Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Sciences 176 (2006) 2771–2790
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

MDSM: Microarray database schema matching using the Hungarian method

Yi-Ping Phoebe Chen [a,b,*], Supawan Promparmote [a], Frederic Maire [c]

[a] *Faculty of Science and Technology, School of Information Technology, Deakin University, 221 Burwood Highway, Melbourne, Vic. 3125, Australia*
[b] *Australia Research Council, Centre in Bioinformatics, Australia*
[c] *Centre for Information Technology Innovation, Faculty of Information Technology, School of Software Engineering and Data Communications, Queensland University of Technology, Australia*

Received 7 April 2004; received in revised form 21 November 2005; accepted 29 November 2005

Abstract

Current microarray databases use different terminologies and structures and thereby limit the sharing of data and collating of results between laboratories. Consequently, an effective integrated microarray data model is required. One important process to develop such an integrated database is schema matching. In this paper, we propose an effective schema matching approach called MDSM, to syntactically and semantically map attributes of different microarray schemas. The contribution from this work will be used later to create microarray global schemas. Since microarray data is complex, we use microarray ontology to improve the measuring accuracy of the similarity between attributes. The similarity relations can be represented as weighted bipartite graphs. We determine the best schema matching by computing the optimal matching in a bipartite graph using the Hungarian optimisation method. Experimental results show that

* Corresponding author. Address: Faculty of Science and Technology, School of Information Technology, Deakin University, 221 Burwood Highway, Melbourne, Vic. 3125, Australia.
E-mail address: phoebe@deakin.edu.au (Y.-P.P. Chen).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2005.11.015

our schema matching approach is effective and flexible to use in different kinds of database models such as; database schema, XML schema, and web site map. Finally, a case study on an existing public microarray schema is carried out using the proposed method.

Keywords: Microarray database schema; Schema matching; Hungarian method; Similarity function

1. Introduction

Traditionally, molecular biology experiments were based on one gene at a time; this was a limitation in obtaining the total picture of a gene function. With the advent of DNA microarray technology, researchers are able to gain a better understanding of the interactions among thousands of genes simultaneously. Such technological innovation has led to new insights into fundamental biological problems such as; gene discovery, gene regulation, disease diagnosis, drug discovery, and toxicology [9–11,23,24].

However, a biological experiment, typically, requires tens or hundreds of microarray, where a single microarray generates between 100,000 and a million fragments of data [9–11]. The organisation of such a huge-volume of data, produced by microarray techniques, is one of the biggest challenges that scientists in bioinformatics are facing. Only a limited number of efficient and public databases are available to store microarray data (http://www.cbil.upenn.edu/RAD2, http://genex.sourceforge.net/, http://staffa.wi.mit.edu/chipdb/public/, http://www.ebi.ac.uk/arrayexpress/, http://genome-www5.stanford.edu/); however, existing public microarray databases have their own distinct storage structures and implementations, and different hardware platforms, DBMS, data models and data languages. In addition, these databases are created by different developers; unavoidably they might use different definitions and terms to describe the same domain or concept. Even though there are efforts to develop microarray data resources that correspond to the standard Microarray Gene Expression Data (MGED) ontology (http://www.cbil.upenn.edu/Ontology/MGED_ontology.html), their databases are still not in final shape. As a result, this hampers the sharing of data with other laboratories and the collating of experimental results. Fortunately, these limitations have been previously addressed in fields outside the life sciences, particularly in the realm of commercial business. One successful approach to elucidate these limitations is database integration.

An integrated microarray database has been proposed in our previous work [16]. One important task in our integrated architecture is to create global microarray schema. This can be done by taking schemas as input to produce

a map between schema elements that correspond semantically to each other; this process is simply called schema matching. Schema matching has been investigated by many researchers [1,3–5,7,8,12,15,17,19,25]. Currently, schema matching is typically performed manually, which is a tedious, time-consuming, error-prone, and expensive process. This aggravates the problem since databases and applications are becoming more complex. That is, the larger the schemas are, the more the number of matches to be performed increases. Therefore, a faster and less labor-intensive integration approach is desirable.

Moreover, most existing systems are not generic as they support a limited number of data models and applications. For example, LSD [4] is limited to XML, and DIKE [11] is limited to ER sources. Schema matching is expected to be applied to many different data models and applications, such as database schema, XML schema, UML model, and website map [7].

The goal of this paper is to develop an effective Microarray Database Schema Matching (MDSM), using a combinatorial optimisation called Hungarian method [22,13]. To address this complex problem and deal with the large variety of microarray data models and applications, MDSM is designed to (1) be a fast and semi-automatic approach, (2) reconcile the structures and terminologies of the two microarray schemas, and (3) support generic models and applications. A case study of public microarray schemas RAD (http://www.cbil.upenn.edu/RAD2) and GeneX (http://genex.sourceforge.net/) is undertaken to prove that our approach is flexible and pragmatic. To our knowledge, this work is the first application of combinatorial optimisation to schema matching.

