Information Sciences 224 (2013) 49–61

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Biomedic Organizations: An intelligent dynamic architecture for KDD

Juan F. De Paz [a,1], Javier Bajo [b,*], Vivian F. López [a,1], Juan M. Corchado [a,1]

[a] Departamento Informática y Automática, University of Salamanca, Plaza de la Merced s/n, 37008 Salamanca, Spain
[b] Facultad de Informática, Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, Campus Montegancedo, Boadilla del Monte, 28660 Madrid, Spain

ARTICLE INFO

Article history:
Received 17 November 2009
Received in revised form 12 October 2012
Accepted 16 October 2012
Available online 5 November 2012

Keywords:
Multi-agent system
Case-based reasoning
Microarray
Neural network
Case-based planning

ABSTRACT

The application of information technology in the field of biomedicine has become increasingly important over the last several years. This study presents the Intelligent Biomedic Organizations (IBOs) model, an intelligent dynamic architecture for knowledge discovery in biomedical databases. It involves an organizational model specially designed to support medical personnel in their daily tasks and to establish an innovative intelligent system to make classifications and predictions with huge volumes of information. IBO is based on a multi-agent architecture with Web service integration capability. The core of the system is a type of agent that integrates a novel strategy based on a case-based planning mechanism for automatic reorganization. This agent proposes a new reasoning agent model, where the complex processes are modeled as external services. In this sense, the agents act as coordinators of Web services that implement the four stages of the case-based planning cycle. The multi-agent system has been implemented in a real scenario to classify leukemia patients, and the classification strategy includes services such as a novel ESOINN neural network and statistical methods to analyze patient data. The results obtained are presented within this paper and demonstrate the effectiveness of the proposed organizational model.

1. Introduction

Cancer diagnosis is a field requiring novel automated solutions and tools and the ability to facilitate the early detection, even prediction, of cancerous patterns. The continuous growth of techniques for obtaining cancerous samples, specifically those using microarray technologies, provides a great amount of data. Microarray has become an essential tool in genomic research, making it possible to investigate global genes in all aspects of human disease [24]. Currently, there are several kinds of microarrays such as CGH arrays [28] and expression arrays [2]. Expression arrays contain information about thousands of genes in a patient's samples. The genes are represented by a series of probes that are composed of oligonucleotides. The number of oligonucleotides is up to one million per sample. The large amount of data requiring analysis makes it necessary to use data mining techniques in order to reduce processing time.

There are different approaches for decision support systems, including myGrid [30], which base their functionality on the creation of Web services that are implemented according to OGSA (Open Grid Services Architecture) [12]. The main disadvantage, however, is that the user must be the responsible for creating the sequence of actions which resolve specific problems. These systems provide methods for solving complex problems in a distributed way through SOA [11] and Grid

* Corresponding author. Tel.: +34 639771985.
E-mail addresses: fcofds@usal.es (J.F. De Paz), jbajo@fi.upm.es (J. Bajo), vivian@usal.es (V.F. López), corchado@usal.es (J.M. Corchado).
[1] Tel.: +34 923294400x1513.

0020-0255/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2012.10.031

architectures, but lack adaptation capabilities. However new research lines do exist which are focused on reasoning mechanisms with a high capacity for learning and adaptation, notably case-based reasoning (CBR) systems [20], which solve new problems by taking into account knowledge obtained in previous experiences [20] and integrating this information within agents and multi-agent systems. The disadvantage of CBR-based decision support systems for classification in medical databases is the high dimensionality of the data and its corresponding complexity. Multi-agent systems are an emerging alternative that provide distributed entities, called agents, with autonomous reasoning skills that facilitate the development of distributed applications. Moreover, some proposals provide the agents with special capabilities for learning by adapting CBP (case-based planning) mechanisms [15]. CBP–BDI (Belief, Desire, Intention) agents [15] make it possible to formalize systems by using a new planning mechanism that incorporates graph theory and Bayesian networks as a reasoning engine to generate plans, and to incorporate a novel automatic discovery and planning method based on previous plans. The proposed architecture represents an advance in biomedical databases, since it provides a novel method for the retrieval (filtering) of the data, as well as a new model for knowledge extraction that notably helps the human expert to have an understanding of the classification process. As opposed to other architectures used in KDD (as myGrid), IBO can perform automatic reorganizations by using a CBP–BDI planning model that provides the architecture with capacities for an automatic selection of the services.

Bayesian networks are widely used in bioinformatics. Some examples are the use of Bayesian networks to study the relations between genes and proteins [40], or genes and different pathologies [41]. It should be noted that for those cases where the number of variables is very high, new models of bayesian neural networks, such as the Dynamic Bayesian network, are defined. In the case studies presented in this paper, the number of variables is not very high and we do not work with a semi-infinite collection of variables; therefore, it is not necessary to use this kind of bayesian network. Apart from Dynamic Bayesian networks, we have taken into consideration the full Bayesian approach with Bayesian model averaging, which are used to provide better generalization in those cases where the available data is low. These kinds of networks can be observed in [42]. The bayesian networks presented in this paper are used to make predictions about the actions applied during the microarrays analysis and to automatically create workflows; the advances presented in [42] would be useful.

This paper presents IBO, an innovative solution to model decision support systems in biomedical environments, based on a multi-agent architecture which allows integration with Web services and incorporates a novel planning mechanism that makes it possible to determine workflows based on existing plans and previous results. IBO can simulate and analyze laboratory work processes and the behavior of the workers, facilitating any adaptation required when facing anomalous situations and predicting possible risks. In this way, it is possible to analyze large amounts of data from microarrays in a distributed way. The core of IBO is a CBP agent [15] specifically designed to act as Web services coordinator, making it possible to reduce the computational load for the agents in the organization and expedite the classification process.

The IBO model was applied to case studies consisting of the classification of leukemia patients and brain tumors from microarrays, while the multi-agent system developed incorporates novel strategies for data analysis and classification. The process of studying a microarray is called expression analysis [21] and consists of a series of phases: data collection, data pre-processing, statistical analysis, and biological interpretation. These analysis phases basically consist of three stages: normalization and filtering; clustering and classification; and knowledge extraction. In this study, a multi-agent system based on IBO architecture models the phases of expression analysis performed by laboratory workers, and incorporates both innovative algorithms implemented as Web services and filtering techniques based on statistical analysis, allowing a notable reduction of the data dimensionality and a classification technique based on an ESOINN [14] neural network. The core parts of the system are reasoning agents based on the CBP–BDI [15] mechanism.

The next section provides the specific problem description of microarray data analysis. Section 3 describes the main characteristics of the Intelligent Biomedic Organizations and briefly explains its components. Section 4 presents a case study consisting of a distributed multi-agent system for cancer detection scenarios developed using IBO. Finally Section 5 presents the results and conclusions obtained.

