureDone[i] = true then
            continue;                                    {/* Feature is done. Move on to the next. */}
        end if
        $\widetilde{X}_{ib} = @quant_{MI}(X_i, b)$;                  {/* Discretize with one more bit. */}
        r$_{ib} \leftarrow NMI(\widetilde{X}_{ib}, y)$;                     {/* Compute NMI. */}
        if (r$_{ib}$ $-$ r$_{ib-1}$) $> \eta$ then
            $Q^i(.) = @quant_{MI}(b)$;                          {/* Keep (better) quantizer. */}
            $\widetilde{X}_i = Q^i_b(X_i)$;
        else
            featureDone[i] = true;       {/* Small increase in NMI. Stop allocating bits for feature. */}
        end if
    end for
end for
```

level of gene i in time point j. In Figure 2 a three symbol alphabet $\Sigma = \{D, N, U\}$ was used, where D corresponds to *down-regulation*, N to *no change*, and U to *up-regulation*. Consider now the matrix obtained by preprocessing matrix A using a simple alphabet transformation, that appends the column number to each symbol in the matrix and the generalized suffix tree built for the set of strings corresponding to each row in A. CCC-Biclustering is a linear time biclustering algorithm that finds and reports all maximal CCC-Biclusters based on their relationship with the nodes in the generalized suffix tree (see Figure 2 and Algorithm 3).

3.2 Meta-Biclustering

From the whole set of computed biclusters for all the patients (in [3] only the 25% most significant ones, in terms of p-value [2] were used), we compute the similarity matrix, S, where S_{ij} is the similarity between biclusters B_i and B_j. This similarity is computed with an adapted version of the Jaccard Index given by $S_{ij} = J(B_i, B_j) = \frac{|B_{11P}|}{|B_{01}|+|B_{10}|+|B_{11}|}$, where $|B_{11P}|$ is the number of elements common to the two biclusters that have the same symbol. $|B_{10}|$ and $|B_{01}|$ represent the number of elements belonging exclusively to bicluster B_i and B_j, respectively. Finally, $|B_{11}|$ represents the number of elements in common to both biclusters, regardless of the symbol. Note that it is important to consider the discretized symbols, since we are also comparing biclusters from different patients, and biclusters sharing the same genes and time



	T1-T2	T2-T3	T3-T4	T4-T5	T5-T6
G1	_	U	D	U	N
G2	D	U	_	U	_
G3	_	N	_	U	N
G4	U	_	D	U	U
G5	U	_	U	D	U

Figure 2: Maximal CCC-Biclusters in the discretized matrix and related nodes in the suffix tree.

Algorithm 3 CCC-Biclustering [2]

Input: Discretized gene expression matrix A
Output: Set of CCC-Biclusters

```
Perform alphabet transformation.
Obtain the set of strings $\{S_1, \ldots, S_{|R|}\}$.
Build a generalized suffix tree $T$ for $\{S_1, \ldots, S_{|R|}\}$.
for each internal node $v \in T$ do
    Mark $v$ as "Valid".
    Compute the string-depth $P(v)$
end for
for each internal node $v \in T$ do
    Compute the number of leaves $L(v)$ in the subtree rooted at $v$.
end for
for each internal node $v \in T$ do
    if there is a suffix link from $v$ to a node $u$ and $L(u) = L(v)$ then
        Mark node $u$ as "Invalid".
    end if
end for
for each internal node $v \in T$ do
    if $v$ is marked as "Valid" then
        Report the CCC-Bicluster that corresponds to $v$.
    end if
end for
```

points may not represent similar expression patterns. The similarity matrix S ($0 \leq S_{ij} \leq 1$) is then turned into a distance matrix D, where $D_{ij} = 1 - S_{ij}$. Using D, we perform a hierarchical clustering of the biclusters, building a dendrogram representing their similarity relationship. An example of such a dendrogram is shown in Figure 3. Using the dendrogram and a desired cutting-level, we obtain K meta-biclusters (clusters of similar biclusters).