The structure of this paper is organised as follows. An overview of MDSM is described in Section 2. Section 3 describes the formalisation of problem. Section 4 explains how the Hungarian computes optimal matching. Section 5 describes experiments with MDSM on real microarray schemas. Section 6 presents the experimental evaluation and comparative discussion on MDSM with other systems. Finally, Section 7 concludes this paper.

2. Overall approach

This section provides an overview of the MDSM approach. MDSM consists of two main parts: *Attribute–attribute scoring*, and *Schema–schema scoring*. Each part is explained below.

2.1. Attribute–attribute scoring

A specific domain in our study is microarray database schema, which is much more complex than business domain—not only in the types of data

stored, but also in terms of richness and the constraints working upon relationships between those data. Because of its complex nature, additional information is desirable to specify the similarity of attributes between two different schemas. In our domain, MGED microarray ontology is used as supplementary knowledge.

To guarantee that MDSM can be applied to different kinds of data models, we define that a schema is simply a finite set of attributes, for instance, schema $X = \{x_1, x_2, \ldots, x_m\}$ and schema $Y = \{y_1, y_2, \ldots, y_n\}$.

The mapping results that can be denoted as $O_x = \{o_{x_1}, o_{x_2}, \ldots, o_{x_m}\}$ and $O_y = \{o_{y_1}, o_{y_2}, \ldots, o_{y_n}\}$, and are a number of ontological elements that semantically correspond to a number of attributes in schema X and Y, respectively. Each element of O_x links every element of O_y, with individual scores, and vice versa. In other words, each attribute of X links every attribute of Y, with unique scores. These mapping results can be represented as a weighted bipartite graph, in which elements of O_x and O_y correspond to nodes, links between O_x and O_y correspond to edges and individual scores correspond to weights (w_{ij}). Fig. 1 demonstrates the mapping results in a weighted bipartite graph.

2.2. Schema–schema scoring

The schema–schema matching score identifies how well two schemas correspond to each other. The score is calculated by the sum of every best attribute–attribute matching score in those two schemas.

By repeating this procedure on every pair-wise schema of two different databases, we can achieve similarity matrix, M that contains the similarity scores between different schemas. This similarity matrix, M will be beneficial for integrating schemas to subsequently develop a microarray global schema. Note that in this investigation, we only target the schema matching approach.

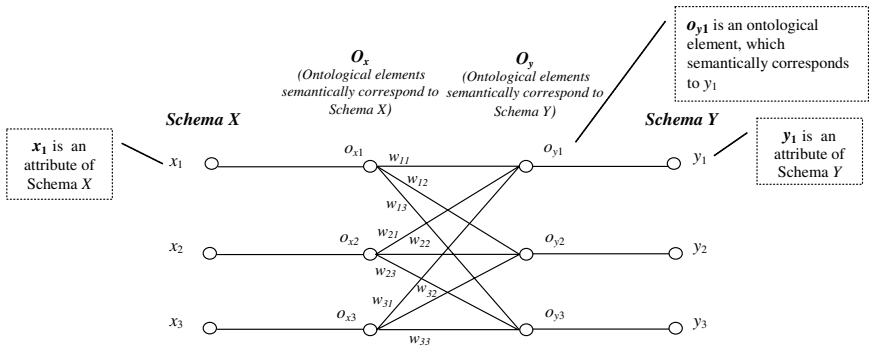

Fig. 1. A weighted bipartite graph that represents the mapping results.

3. Formalisation of problem

Based on the approach discussed in Section 2, problems can be characterised by obtaining the individual attribute–attribute matching scores and finding the best attribute–attribute matching candidates. The following two major issues are discussed in this section:

- the similarity function (to produce the attribute–attribute matching score),
- optimal matching (to find the best attribute–attribute matching).

3.1. Similarity function

The similarity function assigns real values to every link between elements of ontology O_x and O_y (or actually, between attributes of schema X and Y). Those values indicate how well two attributes relate to each other. Our attribute–attribute matching score is the average of syntactic and semantic similarities between two elements as shown in Eq. (1):

$$Sim(a,b) = \frac{Sim_{syn}(a,b) + Sim_{sem}(a,b)}{2} \tag{1}$$

where $Sim_{syn}(a,b)$ and $Sim_{sem}(a,b)$ are the syntactic and semantic similarities, respectively.