2. Microarray data analysis

Microarray has become an essential tool in genomic research, making it possible to investigate global gene expression in all aspects of human diseases. Microarray technology is based on a database of gene fragments called ESTs (Expressed Sequence Tags), which are used to measure target abundance using the scanned fluorescence intensities from tagged molecules hybridized to ESTs [22]. Specifically, the HG U133 plus 2.0 [2] are chips used for expression analysis. These chips analyze the expression level of over 47,000 transcripts and variants, including 38,500 well-characterized human genes. It consists of more than 54,000 probe sets and 1,300,000 distinct oligonucleotide features. The HG U133 plus 2.0 provides multiple, independent measurements for each transcript. The use of Multiple probes provides a complete data set with accurate, reliable, reproducible results from every experiment. Microarray technology is a critical element for genomic analysis and allows an in-depth study of molecular characterization of RNA expression, genomic changes, epigenetic modifications or protein/DNA unions.

Expression arrays [2] are a type of microarray that have been used in different approaches to identify the genes that characterize certain diseases [31,23]. In all cases, the data analysis process is essentially composed of three stages: normalization and filtering; clustering; and classification. The first step is critical to achieve both a good normalization of data and an initial filtering to reduce the dimensionality of the data set with which to work [3]. Since the problem at hand is working with

high-dimensional arrays, it is important to have a good pre-processing technique that can facilitate automatic decision-making about the variables that will be vital for the classification process. In light of these decisions it will be possible to reduce the original dataset.

Case-based reasoning [26] is particularly applicable to this problem domain because it (i) supports a rich and evolvable representation of experiences, problems, solutions and feedback; (ii) provides efficient and flexible ways to retrieve these experiences; and (iii) applies analogical reasoning to solve new problems [18]. CBR systems can be used to propose new solutions or evaluate solutions to avoid potential problems. The research in [1] suggests that analogical reasoning is particularly applicable to the biological domain, in part because biological systems are often homologous (rooted in evolution). In [4] a mixture of experts for case-based reasoning (MOE4CBR) is proposed. It is a method that combines an ensemble of CBR classifiers with spectral clustering and logistic regression, but does not incorporate extraction of knowledge techniques and does not focus on dimensionality reduction.

3. Intelligent Biomedic Organizations

IBO (Intelligent Biomedic Organizations) is an organizational model for biomedical environments based on a multi-agent dynamic architecture that incorporates agents with skills to generate plans for the analysis of large amounts of data. The core of IBO is a novel mechanism for the implementation of CBP mechanism stages through Web services. This mechanism provides a dynamic self-adaptive behavior in order to reorganize the environment. Moreover, IBO provides communication mechanisms that facilitate integration with SOA architectures.

IBO was initially designed to model laboratory environments oriented to the processing of data from expression arrays. To do this, IBO defined specific agent types and services. The agents act as coordinators and managers of services, while the

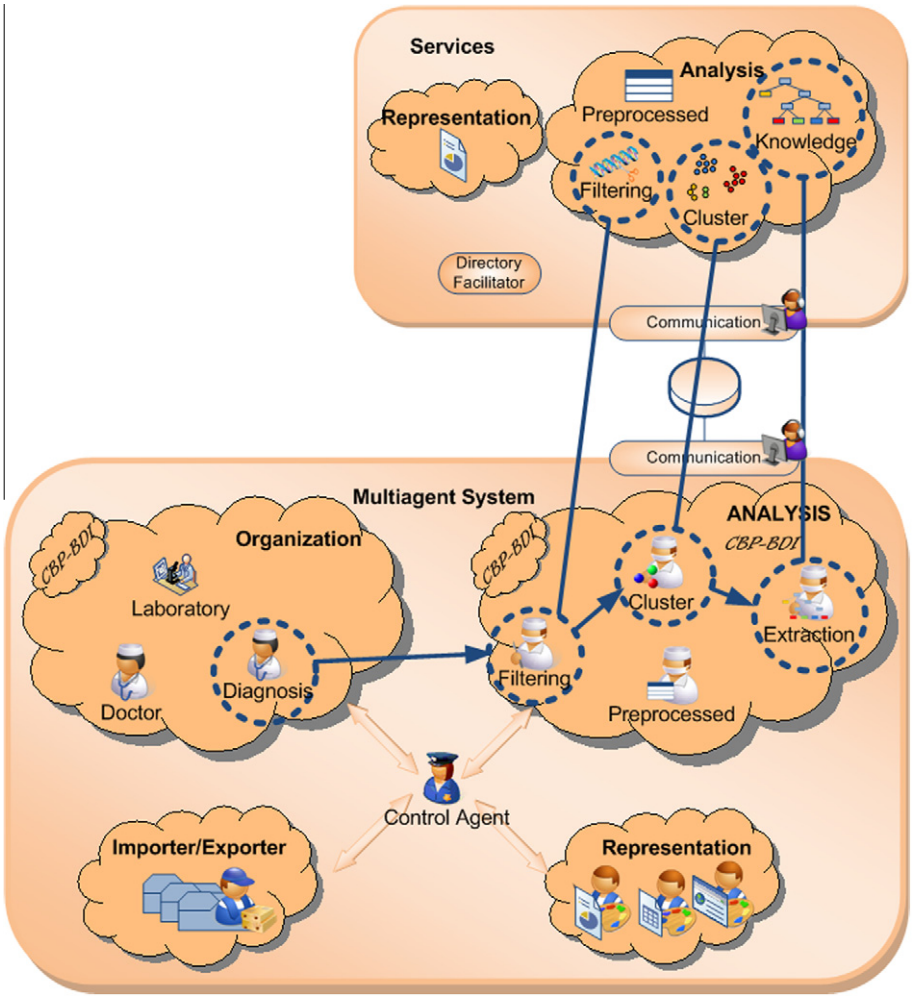

Fig. 1. IBO architecture.

services are responsible for carrying out the processing of information by providing replication features and modularity. The different types of agents are distributed in layers within the IBO according to their functionalities, thus providing an organizational structure that includes an analysis of the information and management of the organization, and making it possible to easily add and eliminate agents from the system.

The agent layers constitute the core of IBO and define a virtual organization for massive data analysis, as can be seen in Fig. 1. Fig. 1 shows four types of agent layers:

- Organization: The organization agents run on the user devices or on servers. The agents installed on the user devices create a bridge between the devices and the system agents which perform data analysis. The agents installed on servers will be responsible for conducting the analysis of information following the CBP–BDI [15] reasoning model. The agents from the organizational layer should be initially configured for the different types of analysis that will be performed.
- Analysis: The agents in the analysis layer are responsible for selecting the configuration and the flow of services best suited to the problem that needs to be solved. They communicate with Web services to generate results. The agents of this layer follow the CBP–BDI [15] reasoning model. The workflow and configuration of the services to be used is selected with a Bayesian network and graphs, using information that corresponds to the previously executed plans. The agents at this layer are highly adaptable to the case study to which the IBO is applied. Specifically, the microarray case study includes the agents that are required to carry out the expression analysis, as shown in Fig. 1.
- Representation: These agents are in charge of generating tables with classification data and graphics for the results.
- Import/Export: These agents are in charge of formatting the data in order to adjust them to the needs of agents and services.

