Mining Rules for the Automatic Selection Process of Clustering Methods Applied to Cancer Gene Expression Data

André C.A. Nascimento[1], Ricardo B.C. Prudêncio[1], Marcilio C.P. de Souto[2], and Ivan G. Costa[1]

[1] Center of Informatics, Federal University of Pernambuco, Recife, Brazil
{acan,rbcp,igcf}@cin.ufpe.br
[2] Dept. of Informatics and Applied Mathematics, Fed. Univ. of Rio Grande do Norte, Natal, Brazil
marcilio@dimap.ufrn.br

Abstract. Different algorithms have been proposed in the literature to cluster gene expression data, however there is no single algorithm that can be considered the best one independently on the data. In this work, we applied the concepts of *Meta-Learning* to relate features of gene expression data sets to the performance of clustering algorithms. In our context, each *meta-example* represents descriptive features of a gene expression data set and a label indicating the best clustering algorithm when applied to the data. A set of such meta-examples is given as input to a learning technique (the *meta-learner*) which is responsible to acquire knowledge relating the descriptive features and the best algorithms. In our work, we performed experiments on a case study in which a meta-learner was applied to discriminate among three competing algorithms for clustering gene expression data of cancer. In this case study, a set of meta-examples was generated from the application of the algorithms to 30 different cancer data sets. The knowledge extracted by the meta-learner was useful to understanding the suitability of each clustering algorithm for specific problems.

1 Introduction

New biotechnology methodologies, such as microrrays, allow the measurement of the expression of all genes of a cell sample. Medical researchers can use such methodologies to measure the expression of cancer cell samples of several patients with distinct cancer types. With these data, machine learning methods can be applied to perform computational diagnosis, i.e., to classify the type of a cancer cell based only on the gene expression profile. Another analysis of particular interest is the application of clustering to search for cancer tissues sharing similar molecular signatures. As demonstrated in [1] and [2], this kind of analysis does not only allows to distinguish between distinct cancer types, but also it has lead to the discovery of new cancer sub-types. Such gene expression data sets impose

C. Alippi et al. (Eds.): ICANN 2009, Part II, LNCS 5769, pp. 20–29, 2009.

several challenges to clustering methods, as they usually have a small number of observations (<100 cancer tissues), high dimensionality (> 1,000 of genes), the distribution of cancer types is unbalanced and there is a high level of noise [3].

While several clustering methods have been proposed in the bioinformatics literature, there is no consensus in the community on which method should be preferably used [4,5,6]. Recently, [7] performed a large scale evaluation of classical clustering methods over 35 data sets of cancer gene expression, which showed that k-means and mixture of multivariate Gaussians had best clustering performance for most of the data sets. That work also showed that hierarchical methods perform poorly for the majority of the sets. Despite of these experimental evidences, medical researchers are still faced with the question on which is the most appropriate method for a particular data set. As in other Machine Learning domains, there is a large variety of clustering algorithms considered suitable to be employed in the cluster analysis of given gene expression data sets. The selection of such algorithms requires empirical knowledge that is not easy to acquire. In general, the choice of algorithms is basically driven by the familiarity of biological experts to the algorithm, rather than the characteristics of the algorithms themselves and of the data [6].

This work is a first attempt to investigate the performance of clustering algorithms on gene expression data, by extracting rules that relate the characteristics of the data sets of gene expression to the performance achieved by the algorithms. The proposed work is directly derived from the Meta-Learning framework [8,9], originally proposed to support algorithm selection for classification and regression problems. According to [10], Meta-Learning can be defined by considering four aspects: (a) the problem space, P, representing the set of instances of a given problem class (usually classification and regression problems); (b) the meta-attribute space, M, that contains characteristics used to describe the problems (e.g., number of training examples, correlations between attributes, among others); (c) the algorithm space, A, that is the set of candidate algorithms to solve the problems in P; (d) a performance metric, Y, that measures the performance of an algorithm on a problem (e.g., classification accuracy estimated by cross-validation).