$Sim_{syn}(a,b)$ is a function that determines a probability for the syntactic similarities between elements on the basis of their name [6,16,20]. Here, we have used the n-grams based string matching technique to measure this syntactic possibility. The text strings are decomposed into n-grams, which are the contiguous characters of text strings. For example, Di-grams represent two characters in length and Tri-grams represent three characters. Basically, the probability of similarity between two strings is a proportion of the number of similar n-grams and the total number of unique n-grams in the strings. Consequently, a syntactic similarity $Sim_{syn}(a,b)$ can be defined as

$$Sim_{syn}(a,b) = \frac{|2 \times \sum_{t \in n\text{-}grams(a) \cap n\text{-}grams(b)} \log P(t)|}{|\sum_{t \in n\text{-}grams(a)} \log P(t)| + |\sum_{t \in n\text{-}grams(b)} \log P(t)|} \tag{2}$$

where $n\text{-}grams(a)$ and $n\text{-}grams(b)$ are the set of n-grams in a and b, respectively. $P(t)$ is the probability of a n-grams occurring in a word.

$Sim_{sem}(a,b)$ is a similarity measurement which computes the semantic distance between elements within a single ontology [6,14,20,21]. A single ontology can be represented as a graph-based model in which the elements are the nodes and the links between two elements are the edges. The semantic distance between two elements is the shortest linking path between them. We define the semantic similarity $Sim_{sem}(a,b)$ as

$$Sim_{sem}(a,b) = \frac{2 \times N_3}{N_1 + N_2 + 2 \times N_3} \tag{3}$$

where N_1 and N_2 are the numbers of links from a and b, respectively, to their most specific common superclass C; and N_3 is a number of links from C to the root of the ontology (''MGEDOntology'' in our case). Fig. 2 is a fragment of microarray ontology. The hierarchical illustration shows 20 classes (represented in rectangular shape) and six individuals (represented in oval shape).

Example. The similarity score for two elements, namely, *Organism* and *NCBI_taxon_id*, can be obtained as follows. Typically, a comparison of two elements must be performed on the elements with the same type. In this situation, types of *Organism* and *NCBI_taxon_id* are different: one is class and another is individual. If the first element (*Organism*) is a base, the second element (*NCBI_taxon_id*) must refer to the class which it belongs to. Since *NCBI_taxon_id* individual is an instance of *Organism* class, it is self-comparison between *Organism* classes.

Tri-grams (*Organism*) is $\{Org, rga, gan, ani, nis, ism\}$. Using Eq. (2), the syntactic similarity between *Organism* and *NCBI_taxon_id* ($Sim_{syn}(Organism, NCBI\_taxon\_id)$) evaluates to 1.

$$Sim_{syn}(Organism, NCBI\_taxon\_id) = \frac{2 \times \left|6 \times \log\frac{1}{6}\right|}{\left|6 \times \log\frac{1}{6}\right| + \left|6 \times \log\frac{1}{6}\right|} = 1 \tag{4}$$

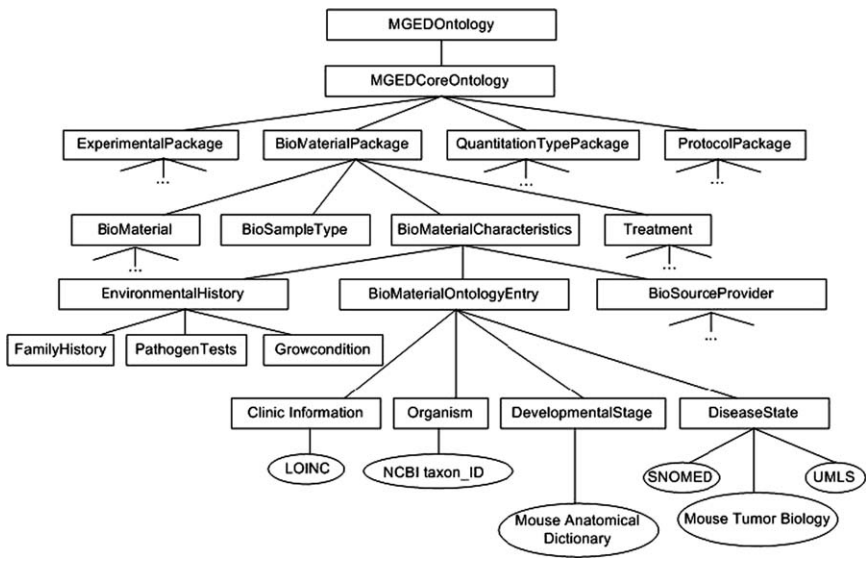

Fig. 2. A partial microarray ontology and a root ''MGEDOntology''.

From Eq. (3), the semantic similarity between *Organism* and *NCBI_taxon_id* can be obtained:

$$Sim_{sem}(Organism, NCBI\_taxon\_id) = \frac{2 \times 4}{1 + 1 + 2 \times 4} = 0.8 \tag{5}$$

where N_3 (= 4) is a number of links from *BioMaterialOntologyEntry* to *MGEDOntology*, N_1 (= 1) is a number of links from *Organism* to *BioMaterialOntologyEntry* and N_2 (= 1) is a number of links from *Organism* (a class that *NCBI_taxon_id* belongs to) to *BioMaterialOntologyEntry*.