The Controller agent manages the agents available in the different layers of the multi-agent system. It allows the registration of agents in the layers, as well as their use in the organization.

The services layer, as shown in the top part of Fig. 1, is divided into two groups:

- Analysis Services: Analysis services are services used by analysis agents for carrying out different tasks. Analysis services include services for pre-processing, filtering, clustering and extraction of knowledge. Fig. 1 illustrates how the analysis layer agents invoke these services in order to carry out the different tasks corresponding to microarray analysis.
- Representation Services: These services generate graphics and result tables.

As shown in Fig. 1, the agents from the different layers interact to generate the plan for the final analysis of the data. For example, in order to carry out its task, the Diagnosis agent at the organizational layer uses a specific sequence to select agents from the analysis layer. In turn, the analysis layer agents select the services that are necessary to carry out the data study, and the filtering agent at the analysis layer selects from the services and workflow that are appropriate for the data. Within the services layer, there is a service called Facilitator Directory that provides information on the various services available and manages the XML file for the UDDI (Universal Description Discovery and Integration).

3.1. Coordinator agent based on CBP–BDI

The agents in the organization layer and the agents in the analysis layer have the capacity to learn from the analysis carried out in previous procedures. To do so, they adopt the CBP reasoning model, a specialization of CBR [20]. Case-based planning (CBP) is the idea of planning as remembering [15]. In CBP, the solution proposed to solve a given problem is a plan, so this solution is generated by taking into account the plans applied to solve similar problems in the past. The problems and their corresponding plans are stored in a plans memory. The CBP–BDI agents stem from the BDI model and establish a correspondence between the elements from the BDI model and the CBP systems. The BDI model [5,9] adjusts to the system requirements since it is able to define a series of goals to be achieved based on the information that has been registered with regard to the world. The CBP–BDI agents make it possible to formalize the available information as beliefs, define the goals and actions available for solving the problem, and define the procedure for solving new problems by adopting the CBP reasoning cycle. The terminology used is the following:

The environment M and the changes that are produced within it are represented from the point of view of the agent. Therefore, the world can be defined as a set of variables that influence a problem faced by the agent

$$M = \{\tau_1, \tau_2, \cdots, \tau_s\} \text{ with } s < \infty \tag{1}$$

The beliefs are vectors of some (or all) of the attributes of the world taking a set of concrete values

$$B = \{b_i / b_i = \{\tau_1^i, \tau_2^i, \cdots, \tau_n^i\}, \quad n \leqslant s \quad \forall i \in N\}_{i \in N} \subseteq M \tag{2}$$

A state of the world $e_j \in E$ is represented for the agent by a set of beliefs that are true at a specific moment in time t.

Let $E = \{e_j\}_{j \in N}$ set the status of the World. If we fix the value of t, then

$$e_j^t = \left\{b_1^{jt}, b_2^{jt}, \cdots b_r^{jt}\right\}_{r \in N} \subseteq B \quad \forall j, t \tag{3}$$

The desires are imposed at the beginning and are applications between a state of the current world and another that it is trying to reach

$$d : \underset{e_0}{E} \underset{\rightarrow}{\rightarrow} \underset{e^*}{E} \tag{4}$$

Intentions are the way that the agent's knowledge is used in order to reach its objectives. A desire is attainable if the application i, defined through n believes in the existence of:

$$i : \underset{(b_1,\ b_2,\cdots,\ b_n,\ e_0)}{BxBx\overset{n)}{\cdots}xBxE} \underset{\rightarrow}{\rightarrow} \underset{e^*}{E} \tag{5}$$

In our model, intentions guarantee that there is enough knowledge in the beliefs base for a desire to be reached via a plan of action.

We define an agent action as the mechanism that provokes changes in the world making it change the state,

$$a_j : \underset{e_i}{E} \underset{\rightarrow}{\rightarrow} \underset{a_j(e_i)=e_j}{E} \tag{6}$$

Agent plan is the name we give to a sequence of actions that, from a current state e_0, defines the path of states through which the agent passes in order to reach the other world state.

$$\begin{aligned} &p_n : \underset{e_0}{E} \underset{\rightarrow}{\rightarrow} \underset{p_n(e_0)=e_n}{E} \\ &p_n(e_0) = e_n = a_n(e_{n-1}) = \cdots = (a_n \circ \cdots \circ a_1)(e_0) p_n \equiv a_n \circ \cdots \circ a_1 \end{aligned} \tag{7}$$

In IBO, services correspond to the actions that can be carried out and that determine the changes in the initial problem data. Each of the services is represented as a node in a graph, allowing each plan to be represented by a path in the graph. The presence of an arch that connects to a specific node implies the execution of a service associated with the end node. As a result, a sequence of nodes represents a sequence of actions/services and the order in which they are carried out, so that each plan identified in (7) can be represented as a route in a graph. Each of the nodes in the graph is associated with a set of variables with a corresponding value, thus forming a set of beliefs that describe each of the states of the graph. Additionally it is necessary to indicate that each of the nodes corresponding to the services is also included in a pair of fictitious nodes that correspond to the start and end nodes. The start and end nodes are necessary to establish the initial service of a plan, as well as to be able to establish the end point of a specific plan. Fig. 2 plan p_1 provides a graphical representation of a service plan. The path defines the sequence of services from the start node to the end node. The plan described in the graph is defined by the following sequence $(s_7 \circ s_5 \circ s_3 \circ s_1)(e_0)$. e_0 represents the original state that corresponds to Init, which represents the inicial problem description e_0. Final represents the final state of the problem e^*.

This way, a CBP–BDI agent works with plans that contain information associated with the actions that it should carry out, i.e., each analysis layer agent defines its own memory of plans with the information it needs. The information required for each of the agents at the analysis layer depends on the agent's functionality. Some agents require executable actions such as service compositions, while others only need to select the service that best suits its needs without having to carry out any composition. Table 1 provides an example of the description of the case structure for a filtering agent. As shown, a filtering agent will consider the number of cases, the number of variables and the optimization, quality and description of a problem.

Additionally, the information for the plans is defined by the sequence of actions/services applied. Finally, it is necessary to define the structure for each service. Table 2 shows the structure of a service that is defined by the number of variables, the name of the services, and a list of parameters used by the service.