3.3 Classification

The final step is the inference of the patients' response class. For this purpose, we build a binary matrix, C, with N_P rows (number of patients) and N_{MB} columns (number of meta-biclusters). C_{ij} equals 1 if patient$_i$ has at least one bicluster represented by Meta-Bicluster$_j$, and equals

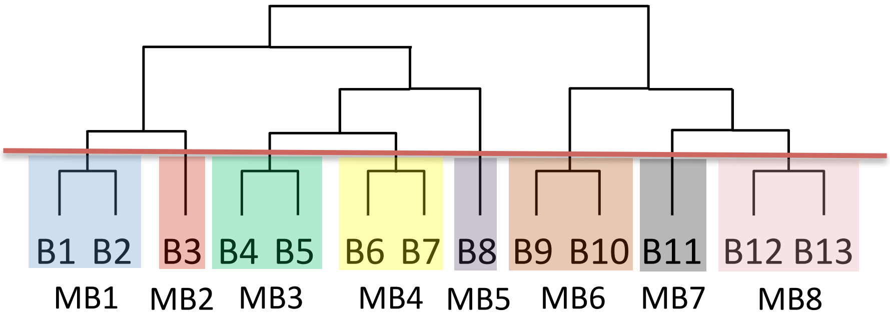


Figure 3: Meta-biclusters represented as clusters of biclusters in the dendogram.

0 otherwise. This binary matrix C is then used as input to supervised learning classifiers. In this work, we use *decision trees* (DT), *k-nearest neighbors* (kNN), *support vector machines* (SVM), and *radial basis function network* (RBFN) classifiers, available in the Weka toolbox (`www.cs.waikato.ac.nz/ml/weka`).

4 Results and Discussion

In this Section, we present and discuss the specificities of the MS case study, including the dataset description and preprocessing (Subsection 4.1). The main results obtained with the proposed classification approach are also shown and discussed in Subsection 4.2.

4.1 Dataset and Preprocessing

The dataset used as case study in this work was collected by Baranzini et al. [4]. Fifty two patients with relapsing-remitting (RR)-MS were followed for a minimum of two years after the treatment initiation. Then, patients were classified according to their response to the treatment, as good or bad responders. Thirty two patients were considered good responders, while the remaining twenty were classified as bad responders to IFN-β. Seventy genes were pre-selected by the authors based on biological criteria, and their expression profile was measured in seven time points (initial point and three, six, nine, twelve, eighteen and twenty-four months after treatment initiation), using one-step kinetic reverse transcription PCR [4]. In summary, from a machine learning perspective we have a binary classification problem with a total of $n = 52$ instances with 32 good responders and 20 bad responders on a d =470-dimensional space (70 genes × 7 time points).

In order to apply CCC-Biclustering [2], as part of the proposed meta-biclusters classifier, we normalized the expression data by time point to zero mean and unitary standard deviation, and discretized it using the techniques in Subsection 3.1.1: EFB and U-LBG1 (Algorithm 1) as unsupervised approaches, and MID (Algorithm 2) as a supervised technique. We note that, unlike in our previous work [3], discretization is now based on the whole training set, whereas before it was done individually for each patient. Instead of trying to design quantizers for each patient, we now group the data from several patients in each 5×4 cross-validation loop (with the same partitions as in [7]), and learn 490 quantizers, one for each feature ($d = 490$ features, resulting from 70 genes and 7 time points per gene). We remember that, in this work, we use all the computed biclusters, whereas in [3] only the 25% most significant ones were used (in terms of p-value, as in [2]).

In the case of standard classifiers, not able to deal with missing values directly, these were filled with the average of the closest neighboring values, after data normalization. Although CCC-Biclustering is able to handle missing values, for comparison purposes, the results reported in this paper were obtained with filled missing values, also for the meta-biclusters classifier.

4.2 Performance Evaluation

In this Subsection, some experimental results are reported, concerning the classification accuracy (assessed by cross-validation) of 4 well-known classifiers, mentioned in Subsection 3.3. We extracted several measures, including confusion matrices, kappa-statistics, weighted precision and recall, etc. Nonetheless, taking into account the obtained results and space constraints, we decided to show only the mean prediction accuracy values along with their standard deviation. However, to better understand the behavior of the classifiers, we will mention some of the other complementary metrics when necessary.