The similarity score between *Organism* class and *NCBI_taxon_id* individual can be calculated as follows:

$$Sim(Organism, NCBI\_taxon\_id) = \frac{0.8 + 1}{2} = 0.9 \tag{6}$$

3.2. Optimal matching

As illustrated in Fig. 1, the element o_{x_1} links to elements o_{y_1}, o_{y_2} and o_{y_3}, with score w_{11}, w_{12}, and w_{13}. The best pair-wise match for element o_{x_1} is the element o_{y_i} such that w_{1i} is maximum. In other words, the attribute x_1 relates to attribute y_i more than the others, with score w_{1i}.

The existing microarray databases are made up of a large number of schemas, for example, GeneX consists of 30 schemas in their database model. It would be error-prone and laborious to match those schema elements manually. Enumerating all possible matching does not scale well with the size of the bipartite graph, as the number of candidate matchings is exponential in the number of vertices of the bipartite graph. In Section 4, we review different approaches to the bipartite graph matching problem and show that the method known as *the Hungarian method* presents a number of advantages.

4. Maximal weight matching

Given a bipartite graph, $G = (O_x, O_y, O_x \times O_y)$ where O_x and O_y are a finite set of *nodes* and $O_x \times O_y$ is a set of unordered pairs of nodes called *edges*. A matching in a graph G is a set of edges, where no two of which are incident to the same node. A *maximum matching* is a matching such that the sum of the weights of its edges is maximum.

A quick way to build a matching is to start with an empty set of edge M and incrementally add the largest edge e to M that leaves $M \cup \{e\}$ a matching. This greedy approach is fast and simple, but unfortunately does not guarantee the return of a maximum matching.

The following theorem [1, p. 286], based on work by Egervary done in 1931, relates the matching problem to linear programming. Let A be the vertex–edge

incidence matrix of the graph G. Let w be the weight vector and x the characteristic vector of the matching. That is, $x_i = 1$ if the ith edge belongs to the matching (otherwise $x_i = 0$).

Theorem (Egervary). *The optima in the linear duality programming duality equation*

$$\max\{w^{\mathrm{T}}x | x \geqslant 0, Ax \leqslant 1\} = \min\{y^{\mathrm{T}}1 | y \geqslant 0, y^{\mathrm{T}}A \geqslant w^{\mathrm{T}}\}$$

are attained by integer vector x and y.

In other words, if one has access to an optimisation library that contains a linear program solver, the maximum weighted matching can be solved by finding the solution in x to the optimisation problem $\max\{w^{\mathrm{T}}x | x \geqslant 0, Ax \leqslant 1\}$.

The particular form of the matrix A (binary matrix with exactly two ones per column), means that the optimisation problem $\max\{w^{\mathrm{T}}x | x \geqslant 0, Ax \leqslant 1\}$ can be solved using a purely combinatorial method. This algorithm is known as *The Hungarian method.* The proof of the validity of this algorithm is based on the Egervary theorem. The interested reader is referred to Schrijver's text [22] for the full theoretical derivation. Here, we simply explain the predominant ideas behind the algorithm. Firstly, we consider the special case where the weights are binary values, then we will consider the general case (weights take any non-negative values).

4.1. Binary valued weights case

In this case, the construction of a maximum matching can be done incrementally by searching for *augmenting paths*. Whenever an augmenting path is found, we can improve the current matching. Let M be a matching, and P be a path in a graph G. A path P is said to be an augmenting path if, and only if;

(1) The beginning and end nodes of P are not in M, and
(2) P is a sequence of edges alternately not in M and in M.

An example of an augmenting path is shown in Fig. 3.

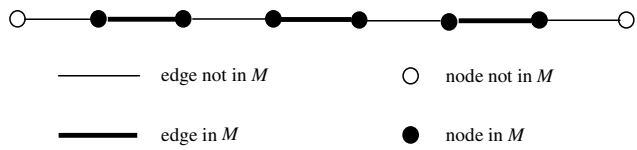

Fig. 3. An augmenting path.

Given a matching M and an augmenting path P, a better matching $M' = (M - P) \cup (P - M)$ can be constructed. It is easy to show that M' has one more edge than M. That is, $|M'| = |M| + 1$. In other words, if M is a maximum matching, no augmenting path exists. The reciprocal is also true.

Theorem 1. *M is a maximum matching in a graph if, and only if, augmenting path exists.*

4.2. General case

The algorithm for the general case (non-negative weights) is an extension of the augmenting path trick. We start with a matching $M = \emptyset$. Given a matching M, we build an auxiliary graph D_M that will allow us to derive from M, a new matching M' with a larger total weight.

Let D_M be the directed graph obtained from G, by orienting each edge e in G according to the following rules:

- If $e \in M$, then set the length l_e of e in D_M to $l_e = w_e$.
- If $e \notin M$, then set the length l_e of e in D_M to $l_e = -w_e$.