CBP–BDI agents use the information contained in cases in order to perform different types of analyses. As previously explained, an analysis assumes the construction of the graph that will determine the sequence of services to be performed. The construction process for the graph can be broken down into a series of steps that are explained in detail in the following subsections (we will focus on one agent in particular within the analysis layer, specifically the filtering agent): (1) Extract the set of cases similar to the current case with the best and worst output, (2) Generate the directed graph with the information from the different plans, (3) Generate a TAN classifier for the cases with the best and worst output respectively, using the Friedman et al. [13] algorithm, (4) Calculate the execution probabilities for each service with respect to the classifier gener-

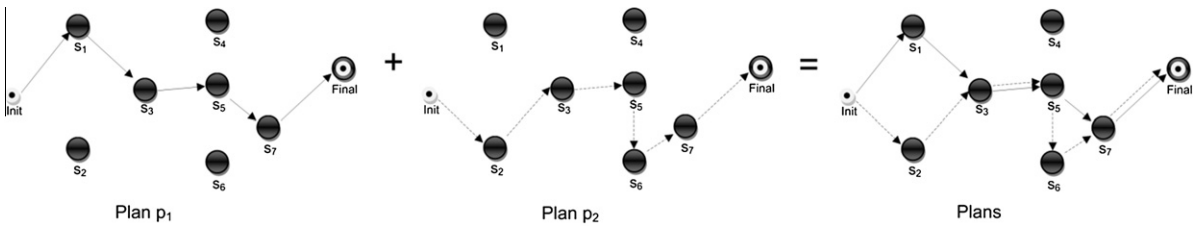

Fig. 2. Composition of graphs.

Table 1
Filtering agent case.

Variable	Field type	Variable	Field type
numberCases	Integer	Quality	Real
numberVariables	Integer	Problem	Integer
Optimization	Integer		

Table 2
Service.

Task field	Field type
numberVariablesFinal	Int
Service	String
Parameter	ArrayList of parameter

ated in the previous step, (5) Adjust the connections from the original graph according to a metric, and (6) Construct the graph.

3.1.1. Similar cases

The selection of the most similar cases is made by using the cosine distance [29]. The values are normalized given the previous calculation of distance, in order to avoid a dependency on the units and to be able to compare different measures. In the memory of cases shown in Tables 1 and 2, the cosine distance is calculated for the numberCases and numberVariables parameters. Moreover, the variables optimization and problem of the retrieved cases have to be equal to the variables optimization and problem of the new case.

3.1.2. Constructing a directed graph

The plans represented in graphical form are joined to generate one directed graph that makes it possible to define the new plans based on the minimization of a specific metric. That way, for example, given the graphs shown in Fig. 2, a new graph is generated, which joins the information corresponding to both graphs.

The new plans are generated through the construction of the graph of plans shown in Fig. 2. Each of the arcs in the plans graph has a corresponding weight with which it is possible to calculate the new route to be executed, which defines the plan obtained from the recovered plans. The weights are estimated based on the existing plans by applying a TAN classifier and the probabilities of execution of the services. The probabilities that a particular number of services may have been executed to classify the efficient and inefficient plans obtained with the TAN are combined with the probabilities of execution of the servicies to update the weigths. The TAN classifier provides a tree that takes into account two Bayesian networks. The entry data to the Bayesian networks is broken down into the following elements: Plans with a high efficiency are assigned to class 1, Plans with a low efficiency are assigned to class 0.

3.1.3. TAN classifier

The TAN classifier is constructed based on the plans recovered that are most similar to the current plan, distinguishing between efficient and inefficient plans to generate the model (the tree). Thus, by applying the Friedman et al. [13] algorithm, the two classes that are considered are those of efficient and inefficient. The Friedman–Goldsmidtz algorithm makes it possible to calculate a Bayesian network based on the dependent relationships established through a metric. The metric considers the dependent relationships between the variables according to the classifying variable. In this case, the classified variable is efficient and the remaining variables indicate whether a service is or is not available. The metric proposed by Friedman can be defined as:

$$I(X,Y|Z) = \sum_{x\in X}\sum_{y\in Y}\sum_{z\in Z} P(x,y,z)\cdot \log\left[\frac{P(x,y|z)}{P(x|z)\cdot P(y|z)}\right] \tag{8}$$

where X, and Y represents the services and Z the classes (low efficiency, high efficiency). Based on the previous metric, the probabilities P are estimated according to the frequencies of the data.

3.1.4. Probabilities of the services

Once the TAN model has been calculated for each of the classes, we proceed to calculate the probability of execution for each of the services. These probabilities consider the dependences between services and influence the final value of the weights assigned to the arcs in the graph. The probabilities are calculated taking the TAN model into account. Assuming that the set of random variables can be defined as follows $U = \{X_1, X_2, \ldots, X_n\}$, we can assume that the variables are independent. A Bayesian network for U is defined as a tuple formed by two elements $B = \langle G, T\rangle$ where G represents an acyclic directed graph in which the nodes are variables and the connections between the nodes for T contain the connection probabilities

between variables. The probabilities are represented by $P_B(x_i|\pi_{x_i})$ where x_i is a value of the variables X_i and $\pi_{x_i} \in \Pi_{X_i}$ where π_{x_i} represents one of the parents for the node X_i. Thus, a Bayesian network B, defines a single set probability distribution over U given for

$$P_B(X_1, X_2, \ldots, X_n) = P_B(X_n|X_{n-1}, \ldots, X_1) \cdot P(X_{n-1}, \ldots, X_1) = \prod_{i=1}^{n} P_B(X_n|X_{n-1}, \ldots, X_1) = \prod_{i=1}^{n} P_B(X_i|\Pi_{X_i})$$

3.1.5. Connection considerations

Using the TAN model, we can define the probability that a particular number of services may have been executed for classes 1 and 0 for the efficient and inefficient plans as explained in Section 3.1.2. This probability is used, together with the probability of execution, to determine the final value for the weight with regards to the quality of the plans recovered. Assuming that the probability of having executed service i for class c is defined as follows $P(i, c)$ the weight of the arcs is defined according to the following Eq. (9). The function has been defined in such a way that the plans of high quality are those with values closest to zero.

$$c_{ij} = P(j,1) \cdot I(i,j,1) \cdot t_{ij}^1 - P(j,0) \cdot I(i,j,0) \cdot t_{ij}^0 \tag{9}$$

$$t_{ij}^1 = \frac{\sum_{p \in G_{ij}^1, s \in G^1} (1 - (q(p) - \min(q(s)))) + 0.1}{\#G_{ij}^1} \quad t_{ij}^0 = \frac{\sum_{p \in G_{ij}^0, s \in G^0} q(p) - \min(q(s)) + 0.1}{\#G_{ij}^0} \tag{10}$$

where $I(i, j, 1)$ is the probability that service i for class 1 is executed before that of service j, and $P(j, 1)$ the probability that service j for class 1 is executed. The value is obtained based on the Bayesian network defined in the previous step. $I(i, j, 0)$ is the probability that service i for class 0 is executed before that of service j, and $P(j, 0)$ the probability that service j for class 0 is executed. The value is obtained based on the Bayesian network defined in the previous step. G_{ij}^s is the set of plans that contain an arc originating in j and ending in i for class s, G^s is the set of plans for class s, and $q(p)$ is the quality of plan p which also defined the execution time for the plan. The significance depends on the measure of optimization in the initial plan where $\#G_{ij}^s$ is the number of elements in the set, and c_{ij} is the weight for the connection between the start node j and the end node i.