In this framework, Meta-Learning receives as input a set of meta-examples, in which each meta-example is derived from the empirical evaluation of the algorithms in A on a given problem in P. More specifically, each meta-example stores: (1) the values of the meta-attributes M extracted from a problem; and (2) the best candidate algorithm, considering the performance information Y. Hence, the *meta-learner* is only another learning technique that relates a set of predictor attributes (the meta-attributes) to a target attribute (the best algorithm).

The concepts of Meta-Learning have been extensively applied to select algorithms for classification and regression tasks (e.g., [11,12]). In recent years, Meta-Learning has been extended to other domains of application, as reported in [10]. In [13,14], for instance, the authors proposed the use of Meta-Learning to select algorithms for time series forecasting. In [15], the authors applied Meta-Learning to support the design of planning systems. In [16], Meta-Learning is employed to analyze the performance of meta-heuristics for optimization problems. Considering these applications, Meta-Learning can be viewed as a more general framework to algorithm selection. Hence, one would expect it to be useful in analyzing experiments in clustering of gene expression data.

In the current work, we applied a Meta-Learning procedure to analyze the experiments performed with three clustering algorithms (k-means, finite mixture of Gaussians and spectral clustering), since they were the winners among the seven clustering methods considered initially, on 30 data sets of cancer gene expression. Each data set was described by 13 descriptive meta-attributes and associated to a class label, which indicates the best clustering algorithm among the three candidates. In order to verify the viability of our proposal, different learning techniques (including Support Vector Machines, k-NN and two ensemble techniques) were used as meta-learners. We also applied the MLRules ensemble algorithm to extract interpretable knowledge, which provided useful insights on what makes an algorithm to perform better than another.

Section 2 describes the generation of meta-examples in our domain, as well as the techniques used for Meta-Learning. Section 3 introduces the experiments that evaluate the Meta-Learning process and discusses the obtained results. Finally, Section 4 presents some final remarks and future work.

2 Experimental Work

This research is directly derived from a previous work [7], in which we performed an empirical evaluation of clustering methods on different data sets of cancer gene expression. In the present work, we applied Meta-Learning to analyze the results of our clustering experiments, aiming to extract useful knowledge for selecting clustering methods. In this section, we briefly describe the experiments performed in [7], followed by the description of how the meta-examples were produced in the current work.

In [7], seven distinct clustering algorithms were analyzed: single linkage (SL), complete linkage (CL), average linkage (AL), k-means (KM), finite mixture of Gaussians (FMG), spectral clustering (SP), and Shared Nearest Neighbors algorithm (SNN). Also, four different proximity measures were employed, when applicable: Pearson's Correlation coefficient, Cosine, Spearman's correlation coefficient and Euclidean Distance. In the case of the Euclidean Distance, four different versions were applied: original (Z0), standardized (Z1), scaled (Z2) and ranked (Z3). The algorithms were evaluated in [7] over a set of 35 microarray datasets (See Table 1). These data sets present different values for characteristics such as type of microarray chip (second column), number of samples (third column), number of classes (fourth column) and distribution of samples within the classes (fifth column). In terms of the data sets, it is important to point out that microarray technology is usually available in two different platforms, cDNA and Affymetrix.

2.1 Meta-data

For each gene expression data set, we generated a meta-example composed by features (meta-attributes) that describe the data set and a label indicating the algorithm that obtained the best results. The criterion used for this labeling process and the meta-attributes considered are described in this section.