Let us call F_x the nodes of O_x not incident to any edge of M. Similarly, F_y denotes the nodes of O_y not incident to any edge of M. If there exists a path in D_M from F_x to F_y, we determine P to be the shortest such path, then reset the current matching to $M' = (M - P) \cup (P - M)$. We repeat this process of building the auxiliary graph D_M, searching for the shortest path P from F_x to F_y, resetting M if the search is successful and continue to do so until no such path P can be found. At that point, we are guaranteed to have an optimal match.

According to [1], the time complexity of this algorithm is $O(n(m + n\log n))$, where n is the number of vertices and m is the number of edges of G.

5. An example of using MDSM on existing microarray database schema

This section provides an example of using MDSM on the fragment schemas from public microarray databases, such as GeneX and RAD. Assume that attributes from two example schemas correspond to ontological elements in Fig. 4, and that they link to each other with scores in Table 1. Consider the problem in a weighted bipartite graph $G = (O_x, O_y, O_x \times O_y)$. The ontological elements of GeneX attributes are a set of nodes O_x, ontological elements of RAD attributes are a set of nodes O_y nodes, links between those nodes are edges, and weights of edges are scores.

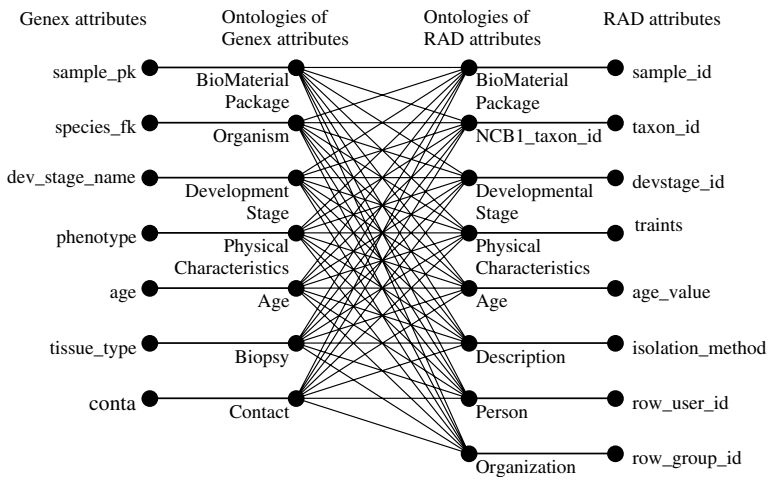

Fig. 4. The ontological elements that correspond to attributes from GeneX and RAD schema.

Table 1
An example of similarity scores that show how well two elements correspond to each other

	Bio-Material package	NCBI_ taxon_id	Develop-mental stage	Physical charac-teristics	Age	Descrip-tion	Person	Organi-sation
BioMaterial package	0.9	0.14	0.14	0.33	0.33	0.125	0.165	0.165
Organism	0.14	0.9	0.4	0.1	0.1	0.27	0.1	0.1
Development_ stage	0.33	0.1	0.47	0.4	0.4	0.1	0.18	0.18
Physical characteristics	0.33	0.1	0.21	0.9	0.4	0.1	0.18	0.18
Age	0.33	0.1	0.21	0.4	0.9	0.1	0.18	0.18
Biopsy	0.33	0.1	0.21	0.4	0.4	0.1	0.18	0.18
Contact	0.2	0.125	0.175	0.18	0.18	0.11	0.3	0.3

Fig. 4 represents the initial graph. A matching M is illustrated in Fig. 5(c). Only nodes that are not incident to any matched edges will be added to a matching M. It can be seen from Fig. 5(c), node *Biopsy* is not in a matching M. Node *Biopsy* cannot be matched to node *Physical Characteristics* or node *Age* because both nodes have already matched with the others. M is not a maximum matching; therefore, searching for an augmenting path is the next step. However, when no augmenting path that corresponds to M is found, the dual variables u_i and v_j must be changed. Dual variables will be modified four times. Here, we only show the final new values of variables u_i and v_j as depicted in Fig. 5(d). One rebuilds the auxiliary graph G_A by using new values of u_i and