3.1.6. Graph construction

Once the graph for the plans has been constructed, the minimal route going from the start node to the end node is calculated. In order to calculate the shortest/longest route, the Floyd algorithm is applied. The route defines the new plan to be executed and depends on the measure to maximize or minimize. Once the execution of the proposed plan finishes, the human expert evaluates the efficiency of the plan (efficient, non-efficient), and the evaluation is stored together with the results obtained. The graph is reconstructed each time a new analysis is performed. To do this, the stored plans data is taken into consideration. The time used to reconstruct the workflow is very low compared to the time used during the different steps of the analysis, so it can be considered as non-significative for the overall performance of the system.

4. Case study: Using IBO to develop a decision support for patient diagnosis

The IBO multi-agent architecture was used to develop a decision support system for the classification of leukemia, CLL leukemia and patients with brain tumors. This paper presents the results obtained for leukemia. The microarrays used in the case studies contain information that corresponds to the patients affected by leukemia and brain tumors. The data for leukemia patients was obtained with a HG U133 plus 2.0 chip and corresponded to 212 patients affected by five different types of leukemia (ALL, AML, CLL, CML, MDS) [10].

IBO was used to model the organizations corresponding to each case study, and a support system was provided for the decision based on obtaining a classification method and clustering patients, as well as a detection method for the patterns that characterize the different diseases for each patient. The aim of the tests performed is to determine whether the system is able to classify new patients based on the previous cases analyzed and stored. The developed agents and services are explained below.

4.1. Services layer

The services implement the algorithms that allow the analysis expression of the microarrays [21,10]. These services are invoked by the agents and present novel analysis techniques. The services are broken down into the categories that are necessary for performing expression analysis: preprocessed, filtered, cluster-classification, knowledge extraction.

4.1.1. Pre-processing service

This service implements the RMA algorithm and a novel control and errors technique. The RMA (*Robust Multi-array Average*) [17] algorithm is frequently used for pre-processing Affymetrix microarray data. RMA consists of three steps: (i) Background Correction; (ii) Quantile Normalization (the goal of which is to make the distribution of probe intensities the same for

arrays); and (iii) Expression Calculation. During the Control and Errors phase, all probes used for testing hybridization are eliminated. Occasionally, some of the measures made during hybridization may be erroneous; although this is not the case with the control variables. In this case, the erroneous probes that were marked during the RMA must be eliminated.

4.1.2. Filtering service

The filtering service eliminates the variables that do not allow classification of patients by reducing the dimensionality of the data. These services are used for filtering: **Variability**. The first stage is to remove the probes that have low variability according to the following steps: Calculate the standard deviation for each of the probes, standardize the high values, discard probes for which the value of z meet the following condition: $z < \alpha$. **Uniform Distribution**. All remaining variables that follow a uniform distribution are eliminated. The contrast of assumptions followed is explained below, using the Kolmogorov–Smirnov [7] test. **Correlations**. The linear correlation index of Pearson is calculated and correlated variables are removed so that only the independent variables remain. **Cufoff points**. Delete those probes which do not have significative changes in the density of individuals.

4.1.3. Clustering service

This addresses both the clustering and the association of a new individual with the most appropriate group. The services included in this layer are: the ESOINN [14] neural network. Additional services in this layer for clustering are the Partition around medoids (PAM) [27] and dendrograms [19].

The classification is carried out bearing in mind the similarity of the new case using the NN cluster and the SVM (Support Vector Machine) [35]. The similarity measure used is as follows:

$$d(n,m) = \sum_{i=1}^{s} f(x_{ni}, x_{mi}) * w_i \tag{11}$$

where s is the total number variables, n and m the cases, w_i the value obtained in the uniform test, and f the Minkowski [16] Distance that is given for the following equation:

$$f(x,y) = \sqrt[p]{\sum\nolimits_i |x_i - y_j|^p} \text{ with } x_i, \quad y_j \in R^p \tag{12}$$

This dissimilarity measure weights the probes that have the least uniform distribution, since these variables do not allow a separation.

4.1.4. Knowledge extraction service

The knowledge extraction technique applied was the CART (Classification and Regression Tree) [6] algorithm. The CART algorithm is a non-parametric test that can extract rules to explain the classification carried out. There are others techniques to generate the decision trees, such as methods based on ID3 trees [25], although the results can be considered very similar to those provided by CART.

4.2. Agent layer

The agents in the analysis layer implement the CBP–BDI reasoning model with which they select the flow for services delivery and decide the value of different parameters based on previously made plans. A measure of efficiency is defined for each of the agents to determine the best course of recovery for each phase of the analysis process.

In the preprocess stage of the analysis layer, only one service is available, so the agent only has to select the settings. The efficiency is calculated by the deviation in the microarray once it has been preprocessed. At the filtering stage, the efficiency of plan p is calculated by the relationship between the proportion of probes and the resulting proportion of individuals falling ill.

$$e(p) = \frac{s}{N} + \frac{i'}{I} \tag{13}$$

where s is the final number of variables, N is the initial number of probes, i' the number of misclassified individuals and I the total number of individuals. In the clustering and classification phases the efficiency is determined by the number of misclassified individuals. Finally, during the knowledge extraction process the CART technique was implemented, together with alternative extraction of knowledge techniques. Efficiency is determined by the number of misclassified individuals.

5. Results and discussion

IBO was applied to three different case studies and a number of tests were carried out in each one. We will present the results obtained for one of the case studies. The tests were oriented to evaluating both the efficiency and the adaptability of the approach. In the following paragraphs we present the specific results obtained and extract the subsequent conclusions, which are presented in Section 6.

The first experiment consisted of evaluating the services distribution system in the filtering agent for the case study that classified patients affected by different types of leukemia. The first step that was carried out was to incorporate each of the services based on the plans. According to the identification of the problem described in Table 1 and the algorithm in Section 3.1.1, the filtering agent will select the plans with the greatest efficiency, considering the different execution workflows for the services in the plans. Table 3 shows the efficiency obtained for the service workflows obtained using the equation, (13) which provided the best results in previous experiences. The values in the table indicate the application sequence for the services within the plan. A blank cell indicates that a service is not invoked for that specific plan.

Based on the plans shown in Table 3, a new plan is generated following the procedures indicated in Section 3.1. Once the new plan has been generated, it is necessary to establish the configuration parameters for the services that can define the signification levels for the statistical tests and other parameters that are used to carry out the filtering process. Table 4 shows the possible service configurations for the new plan generated. The bold text indicates the configuration with optimal efficiency. The filtering agent selects the values that have provided better results based on the measure of the previously established efficiency Eq. (13).