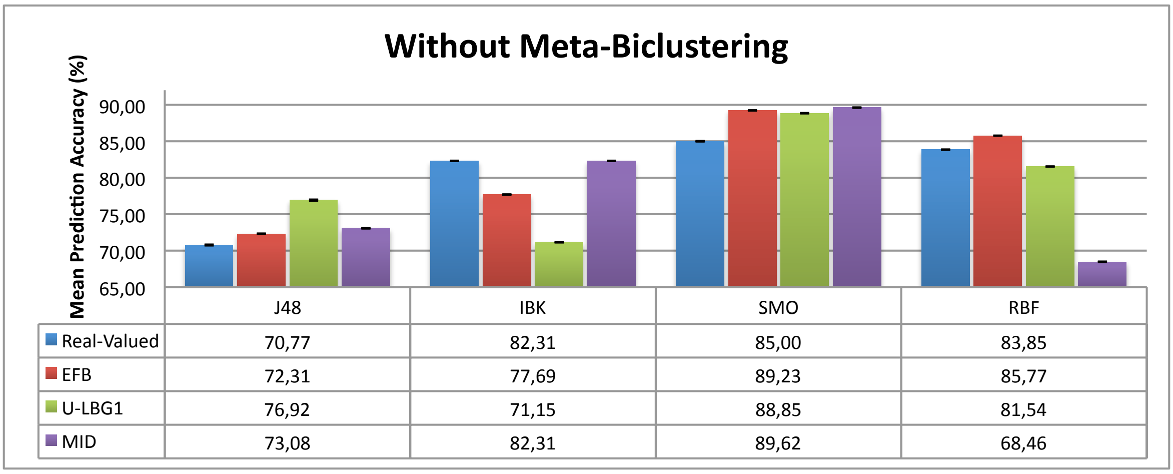
Given the poor results obtained with the application of standard classifiers on the first discretized versions of the data in [3], we decided to solve this issue by: 1) learning the quantizers in the whole training set, instead of individually per patient; 2) studying the performance of the classifiers when using supervised and unsupervised discretization techniques instead of solely unsupervised techniques. We then perform classification, using those discretized versions, in two distinct but related scenarios: without and with meta-biclustering.

4.2.1 Classification without Meta-Biclustering: real-valued and discretized versions

Figure 4 shows the mean prediction accuracy values obtained for the different state-of-the-art classifiers in the real-valued expression data, and using different discretized versions by EFB, U-LBG1, and MID using q =3 bits, as described in Subsection 3.1.1. Since the correspondent standard deviation values are so low (always $< \mathbf{0.1}$) their bars are almost imperceptible.

In contrast with what was reported in [3], Figure 4 shows that the use of these new discretization approaches causes no significant drop in the mean prediction values; in fact, the results obtained with the discretized versions of the data are, in some cases, better than those obtained with the real-valued dataset. This allows us to discard the discretization as the main problem with our method, as we hypothesized in [3], and focus on the improvement of the other steps. However, this conclusion needs to be supported by a set of more comprehensive experiments on different datasets (not reported here due to both time and space constraints).

Regarding this discretization results, we can conclude that the supervised discretization methods do not lead to the highest accuracy, as compared to the unsupervised approaches. As it happens with (non time series) microarray data, the SVM classifiers attain the best results (see, for instance [10, 15]). In fact, the MID discretization with Weka's SMO classifier achieves the overall highest accuracy (89.62%), being much higher than the baseline of good responders (61.54%). Moreover, the metrics of kappa-statistics, precision and recall for the SMO classifier, averaged across the different discretization techniques is, respectively, $k = 0.770$; precision $= 0.906$ and recall $= 0.892$, with standard deviation lower than 0.01.