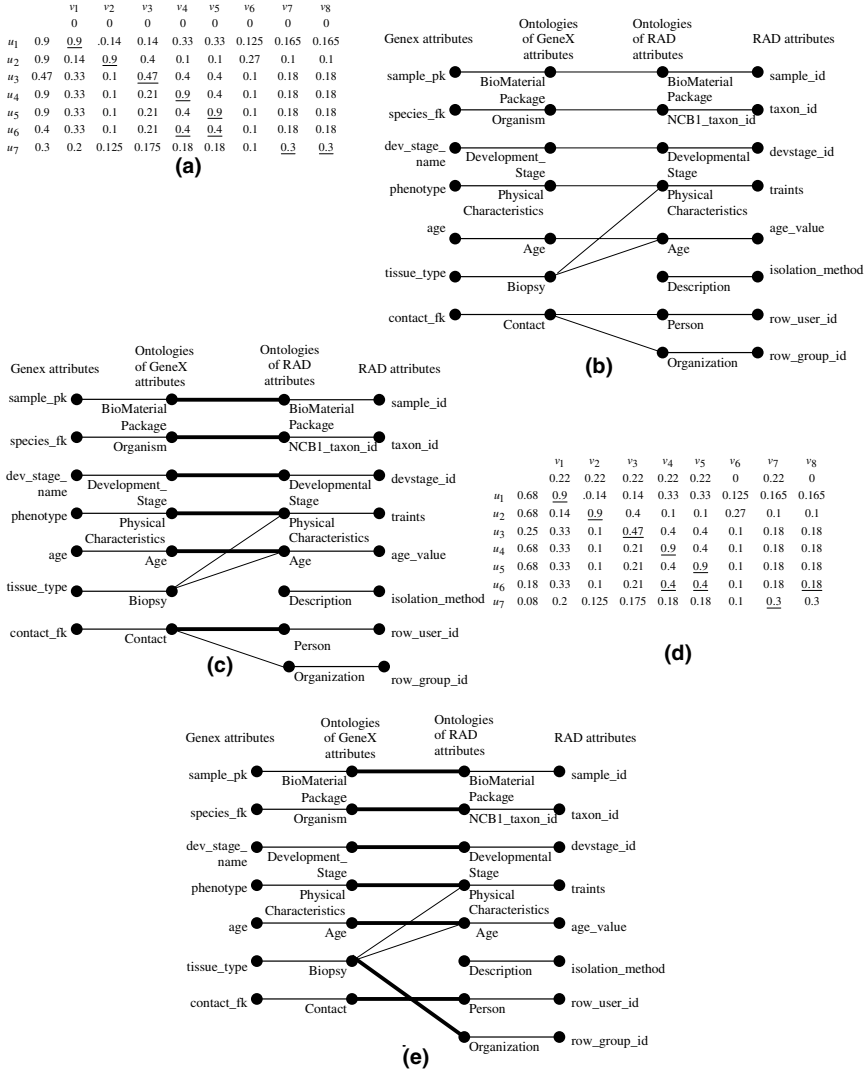

Fig. 5. An example of using the Hungarian method to achieve maximal weighted matching in a bipartite graph.

v_j. The new edge $\{biopsy, organisation\}$ is discovered. Fig. 5(e) shows a new matching M in G_A. Node *Biopsy* is now in a matching M. Every node in O_x is placed in M; therefore, M is a maximum matching. The sum of all weights in a matching M is optimal total weight, which is 4.55. This implies that two fragment schemas of GeneX and RAD correspond to each other with score 4.55.

6. Experimental evaluation

6.1. Test data

Data for this experiment was taken from the real-world schemas of five different microarray resources (http://www.cbil.upenn.edu/RAD2, http://genex.sourceforge.net/, http://staffa.wi.mit.edu/chipdb/public/, http://www.ebi.ac.uk/arrayexpress/, http://genome-www5.stanford.edu/). Five relations were extracted and mapped into the MGED microarray ontology which is used as an additional dictionary. The matching between the possible combinations of those five relations was performed.

6.2. Performance measures

Here, the word "performance" is defined as a set of correct mapping to pairs of schema attributes. In order to measure the MDSM performance, we used three common measures, *precision*, *recall*, and *overall* based on the bounded area A, B, C and D as shown in Fig. 6.

- $precision = \frac{|C|}{|B|+|C|}$ specifies the ratio of real correspondences among derived matches discovered by the matching algorithm.
- $recall = \frac{|C|}{|A|+|C|}$ specifies the ratio of real correspondences among true matches based on manual matching.
- $overall = recall * (2 - \frac{1}{precision})$ measures the overall quality of the matching algorithm as functions of both precision and recall. Unlike *precision and recall*, *overall* value can be negative if $precision < 0.5$, or the number of *false positives* is more than the number of *True.*

6.3. Analysis of results

Fig. 7 shows the Match Quality of the MDSM algorithm. The measures were determined for both single match experiments and the entire evaluation;

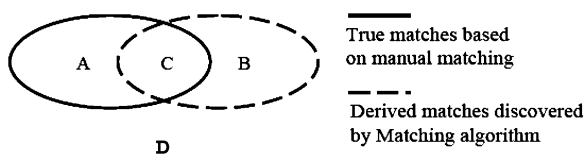

Fig. 6. A or *False Negatives* are matches needed but not automatically discovered, B or *false positives* are matches not needed but discovered by matching algorithm, C or *true positives* are matches discovered by both manual matching and the matching algorithm, and D or *True Negatives* are false matches.