Once the service distribution process and the selection of parameters for a specific case study were evaluated, it seemed convenient to evaluate the adaption of this mechanism to case studies of a different nature. To do so, we once again recovered the plans with the greatest efficiency for the different workflows and case studies, and proceeded to calculate the Bayesian network and the set of probabilities associated with the execution of services as mentioned in Sections 3.1.3 and 3.1.4. Once the graph plans were generated, a more efficient plan was generated according to the procedures indicated in Section 3.1.6, with which we can obtain the plan that best adjusts to the data analysis.

The filtering agent uses the data presented in Tables 3 and 4 to select the execution workflow that is best adapted to the case study. The results obtained for the different node connections, following the procedure of Section 3.1.6, are shown in Table 5. The subscripts S_{ij} shown in Table 5 correspond to the connection between services. For example, S_{01}, means that service 0 is executed first and is then followed by service 1.

IBO uses the information shown in Table 5 to construct the plans graph. The graph information is used to generate the new plan, so that the maximum route that links the first and last node is constructed. Fig. 3 shows the graph and the final route followed in bold. As can be seen in Fig. 3, the plan selected is the plan formed by the sequence S_{01}, S_{12}, S_{23}, S_{3f}. This plan does not match with any of the plans previously created in Table 3. When implementing the plan, the final efficiency obtained was 0.1277, which slightly improves the efficiency of plan $p1$.

Table 6 shows the plans generated by the filtering process that best adjusts to the case study in which the system was applied.

Once the plans were generated, the evolution of the system was evaluated for each of the case studies. Fig. 4 shows the evolution of the efficiency according to Eq. (11). This measure of efficiency was selected because it provides a global measure of the results from an expression analysis. As shown, the efficiency improved as the system acquired efficiency over time.

Table 3
Efficiency of the plans.

Variability (z)	Uniform (α)	Correlation (α)	Cutoff	Efficiency	Class
1	2	3	4	0.1401	1
1	2		3	0.1482	1
1				0.1972	0
	1		2	0.1462	1
		1		0.2036	0
	1			0.1862	0
			1	0.1932	0
		1	2	0.186	0

Table 4
Plans of the filtering phase and plan of greater efficiency.

Variability (z)	Uniform (α)	Correlation (α)	Probes	Errors	Efficiency
−1.0	0.25	0.95	2675	21	0.1485
−1.0	0.15	0.90	1341	23	0.1333
−1.0	0.15	0.95	1373	24	0.1386
−0.5	0.15	0.90	1263	24	0.1365
−0.5	0.15	0.95	1340	23	0.1333
−1.0	**0.1**	**0.95**	**785**	**24**	**0.1277**
−1.0	0.05	0.90	353	32	0.1574
−1.0	0.05	0.95	357	34	0.1669
−0.5	0.05	0.9	332	47	0.2278
−0.5	0.05	0.95	337	53	0.2562
−1.0	0.01	0.95	54	76	0.3594

Table 5
Connection weights.

Arc	Weight	Arc	Weight
S_{01}	0.02554452	S_{23}	0.03337883
S_{02}	0.00380053	S_{24}	0.00860609
S_{03}	−0.02555357	S_{2f}	−0.00032338
S_{04}	−0.00035525	S_{34}	0.02107333
S_{12}	0.04152417	S_{3f}	0.02460639
S_{1f}	−0.01069132	S_{4f}	−0.00094166

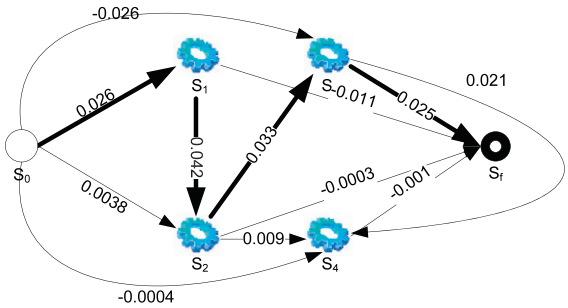

Fig. 3. Plan directed graph.

Table 6
Efficiency of the plans.

Case study	Variability (z)	Uniform (α)	Correlation (α)	**Cutoff**	Efficiency
Leukemia	1	2	3		0.1277

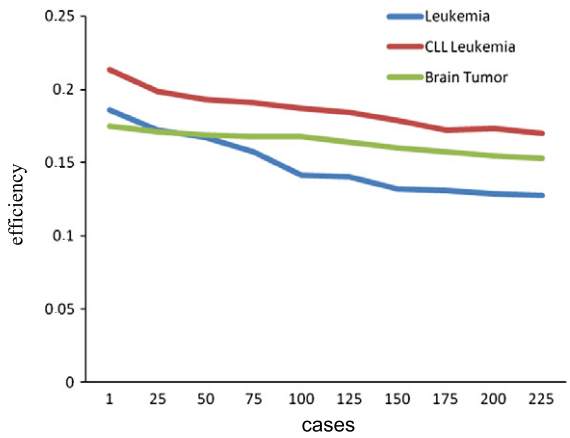

Fig. 4. Evolution of performance based on the filtering metric for the different case studies.

The improvement in the case study with the greatest number of individuals is the most significant. This is partly due to the functioning of the actual CBP–BDIs, which improve their return output with the number of cases. Fig. 4 shows the number of plans previously carried out, and the great efficiency reached by the same plan in the specific number of plans carried out. The *X* axis represents the number of cases, and the *Y* axis represents the efficiency.

Once the filtering phase has finished, the next stage in expression analysis consisted of grouping the individuals. Evaluation of this stage's efficiency is much easier than in the case of the filtering phase since it is associated with the erroneous classification. Selection of the most efficient technique for grouping individuals is very simple as only the most efficient task is selected. Following with the example, in the absence of any previous case with concatenations of services, no workflow with these characteristics is created. The reason is that the plan graph only possesses paths with three nodes, where two

nodes correspond to the first and last node. In other studies, the analysis of different methods was implemented in order to carry out the cluster phase [34].

For the knowledge extraction phase, something similar to the cluster phase occurs. In this stage the CART, OneR [36], DecisionStump [38] and JRIP [37] techniques were applied. The number of hits returned in the classification obtained by applying rules is shown in Table 7. The total number of cases for this sample study is 212. The efficiency is measured by the error rate so that the lower the rate of error, the higher the efficiency.

To analyze the success rate the elements were classified one by one, extracting each of the cases and constructing the models using the rest of the cases. Four different techniques were applied, obtaining the following success rates: 89.2% CART, 93.4% SVM, 76.42% k-neighbours [39], 59.4% OneR [36]. The success rate for SVM is higher than CART, but SVM does not allow knowledge extraction and it is not possible to identify the relevant probes associated to the classification process, making it difficult to extract knowledge and to analyze the data.

In a similar way the results obtained by IBO were compared to alternative methods without the filtering phase. The use of all the cases available provided the following results: For the CART and OneR methods it was necessary to cancel the execution due to complexity of the processing. SVM provided 95.76% success rate and k-neighbors 76.89% success rate. The results obtained are more accurate, but the objective of IBO is to provide knowledge extraction capacities.