Table 1. Gene expression data sets considered

Dataset Name	Array Type	N	k	Samples per class	Class label
Armstrong-2002-v1	Affy	72	2	24, 48	FMG
Armstrong-2002-v2	Affy	72	3	24, 20, 28	FMG
Bhattacharjee-2001	Affy	203	5	139, 17, 6, 21, 20	FMG
Chowdary-2006	Affy	104	2	62, 42	-
Dyrskjot-2003	Affy	40	3	9, 20, 11	FMG
Golub-1999-v1	Affy	72	2	47, 25	KM
Golub-1999-v2	Affy	72	3	38, 9, 25	-
Gordon-2002	Affy	181	2	31, 150	FMG
Laiho-2007	Affy	37	2	8, 29	SP
Nutt-2003-v1	Affy	50	4	14, 7, 14, 15	FMG
Nutt-2003-v2	Affy	28	2	14,14	FMG
Nutt-2003-v3	Affy	22	2	7,15	-
Pomeroy-2002-v1	Affy	34	2	25,9	FMG
Pomeroy-2002-v2	Affy	42	5	10, 10, 10, 4, 8	SP
Ramaswamy-2001	Affy	190	14	11, 10, 11, 11, 22, 10, 11, 10, 30, 11, 11,11, 11, 20	KM
Shipp-2002-v1	Affy	77	2	58,19	SP
Singh-2002	Affy	102	2	50, 52	SP
Su-2001	Affy	174	10	26, 8, 26, 23,12, 11, 7, 27, 6, 28	KM
West-2001	Affy	49	2	25,24	FMG
Yeoh-2002-v1	Affy	248	2	43, 205	FMG
Yeoh-2002-v2	Affy	248	6	15, 27, 64, 20, 79, 43	KM
Alizadeh-2000-v1	cDNA	42	2	21, 21	KM
Alizadeh-2000-v2	cDNA	62	3	42, 9, 11	FMG
Alizadeh-2000-v3	cDNA	62	4	21, 21, 9, 11	FMG
Bittner-2000	cDNA	38	2	19, 19	KM
Bredel-2005	cDNA	50	3	31, 14, 5	FMG
Chen-2002	cDNA	179	2	104, 75	-
Garber-2001	cDNA	66	4	17, 40,4, 5	FMG
Khan-2001	cDNA	83	4	29, 11, 18, 25	-
Lapointe-2004-v1	cDNA	69	3	11, 39, 19	FMG
Lapointe-2004-v2	cDNA	110	4	11, 39, 19, 41	KM
Liang-2005	cDNA	37	3	28, 6, 3	FMG
Risinger-2003	cDNA	42	4	13, 3, 19, 7	KM
Tomlins-2006-v1	cDNA	104	5	27, 20, 32, 13, 12	KM
Tomlins-2006-v2	cDNA	92	4	27, 20, 32, 13	KM

Performance evaluation. In order to evaluate the performance of each combination of algorithm and proximity measure considered, an external validation index was used, the corrected Rand (cR) index [17]. The corrected Rand index takes values from -1 to 1, with 1 indicating a perfect agreement between the partitions generated by the clustering algorithm and the true classes known a priori, and values near 0 or negatives corresponding to cluster agreement found by chance. Unlike the majority of other indices, the cR is not biased towards a given algorithm or number of clusters in the partition [17].

The labeling of each meta-example was done according to the following procedure: at first, for each clustering algorithm, we selected the proximity measure that achieved the best results, i.e., the largest cR indices. In order to do so, we took into account only the partition with the number of clusters equal to the number of actual classes in the respective data set [7]. Finally, in order to detect the best algorithm for each data set, a ranking of the algorithms was made.

Only three algorithms were selected as class labels: FMG, KM and SP, since they were the only winners. In case of ties among these three algorithms, the data set on which it happened was excluded for generating a meta-example. This occurred in five data sets, indicated in Table 1 by a "-" at the last column. Hence, an actual number of 30 meta-examples were produced.

Meta-attributes. For the construction of the meta-dataset we used a set of 14 descriptive attributes (meta-attributes). Some of them were first proposed for the case of supervised learning tasks [9]. Recently, they have been also employed in the non-supervised learning context [18]. The samples (examples) considered in our study are labeled, i.e., they have a class label vector $Y = \{y_i\}$, $y_i \in \{1, ..., k\}$, where k is the number of classes for each data set. The class distribution among examples can be defined as $C = \{c_1, ..., c_k\}$, $c_j = \sum_j^N 1(y_i = j)$. Based on this and in other statistics, we define our set of meta-attributes as:

1. LgE: $\log_{10}$ of the number of examples. A raw indication of the available amount of training data.
2. LgREA: $\log_{10}$ of the ratio of the number of examples by the number of attributes. A rough indicator of the number of examples available to the number of attributes.
3. PMV: percentage of missing values. An indication of the quality of the data.
4. MN: multivariate normality, which is the proportion of examples transformed via T^2 that are within 50% of a Chi-squared distribution (degree of freedom equal to the number of attributes describing the example). A rough indicator of the approximation of the data distribution to a normal distribution.
5. SK: skewness of the T^2 vector. Same as the previous item.
6. Chip: type of microarray technology used (either cDNA or Affymetrix).
7. PFA: percentage of the attributes that were kept after the application of the attribute selection filter.
8. PO: percentage of outliers. In this case, the value stands for the proportion of T^2 distant more than two standard deviations from the mean. Another indicator of the quality of the data.
9. NRE: normalized relative entropy. An indicator of how uniformly examples are distributed among classes, i.e. the divergence between the actual class distribution and an uniform distribution. Its calculation is made using the Kullback-Leibler divergence equation, normalized by $2 \log k$, where k is the total number of classes. Let $P(c_j) = \frac{c_j}{N}$ be the probability of the uniform class distribution, the normalized entropy is given by the equation:

$$NE = \frac{\sum_{j=1}^{k} P(c_j) \log(\frac{P(c_j)}{1/k})}{2 \log k} \tag{1}$$

10. SC_{10}: "small" clusters. A measure of the number of classes with size inferior to the threshold $\theta = 10$. Its value is given by: $SC_\theta = \sum_{j=1}^{k} 1(c_j < \theta)/k$.
11. SC_{15}: same measure of previous item, but with its threshold set to $\theta = 15$.
12. BC: "big" clusters. A measure of the number of classes with size superior to the threshold $\theta = 50$, given by: $BC_\theta = \sum_{j=1}^{k} 1(c_j > \theta)/k$.
13. k-NN outliers: classification error obtained by the k-NN algorithm ($k = 3$) [19]. Another indicator of the quality of the data.

2.2 Meta-learner

We evaluated six algorithms as meta-learners: J48, PART, MLRules [20], Random Forest, k-Nearest Neighbors (k-NN) and also Support Vector Machines (SVM). With the exception of the SVM experiments, which were performed using the libSVM[1] package, all experiments were executed within the WEKA framework [2].

The J48 algorithm is the WEKA implementation of the C4.5 decision tree algorithm. The varied parameters were the confidence factor (from 2^{-15} to 2^{15}) and the minimum number of instances per leaf (from 2 to 7). Another method considered is the PART algorithm, which actually builds a partial C4.5 decision tree in each iteration, making the "best" leaf into a rule. For the experiments with PART, we used the same parameter values evaluated for the J48 algorithm.

The random forest algorithm belongs to the so-called "ensemble methods", a combination of various methods that generate many classifiers (in this case, decision trees) aggregating their results. This method has several features, which include the possibility to be used on a mixture of discrete and continuous descriptors, to classify binary or multi-class data sets and work with data sets where there are more variables than observations. The algorithm also presents good performance even when most predictive variables are noise [21]. For this work, we fixed the number of trees parameter to 100 and varied the number of attributes to be selected in each tree from 1 to 15. Another ensemble method evaluated was the Maximum likelihood rule ensembles (MLRules), which is a recent rule induction algorithm for solving classification problems via probability estimation. The ensemble is built minimizing the negative loglikelihood to estimate the class conditional probability distribution. We varied the minimization technique (Newton and gradient) and the shrinkage parameter in $\{0.1, 0.2, ..., 1\}$.

SVMs are supervised learning methods that construct a separating hyperplane in an n-dimensional space, trying to maximize the margin between the classes. We executed the experiments considering polynomial and RBF kernels. For polynomial kernel, we varied the cost and the degree parameters in the intervals $[2^{-15}, 2^{15}]$ and $[2, 6]$ respectively. For RBF kernel we varied both the cost and gamma parameters in the interval $[2^{-15}, 2^{15}]$. For the k-NN algorithm, we varied k in the interval $[1, 20]$.