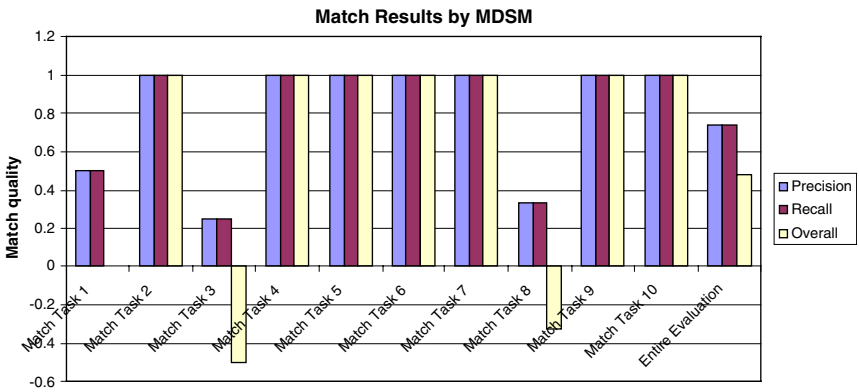

Fig. 7. Match quality of MDSM.

the false/true negatives and positives were counted over all match tasks. The quality measures for each single match experiment allow us to evaluate the actual performance of the matching algorithm and to directly compare the effectiveness of the system to other *positives* [3,6].

We tested MDSM with five existing microarray schemas, each of the five data sources were first manually matched to MGED ontology (an additional dictionary). Matches among data sources were then automatically performed, resulting in 10 match tasks altogether. Like other matching systems, the tested schema sizes were rather small; our source schemas consisted of elements between 34 and 47.

It is clear that MDSM performed very well. For single match evaluation, we achieved the highest match accuracy in match task 2, 4, 5, 6, 7, 9, and 10. Similarly, for the entire evaluation we achieved quite high *precision* (0.74) and *recall* (0.74). The *precision* value specifies that 74% of matches derived from MDSM correspond to manual matches derived from the user. Similarly, the *recall* value has identified that 74% of matches that were based on manual matching correspond to matches derived from MDSM. We achieved the Overall value of 0.481. Overall quality can also be determined by $\frac{|C|-|B|}{|A|+|C|}$. That is, the overall value can be minimised depending on the number of derived matches discovered by the algorithm which are not in manual matches.

The interesting results are in match task 1, 3, and 8. We further studied the impact on these match quality values. One hypothesis was drawn from [2]. They expressed that match quality would degrade with bigger schemas. However, the results from our experiment did not support their assumption. The schema size of match task 6, used for this experiment, was smaller than that of match task 1 and 3; however, the match quality was found to be higher than that of match task 1 and 3.

The above contradiction inspired us to thoroughly investigate the related factors that affect our match quality. The factors are as follows:

(1) The manual matches derived from one user might be different from another user. The scenario is similar as the algorithm is created by one person, while the manual matches are performed by another. Of course, there are a number of mismatches among manual and derived matches.
(2) Without considering other maximum similarities, MDSM tried to match the first element of schema X to an element of schema Y, based on its maximal similarity. This may also cause the mismatches between manual and derived matches. Consider the following scenario that shows the mismatch results among manual and discovered matching. The results based on manual matching show that element x_1 matches to y_3, x_2 to y_2, and x_3 to y_1. Whereas, the matching based on MDSM discovers that element x_1 matches to y_2, x_2 to y_3, and x_3 to y_1.

The manual matching from user

		Schema Y		
		y_1	y_2	y_3
Schema X	x_1	0.5	0.8	<u>0.7</u>
	x_2	0.4	<u>0.9</u>	0.2
	x_3	<u>0.9</u>	0.7	0.6

The derived matching from MDSM

		Schema Y		
		y_1	y_2	y_3
Schema X	x_1	0.5	<u>0.8</u>	0.7
	x_2	0.4	0.9	<u>0.2</u>
	x_3	<u>0.9</u>	0.7	0.6

6.4. Comparative discussion

In this section, we briefly compare MDSM with two other schema matching approaches, namely, Cupid and SemMa. The reason why we selected these systems is that Cupid has been a widely studied matching approach and SemMa is the most recent work produced within the literature. To compare those matching algorithms, we use a schema matching benchmark as summarised in [2,18]. Table 2 shows a summary of the key aspects and evaluations of Cupid, SemMa and MDSM. Since MDSM test problem came from a domain that is completely different from that of both Cupid and SemMa systems, it is difficult to compare their results.