Starting with the information collected with CART, it is possible to present the predictions made by probes for different individuals. Fig. 5 shows a graphical representation of the different individuals and the predictions made. At the top of Fig. 5, a bar for each one of the individuals is shown. The bars that are divided into many segments as probes are available for analysis. The width of each segment corresponds to the luminescence values of the individuals. At the top, two segments in two colors are shown. One represents the actual type of leukemia. At the bottom, the classified type is shown. At the bottom of Fig. 5 there are some parallel coordinates, each one of them representing a probe. The individuals are represented by lines in the color associated with the group. When the human expert selects a set of individuals, the selection values of the parallel coordinates are automatically adjusted and we can see that these individuals are separated from the rest by means of the selected probes, as the rest of the individuals remain deselected.

Table 7
Hits classification using knowledge extraction rules.

Technique	Correct	Efficiency	Class
CART	209	0.014150943	1
OneR	158	0.254716981	0
DecisionStump	137	0.353773585	0
JRIP	198	0.066037736	1

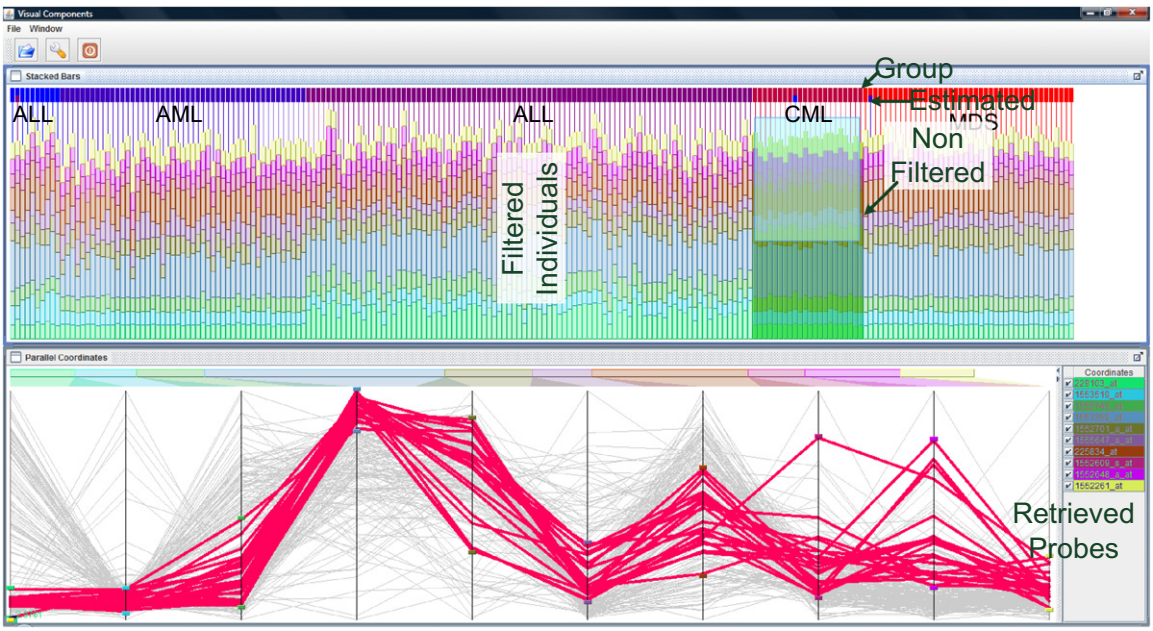

Fig. 5. Representation of classification probes for the five types of leukemia.

Traditionally, in statistics, an analysis like that of Kruskal–Wallis [10] or ANOVA [32] is applied to distinguish the characteristics that differentiate the groups. This article compares IBO and traditional statistic techniques, with to the aim of evaluating the level of improvement introduced in our proposal. Specifically, we focus on the Case Study to classify the five types of leukemia. In this case study, applying Kruskal–Wallis for the selection of probes which differentiate the groups obtained a total of 47,461 relevant probes, while applying ANOVA recovered a total of 45,924. It should be noted that ANOVA was applied despite the fact that the variables do not comply with the normality hypothesis. Thus the result can be considered as insignificant, although it is used in many works [32,33]. As the obtained results demonstrate, traditional statistical analysis cannot be applied to a satisfactory level for the selection of relevant probes. More advanced techniques are necessary, like those proposed in IBO. Furthermore, it must be taken into account that both ANOVA and Kruskal–Wallis can only be applied when groupings are provided, something that is not always available; this is presumably a major handicap with respect to the data filtered in IBO. A more detailed analysis of the techniques integrated in the different phases of the CBR cycle can be seen in [10]. Furthermore, applying ANOVA and Kruskal–Wallis to the 785 filtered variables for the case study of the five types of Leukemia, can eliminate 41 and 40 variables respectively. As can be appreciated, these techniques are also insufficient for the selection of relevant information as the reduction of dimensionality is insignificant.

6. Conclusions

This study has presented the IBO multi-agent architecture and its application to real problems. IBO facilitates task automation by means of intelligent agents capable of autonomously planning the stages of an expression analysis. The characteristics of this novel architecture facilitate an organizational-oriented approach where the dynamics of a real scenario can be captured and modeled into CBP–BDI agents. The agents act as controllers and coordinators of the organization. The complex functionalities of the agents are modeled as distributed services. IBO provides a service oriented approach and facilitates the distribution and management of resources. Moreover, IBO facilitates the distributed execution of complex computational services, reducing the number of crashes in agents. The multi-agent system developed is integrated within Web services, aims to reduce the dimensionality of the original data set, and proposes a novel method of clustering for classifying patients. The multi-agent perspective allows the system to work in a way similar to how human specialists operate in the laboratory, but is able to work with great amounts of data and make decisions automatically, thus reducing significantly both the time required to make a prediction, and the rate of human error arising from confusion. The Bayesian networks make it possible to generate the plans that best adjust to the different case studies, allowing the generation of new plans based on existing information without actually needing an existing memory of plans. If we follow the same procedure as the one established for selecting the parameters, it would be necessary to have extensive memory plans and the definition of a mechanism for carrying out the composition of efficient plans and generating a new and more efficient plan.

The multi agent system simulates the behavior of experts working in a laboratory, making it possible to carry out a data analysis in a distributed manner, as normally done by experts. The system is capable of learning based on previous experiences and of generating new behaviors and, as a result, creates an application that adapts to the information that characterizes the case studies.

Acknowledgements

Special thanks to the Institute of Cancer of Salamanca for the information and technology provided. This work was supported by the Spanish Ministry of Science TIN 2009-13839-C03-03 Project.