	J48	IBK	SMO	RBF
■ Real-Valued	70,77	82,31	85,00	83,85
■ EFB	72,31	77,69	89,23	85,77
■ U-LBG1	76,92	71,15	88,85	81,54
■ MID	73,08	82,31	89,62	68,46

Figure 4: Mean prediction accuracy and standard deviation resulting from the classification without Meta-Biclustering: real-valued and discretized data (EFB, U-LBG1, and MID using 3 bits). Classification is achieved with Weka classifiers: J48 (DT), IBK (kNN), SMO (SVM), and RBF. The 5 x 4-fold CV evaluation , with the same partitions as in [7], is used.

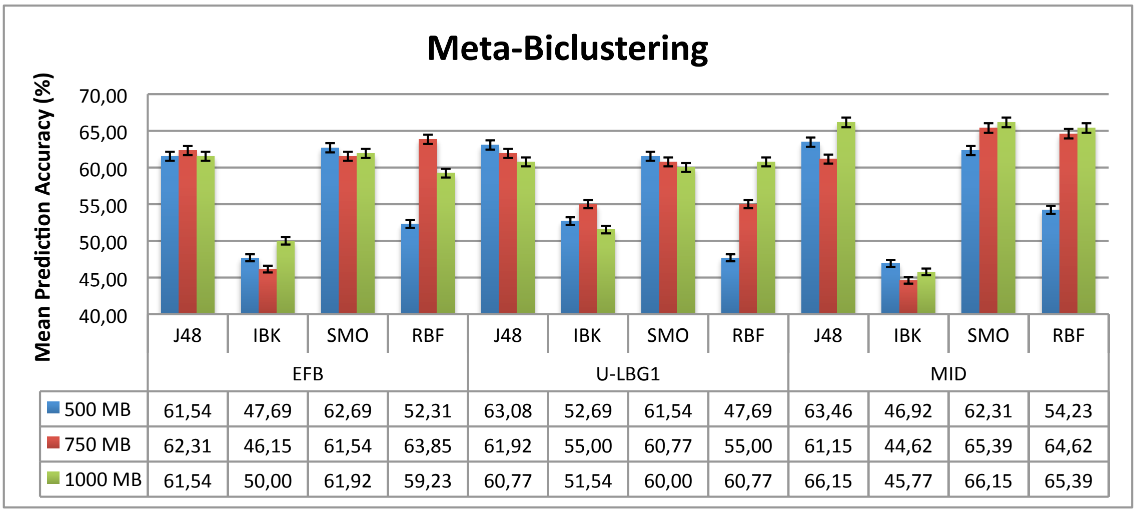
4.2.2 Classification using Meta-Biclusters

Figure 5 summarizes the mean prediction accuracies obtained by the method based on meta-biclusters with different discretization approaches: EFB, U-LBG1, and MID, using $q = 2$ bits. 500, 750 and 1000 meta-biclusters were used in this test. These results are still under, or only slightly above, the baseline of 61.54% of good responders. In fact, when we analyze different metrics and compare them with the previous situation (average across the different discretization algorithms used), the results are much lower. For the best performing classifier, the RBF Network, we have $k = 0.226$; precision $= 0.639$ and recall $= 0.633$ with standard deviation under 0.1. When we compare with a similar SMO classifier, the results are even worse: $k = 0.104$; precision $= 0.537$ and recall $= 0.635$ with standard deviation under 0.2. This reveals the confusion affecting the classifiers when using meta-biclusters, pointing out the need to refine the proposed method. Nevertheless, we consider that this is out of the scope of this paper, and it will be further investigated in our future work.

5 Conclusions and Future Work

The dataset herein considered presents a singular characteristic that may justify the difficulty in the classification: good responders have many similar biclusters in common with other good responders, but also with the bad responders. Bad responders, however, have few similar biclusters in common between the class. This suggests that there are different expression signatures associated to a poor response to IFN-β treatment or an absence of signature present in good responders [7]. From a machine learning perspective, aside from biclustering-based classifiers, this dataset is by itself challenging since it suffers from the so-called "curse of dimensionality", with a small number of instances and a large number of features (52 and 490, respectively).