Classification experiments were developed according to the leave-one-out procedure, with some remarks: as the class distribution of the meta-examples was

[1] http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

[2] http://www.cs.waikato.ac.nz/ml/weka/

unbalanced (class distribution is, respectively, 16 examples for FMG, 10 for KM and 4 for SP), this could led to overfitting towards the classes with larger number of examples (FMG and KM). So, in order to make class distribution more uniform, each example from the FMG, KM and SP classes was replicated 2, 3 and 8 times, respectively. This replication process was performed only in the training data. Thus, by doing so, an example would never be at the same time on the training and test sets. Rule extraction experiments were developed employing the same balanced data, except by the fact that we did not evaluate the accuracy of the obtained model using the leave one out procedure: instead, we utilized the full training set.

3 Results and Discussion

3.1 Meta-learner

The average test accuracy of the leave-one-out experiments realized with the five methods compared can be seen in Table **??**. According to this table, Random Forest obtained the best classification accuracy followed by MLRules. All other methods had a cross-validation accuracy equal or lower than the base line error (taking the majority class as reference). This is probably a consequence of the difficulty of the classification problem, as there are very few samples to classify (30 samples) and one of the classes (SP) has only four samples. Ensemble methods, like Random Forest, are often expected to have a better performance on such difficult classification scenarios, which is confirmed in our study.

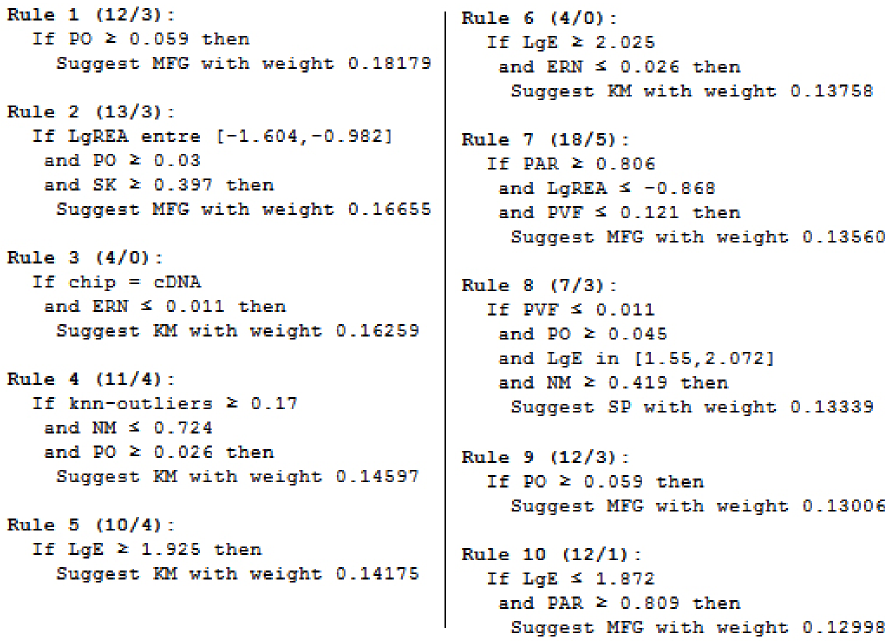
Table 2. Accuracy rates - runs over balanced meta-data

Method	Accuracy
PART	40.00%
J48	30.00%
MLRules	56.67%
k-NN	53.33%
SVM	53.33%
Random Forest	63.33%
Base Line Error	53.33%

3.2 Rule Mining

The next step in our analysis was to extract interpretable knowledge from the meta-learning learning analysis of the data. Our goal is to discover explanatory (partial) models of performance of clustering algorithms on cancer gene expression data. Observing the generated rules, one can notice the suitability of clustering algorithms studied, as well as the actual relations to characteristics (meta-attributes), with respect to the underlying structure in the data sets.