References

[1] J.S. Aaronson, H. Juergen, G.C. Overton, Knowledge discovery in GENBANK, in: Proceedings of the First International Conference on Intelligent Systems for Molecular Biology, AAAI Press, 1993, pp. 3–11.
[2] Affymetrix. <http://www.affymetrix.com/support/technical/datasheets/hgu133arrays_datasheet.pdf>.
[3] N.J. Armstrong, M.A. Van de Wiel, Microarray data analysis: from hypotheses to conclusions using gene expression data, Cellular Oncology 26 (5-6) (2004) 279–290.
[4] N. Arshadi, I. Jurisica, Data mining for case-based reasoning in high-dimensional biological domains, IEEE Transactions on Knowledge and Data Engineering 17 (8) (2005) 1127–1137.
[5] M.E. Bratman, Intention, Plans and Practical Reason, Harvard U.P., Cambridge, 1988.
[6] L. Breiman, J. Friedman, A. Olshen, C. Stone, Classification and Regression Trees, Wadsworth International Group, Belmont, California, 1984.
[7] R. Brunelli, Histogram analysis for image retrieval, Pattern Recognition 34 (2001) 1625–1637.
[8] J.M. Corchado, R. Laza, Constructing deliberative agents with case-based reasoning technology, International Journal of Intelligent Systems 18 (12) (2003) 1227–1241.
[9] J.M. Corchado, J.F. De Paz, S. Rodríguez, J. Bajo, Model of experts for decision support in the diagnosis of leukemia patients, Artificial Intelligence in Medicine 46 (3) (2009) 179–200.
[10] T. Erl, Service-Oriented Architecture (SOA): concepts, technology, and design, Prentice Hall PTR, 2005.
[11] I. Foster, C. Kesselman, J. Nick, S. Tuecke, The Physiology of the Grid: An Open Grid Services Architecture For Distributed Systems Integration, Technical Report of the Global Grid Forum, 2002.
[12] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Machine Learning 29 (1997) 131–163.
[13] S. Furao, T. Ogura, O. Hasegawa, An enhanced self-organizing incremental neural network for online unsupervised learning, Neural Networks 20 (2007) 893–903.
[14] M. Glez-Bedia, J. Corchado, A planning strategy based on variational calculus for deliberative agents, Computing and Information Systems Journal 10 (1) (2002) 2–14.

[15] P.J.F. Groenen, K. Jajuga, Fuzzy clustering with squared Minkowski distances, Fuzzy Sets and Systems 2 (2) (2001) 227–237.
[16] R. Irizarry, B. Hobbs, F. Collin, Y. Beazer-Barclay, K. Antonellis, U. Scherf, T. Speed, Exploration, normalization, and summaries of high density oligonucleotide array probe level data, Biostatistics 4 (2003) 249–264.
[17] I. Jurisica, J. Glasgow, Applications of case-based reasoning in molecular biology, Artificial Intelligence Magazine 25 (1) (2004) 85–95.
[18] L. Kaufman, P. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, Wiley Series in Probability and Statistics, 1990.
[19] J. Kolodner, Case-Based Reasoning, Morgan Kaufmann, 1993.
[20] E. Lander, Initial sequencing and analysis of the human genome, Nature 409 (2001) 860–921.
[21] R.J. Lipshutz, S.P.A. Fodor, T.R. Gingeras, D.H. Lockhart, High density synthetic oligonucleotide arrays, Nature Genetics 21 (1) (1999) 20–24.
[22] O. Margalit, R. Somech, N. Amariglio, G. Rechav, Microarray based gene expression profiling of hematologic malignancies: basic concepts and clinical applications, Blood Reviews 19 (4) (2005) 223–234.
[23] J. Quackenbush, Computational analysis of microarray data, Nature Review Genetics 2 (6) (2001) 418–427.
[24] J. Quinlan, Discovering rules by induction from large collections of examples, in: D. Michie (Ed.), Expert Systems in the Micro Electronic Age, Edinburgh University Press, Edinburgh, 1979, pp. 168–201.
[25] F. Riverola, F. Díaz, J.M. Corchado, Gene-CBR: a case-based reasoning tool for cancer diagnosis using microarray datasets, Computational Intelligence 22 (3-4) (2006) 254–268.
[26] N. Saitou, M. Nie, The neighbor-joining method: a new method for reconstructing phylogenetic trees, Molecular Biology and Evolution 4 (1987) 406–425.
[27] M. Shinawi, S.W. Cheung, The array CGHnext term and its clinical applications, Drug Discovery Today 13 (17-18) (2008) 760–770.
[28] M.W. Sohn, Distance and cosine measures of niche overlap, Social Networks 23 (2) (2001) 141–165.
[29] R. Stevens, R. McEntire, C. Goble, M. Greenwood, J. Zhao, A. Wipat, P. Li, myGrid and the drug discovery process, Drug Discovery Today 2 (4) (2004) 140–148.
[30] M. Taniguchi, L.L. Guan, J.A. Basara, M.V. Dodson, S.S. Moore, Comparative analysis on gene expression profiles in cattle subcutaneous fat tissues, Comparative Biochemistry and Physiology Part D: Genomics and Proteomics 3 (4) (2008) 251–256.
[31] P. Pavlidis, Using ANOVA for gene selection from microarray studies of the nervous system, Methods 31 (4) (2003) 282–289.
[32] J.R. De Haan, S. Bauerschmidt, R.C. van Schaik, E. Piek, L.M.C. Buydens, R. Wehrens, Robust ANOVA for microarray data, Chemometrics and Intelligent Laboratory Systems 98 (1) (2009) 38–44.
[33] J. Bajo, J.F. De Paz, S. Rodríguez, A. Gonzalez, A new clustering algorithm applying a hierarchical method neural network, Logic Journal of the IGPL 19 (2) (2012) 304–314.
[34] V.N. Vapnik, An overview of statistical learning theory, IEEE Transactions on Neural Networks. 10 (1999) 988–999.
[35] R.C. Holte, Very simple classification rules perform well on most commonly used datasets, Machine Learning 11 (1993) 63–91.
[36] W.W. Cohen, Fast effective rule induction, in: Twelfth International Conference on Machine Learning, 1995, pp. 115–123.
[37] X.C. Yin, C.P. Liu, Z. Han, Feature combination using boosting, Pattern Recognition Letters 26 (2005) 2195–2205.
[38] D. Aha, D. Kibler, Instance-based learning algorithms, Machine Learning 6 (1991) 37–66.
[39] R. Chang, R. Shoemaker, W. Wang, A novel knowledge-driven systems biology approach for phenotype prediction upon genetic intervention, IEEE/ACM Transactions on Computational Biology and Bioinformatics 8 (5) (2011) 1170–1182.
[40] R. Chang, M. Stetter, W. Brauer, Quantitative inference by qualitative semantic knowledge mining with bayesian model averaging, IEEE Transactions on Knowledge and Data Engineering 20 (12) (2008) 1587–1600.
[41] R. Chang, W. Brauer, M. Stetter, Modeling semantics of inconsistent qualitative knowledge for quantitative Bayesian network inference, Neural Networks 21 (2-3) (2008) 182–192.