Other properties of this data may also explain some of the challenges faced by the biclustering-based classifiers. These include class imbalance, biasing the prediction towards the good responders, and the reduced number of time points, in comparison with the number of genes, pos-



	EFB				U-LBG1				MID			
	J48	IBK	SMO	RBF	J48	IBK	SMO	RBF	J48	IBK	SMO	RBF
🟦 500 MB	61,54	47,69	62,69	52,31	63,08	52,69	61,54	47,69	63,46	46,92	62,31	54,23
🟥 750 MB	62,31	46,15	61,54	63,85	61,92	55,00	60,77	55,00	61,15	44,62	65,39	64,62
🟩 1000 MB	61,54	50,00	61,92	59,23	60,77	51,54	60,00	60,77	66,15	45,77	66,15	65,39

Figure 5: Mean prediction accuracy and standard deviation resulting from Meta-Biclustering with the discretization techniques EFB, U-LBG1 (unsupervised) and MID (supervised), using $q = 2$ bits. 500, 750 and 1000 meta-biclusters (MB) were considered. Classification is performed by the following Weka classifiers: J48 (Decision Tree), IBK (k-Nearest Neighbors), SMO (Support Vector Machine), and RBF (Radial Basis Function Network). The 5 x 4-fold CV evaluation , with the same partitions as in [7], is used.

sibly resulting in overfitting. Additionally, a common problem in clinical time series analysis is the reduced number of patients, also introducing important inconsistencies. Concerning interpretability, since the results are not very satisfactory, the analysis of the most class-discriminant biclusters will be addressed in future work.

The results presented in this work show that the required discretization step, as opposed to what was believed [3], is not the most limiting aspect of the proposed method. However, it is critical in guaranteeing the completeness and efficiency of the biclustering algorithm, which would otherwise have to rely on problem-specific heuristics, which we want to avoid by using a complete exhaustive search approach. These findings allow us to go beyond the discretization process and improve further steps of the method, since meta-biclustering is still in an embryonic stage. Namely, the chosen set of biclusters to which apply hierarchical clustering should take into consideration their correlation with the responder class, or be reduced by feature selection strategies. Furthermore, the similarity measures between biclusters is a crucial point in the analysis, and should be further investigated. Finally, instead of a binary matrix as the end result of meta-biclustering, we can have a matrix with entries (i, j) corresponding to the number of biclusters of patient i that belong to meta-bicluster j, for example.

Also as future work, we would like to validate the meta-biclustering method on different datasets (even if they are not clinical expression time series, where we would use other biclustering approaches). A good prediction accuracy would then allow us to directly study the most promising genes and/or time points, constituents of the most class-discriminant meta-biclusters, supporting the superiority of this method in terms of interpretability.

Acknowledgements

This work was partially supported by FCT - Fundação para a Ciência e a Tecnologia under projects PTDC/EIA-EIA/ 111239/2009 (NEUROCLINOMICS - Understanding NEUROdegenerative diseases through CLINical and OMICS data integration) and PEst-OE/EEI/LA0021/ 2011. AVC is funded by the doctoral grant SFRH/BD/82042/2011 from FCT. AJF was partially supported by the Polytechnic Institute of Lisbon under Grant SFRH/PROTEC/67605/2010. A preliminary version of this work was presented at PACBB 2012 [3]. The method was not changed but new extensive results studying the impact of discretization are presented. The number of used features (biclusters) was higher: we now use all the found biclusters, whereas before only the 25% most significant ones were used (in terms of p-value, as in [2]). The discretization strategy was changed, since we moved from learning the quantizers for each patient, to doing it for the whole training set in each cross-validation loop. Moreover, we now introduce a supervised discretization technique based on mutual information for this purpose.

References

[1] I. Androulakis, E. Yang and R. Almon. Analysis of time-series gene expression data: methods, challenges, and opportunities. *Annual Review of Biomedical Engineering*, 9:205–228, 2007.

[2] S. C. Madeira, M. C. Teixeira, I. Sá-Correia and A. L. Oliveira. Identification of regulatory modules in time series gene expression data using a linear time biclustering algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(1):153–165, 2010.

[3] A. V. Carreiro, A. J. Ferreira, M. A. T. Figueiredo and S. C. Madeira. Prognostic prediction using clinical expression time series: Towards a supervised learning approach based on meta-biclusters. In *6th Int. Conf. on PACBB*, pages 11–20. Springer-Verlag, 2012.