In order to do so, we used the MLRules algorithm with all data as training set and Newton steps as minimization technique and shrinkage to 0.5. A total of 100 rules were generated, but we selected only the ten rules with biggest weights



```
Rule 1 (12/3):
  If PO ≥ 0.059 then
    Suggest MFG with weight 0.18179

Rule 2 (13/3):
  If LgREA entre [-1.604,-0.982]
   and PO ≥ 0.03
   and SK ≥ 0.397 then
    Suggest MFG with weight 0.16655

Rule 3 (4/0):
  If chip = cDNA
   and ERN ≤ 0.011 then
    Suggest KM with weight 0.16259

Rule 4 (11/4):
  If knn-outliers ≥ 0.17
   and NM ≤ 0.724
   and PO ≥ 0.026 then
    Suggest KM with weight 0.14597

Rule 5 (10/4):
  If LgE ≥ 1.925 then
    Suggest KM with weight 0.14175

Rule 6 (4/0):
  If LgE ≥ 2.025
   and ERN ≤ 0.026 then
    Suggest KM with weight 0.13758

Rule 7 (18/5):
  If PAR ≥ 0.806
   and LgREA ≤ -0.868
   and PVF ≤ 0.121 then
    Suggest MFG with weight 0.13560

Rule 8 (7/3):
  If PVF ≤ 0.011
   and PO ≥ 0.045
   and LgE in [1.55,2.072]
   and NM ≥ 0.419 then
    Suggest SP with weight 0.13339

Rule 9 (12/3):
  If PO ≥ 0.059 then
    Suggest MFG with weight 0.13006

Rule 10 (12/1):
  If LgE ≤ 1.872
   and PAR ≥ 0.809 then
    Suggest MFG with weight 0.12998
```

Fig. 1. Rules induced by the MLRules algorithm

for analysis. The produced rules can be seen in Figure 1. They are listed in a pseudo-code like structure to ease readability.

Here, at each rule one can find, respectively, the method indicated (KM, FMG and SP), the number of meta-examples classified by the node and how many are misclassified (in parenthesis), as well as the rule weight. Interestingly to notice that, in general, rules that suggest the KM method involves the LgE and the ERN meta-attributes. This agrees with literature information, in which this method tends to find equal sized clusters (low ERN, rules 3 and 6) and is very sensitive to a small number of training patterns (low LgE, rules 5 and 6). We can also observe the presence of the meta-attribute PO (percentual of outliers) requiring bigger values in most of the rules that suggest the MFG method, an indication that this method presents good tolerance to datasets with a high number of outliers (rules 1, 2 and 9). Only one rule related to the Spectral algorithm was generated (rule 8), possibly due to the small number of examples labeled with this class available. The Spectral method employed in [18] is based in a Gaussian similarity function, which matches the requirements of data normality. This fact agrees with the assertive $MN > 0.419$ on rule 8. Furthermore, the rule suggests the use of the Spectral method in the presence of outliers.

Another interesting fact is the presence of chip type in rule 3. It is well known in the microarray literature that cDNA and Affymetrix chips generate expression values with distinct characteristics [22]. The cDNA arrays are based on log-ratios of the expression between the reference cell (tumor) and a control cell (healthy cell), whereas Affymetrix data is based only on the tumor cell and expression values should reflect the absolute count of transcripts in that cell. As a result, the log-ratios used in cDNA measurements make the expression values to have a normal distribution. Differently, Affymetrix expression values are positive and have a distribution skewed towards lower expression values. Furthermore, measurements of cDNA chips are less susceptible to probe problems in a specific chip, as a problematic probe will have the same effects to both control and reference values [22]. While there is no consensus in the microarray literature regarding the data quality and microarray platform, the cDNA chip type verification on rule 3 is another indication that data from cDNA microarrays are less sensitive to noise, suggesting the k-means method in this case.

The rules induced by MLRules could be susceptible to overfitting, as there are very few examples in the data sets. Nevertheless, as discussed in previous paragraphs, the rules extracted are in accordance to general knowledge in the clustering literature. Thus, rather than proposing the use of the rules and the attribute thresholds in their own, we interpret them as “soft” guidelines to the choice of a clustering method given a certain cancer gene expression data set.

4 Final Remarks

In this paper, we presented a preliminary study that explores the ability to automatically generate rules to guide the choice of clustering algorithms for gene expression data. One of the our main contributions is to show that two rule-based ensemble classifiers — random forests and MLRules — on average, presented the most accuracy rates in predicting the best clustering algorithm for gene expression data sets. We emphasize that the classification problem analyzed here is a difficult one, as there are very few meta-examples. Thus, no classification method had a high classification accuracy.

Another contribution of this work was to extract rules for the selection of clustering algorithms, by using an rule ensemble algorithm. Overall, the rules extracted give us some interesting guidelines for choosing the method. For instance, in the case of gene expression data from cDNA microarrays, k-means method should not be used when class size distribution is not uniform. Although, when a large number of samples is present, the method is preferred. Finite mixture of Gaussians should be used when there are few samples and a non-uniform class distribution. In cases where the data follows a normal distribution and there’s a large amount of outliers, Spectral clustering is adequate. Such guidelines, based on meta-attributes of data sets, had not been empirically demonstrated before in the gene expression literature. As a future work, we will try to increase the number of meta-examples, as well as investigate other meta-attributes.

Acknowledgments. The authors would like to thank CNPq (Brazilian Agency) for its financial support.

References

1. Golub, T., et al.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286(5439), 531–537 (1999)
2. Alizadeh, A.A., et al.: Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. Nature 403(6769), 503–511 (2000)
3. Spang, R.: Diagnostic signatures from microarrays: a bioinformatics concept for personalized medicine. Biosilico 1(2), 64–68 (2003)
4. Costa, I.G., et al.: Comparative analysis of clustering methods for gene expression time course data. Genetics and Molecular Biology 27(4), 623–631 (2004)
5. Datta, S., Datta, S.: Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes. BMC Bioinformatics 7, 397 (2006)
6. D'haeseleer, P.: How does gene expression clustering work? Nature Biotechnology 23(12), 1499–1501 (2005)
7. de Souto, M.C., et al.: Clustering cancer gene expression data: a comparative study. BMC Bioinformatics 9, 497 (2008)
8. Vilalta, R., et al.: Using meta-learning to support data- mining. Intern. Journal of Computer Science Application 1(31), 31–45 (2004)
9. Giraud-Carrier, C., et al.: Introduction to the special issue on meta-learning. Machine Learning 54(3), 187–193 (2004)
10. Smith-Miles, K.: Towards insightful algorithm selection for optimisation using meta-learning concepts. In: Proceedings of the IEEE International Joint Conference on Neural Networks 2008, pp. 4118–4124 (2008)
11. Brazdil, P., et al.: Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. Machine Learning 50(3), 251–277 (2003)
12. Kalousis, A., Gama, J., Hilario, M.: On data and algorithms - understanding inductive performance. Machine Learning 54(3), 275–312 (2004)
13. Pruděncio, R.B.C., Ludermir, T.B.: Meta-learning approaches to selecting time series models. Neurocomputing 61, 121–137 (2004)
14. Wang, X., et al.: Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series. Neurocomputing (2008) (to appear)
15. Tsoumakas, G., et al.: Lazy adaptive multicriteria planning. In: Proceedings of the 16th European Conference on Artificial Intelligence, ECAI 2004, pp. 693–697 (2004)
16. Smith-Miles, K.: Cross-disciplinary perspectives on meta-learning for algorithm selection. ACM Computing Surveys 41(1), 1–25 (2008)
17. Milligan, G., Cooper, M.: A study of standardization of variables in cluster analysis. Journal of Classification 5, 181–204 (1988)
18. de Souto, M.C.P., et al.: Ranking and selecting clustering algorithms using a meta-learning approach. In: Proceedings of the International Joint Conference on Neural Networks. IEEE Computer Society, Los Alamitos (2008)
19. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man and Cybernetics (3) (1972)
20. Dembczyński, K., Kotłowski, W., Słowiński, R.: Maximum likelihood rule ensembles. In: Proceedings of the 25th International Conference on Machine Learning, ICML, pp. 224–231 (2008)
21. Breiman, L.: Random forests. Machine Learning 45, 5–32 (2001)
22. Cauton, H.C., Quackenbush, J., Brazma, A.: Microarray Gene Expression Data Analysis: A Beginner's Guide. Blackwell Publishing, Malden (2003)