[4] S. Baranzini, P. Mousavi, J. Rio et al. Transcription-based prediction of response to IFN-beta using supervised computational methods. *PLoS Biology*, 3(1), 2005.

[5] T. Lin, N. Kaminski and Z. Bar-Joseph. Alignment and classification of time series gene expression in clinical studies. *Bioinformatics*, 24(13):i147–i155, 2008.

[6] I. Costa, A. Schönhuth, C. Hafemeister and A. Schliep. Constrained mixture estimation for analysis and robust classification of clinical time series. *Bioinformatics*, 25(12):i6–i14, 2009.

[7] A. V. Carreiro, O. Anunciação, J. A. Carriço and S. C. Madeira. Prognostic prediction through biclustering-based classification of clinical gene expression time series. *Journal of Integrative Bioinformatics*, 8(3):175, 2011.

[8] B. Hanczar and M. Nadif. Using the bagging approach for biclustering of gene expression data. *Neurocomputing*, 74(10):1595–1605, 2011.

[9] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, Morgan Kauffmann, 2nd edition, 2005.

[10] A. J. Ferreira and M. A. T. Figueiredo. Feature selection and discretization in microarray data. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Information Retrieval (IC3K-KDIR)*, pages 465–469. Paris, France, 2011.

[11] Y. Yang and G. Webb. Proportional k-interval discretization for naive-bayes classifiers. In *Proceedings of the 12th European Conference on Machine Learning*, (ECML), pages 564–575. Springer-Verlag, London, UK, 2001.

[12] J. Dougherty, R. Kohavi and M. Sahami. Supervised and unsupervised discretization of continuous features. In *International Conference Machine Learning (ICML)*, pages 194–202. Morgan Kaufmann, 1995.

[13] D. Chiu, A. Wong and B. Cheung. Information discovery through hierarchical maximum entropy discretization and synthesis. In *Knowledge Discovery in Databases*, pages 125–140. 1991.

[14] R. Duda, P. Hart and D. Stork. *Pattern Classification.* John Wiley & Sons, 2nd edition, 2001.

[15] P. Meyer, C. Schretter and G. Bontempi. Information-theoretic feature selection in microarray data using variable complementarity. *IEEE Journal of Selected Topics in Signal Processing (Special Issue on Genomic and Proteomic Signal Processing)*, 2(3):261–274, 2008.

[16] U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the International Joint Conference on Uncertainty in AI*, pages 1022–1027. 1993.

[17] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[18] I. Kononenko. On biases in estimating multi-valued attributes. In *Proceedings of the 14th International Joint Conference on Artificial intelligence - Volume 2*, (IJCAI), pages 1034–1040. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995.

[19] L. Kurgan and K. Cios. CAIM discretization algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 16(2):145–153, 2004.

[20] C.-J. Tsai, C.-I. Lee and W.-P. Yang. A discretization algorithm based on class-attribute contingency coefficient. *Inf. Sci.*, 178:714–731, 2008.

[21] Q. Zhu, L. Lin, M. Shyu and S. Chen. Effective supervised discretization for classification based on correlation maximization. In *IEEE International Conference on Information Reuse and Integration (IRI)*, pages 390–395. Las Vegas, Nevada, USA, 2011.

[22] S. Kotsiantis and D. Kanellopoulos. Discretization techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering*, 32(1), 2006.

[23] H. Liu, F. Hussain, C. Tan and M. Dash. Discretization: An Enabling Technique. *Data Mining and Knowledge Discovery*, 6(4):393–423, 2002.

[24] R. Jin, Y. Breitbart and C. Muoh. Data discretization unification. *Knowledge Information Systems*, 19(1):1–29, 2009.

[25] Y. Linde, A. Buzo and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28:84–94, 1980.

[26] A. J. Ferreira and M. A. T. Figueiredo. An unsupervised approach to feature discretization and selection. *Pattern Recognition*, 45(9):3048–3060, 2012.

[27] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

[28] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):24–45, 2004.