While MDSM was tested with five different data models and applications were taken from different microarray data sources for performance, Cupid and SemMa was tested with only one data model. The capability of the last two algorithms to serve as generic schema matching, has thus been brought to scrutiny. Currently, the SemMa program does not support schema formats

Table 2
Summary of characteristics and evaluations of Cupid, SemMa, and MDSM

	Cupid	SemMa	MDSM
References	[8]	[25]	–
Test problems			
Tested schema types	XML	Relational	XML, UML, relational, ER
#Tested schemas/ #Match tasks	2/1	2/2	5/10
Min/Max/Avg schema similarity	–	–	0.1/0.9/0.5
Match performance			
Metadata representation	Extended ER	Relational	Graph
Schema-level match			
Name-based	Name equality; synonyms; homonyms; hypernyms; abbreviations	Name, token equality, synonyms, hyponyms, abbreviations	Name equality; synonyms; homonyms; hypernyms; abbreviations
Constraint-based	Data type and domain compability, referential constraints	Data type and referential constraints	Is-a (inclusion); Relationship
Structure matching	Matching subtrees, weighted by leaves	Table and field similarity	–
Reuse/auxiliary information used	Thesauri, glossaries	Database thesauras and WordNet	MGED microarray ontology
Combination of matchers	Hybrid	Hybrid	Hybrid matchers
Application area	Data translation, but intended to be generic	Schema integration	Schema integration
Match result representation			
Matches	Element and structure level correspondence with similarity value in range $[0,1]$	Element and structure level correspondence with similarity value in range $[0,1]$	Element and linguistic level with auxiliary information correspondence with similarity value in range $[0,1]$
Output format	Links with similarity values	Links with similarity values	Links show the matching nodes (attributes)
Local/global cardinality	1:1/*n*:1	1:1	1:1

(*continued on next page*)

Table 2 (*continued*)

	Cupid	SemMa	MDSM
Quality measure and test methodology			
Employed quality measures	By looking correspondences elements	Recall, precision, and overall	Precision, recall, overall
Subjectivity		1 user	
Pre-match effort	Specifying domain synonyms	Specifying field name, structure and data type	Specifying domains synonyms, homonyms and abbreviations
Best average match quality			
Precision	–	~0.81 to ~0.875	~0.74
Recall	–	~0.315 to ~0.845	~0.74
Overall	–	~0.23 to ~0.655	~0.481
Implementation			
Programming language	VB	C++	Matlab
Remarks			
	Tree matching	–	Algorithms to generate all possible mapping

other than BizTalk (http://msdn.microsoft.com/library/enus/bts_2002/htm/lat_xmltools_editor_intro_cyvg.asp) formatted XML schema. Like Cupid (elements ranging from 40 to 54) and SemMa (elements ranging from 40 to 47), the tested schema size of MDSM was rather small, ranging from 34 to 47 elements. In MDSM, the average similarity between the schemas was around 0.5; this implies that the schemas are much different even though they are from the same domain (microarray databases).

All of those systems initially determine the similarity of attributes from two schemas in a pair-wise fashion, based on element and structure level. With the availability of auxiliary information, the match quality can greatly improve. The similarity values are in the range of $[0, 1]$. The independent schemas were matched directly to each other in Cupid and MDSM, but not in SemMa. In SemMa, the independent schemas were matched to a single global schema. In all systems, the match result consists of attribute correspondence of 1:1 local and global cardinality.

To assess the automatic effort of the matching algorithm, both the *pre-match* and *post-match* effort should be taken into account. Cupid and MDSM systems require specifying domains, synonyms, homonyms and abbreviations before matchers can perform their tasks, whereas SemMa requires the specifying of

field name, structure and data type. The *post-match* efforts have so far not been taken into consideration when evaluating the match quality of systems.

Cupid is the only one system that did not provide a computed quality measure, it measures the matching accuracy by looking at correspondences between elements. SemMa and MDSM used three measures to test match quality: *precision*, *recall*, and *overall*. Even though match quality of SemMa seems to be higher than that of MDSM, SemMa used only two sample relational schemas to test the match quality. Besides, the match quality of SemMa varies according to the scale of schemas, similar to the assumption of Do et al. [2]. Contrary to the assumption, the match quality of MDSM is affected by the factors discussed in Section 6.3. MDSM is able to handle the scalability of schemas because it employed the Hungarian method for performing matches. The Hungarian method uses combinatorial optimisation, which is capable of solving the NP-problem. Even though the speed of performing matches with MDSM was not a concern in this study, we found that less than 1.5 s is taken to match 100 elements. This excludes *pre-match* effort time.

7. Conclusion and future work

The schema matching approach, MDSM, is an important process and is used in our proposed integrated microarray database. The formalisation problem of MDSM can be divided into two main issues: scoring function and optimal matching. The similarities between attributes are scored with respect to their syntactic and semantic data structure. These similarity results can be represented as a weighted bipartite graph; therefore, the Hungarian method is used to find an optimal matching between attributes from different schemas. These matching results are subsequently used for constructing the microarray global schemas.

The significant findings and contributions in this work are as follows:

- The fast and semi-automatic matching method, MDSM, is an effective and practical approach. The key capabilities of MDSM include (1) the reconciliation of structures and terminologies of two microarray schemas and (2) serving as generic data models and applications.
- Based on the experimental evaluation of existing public microarray schemas, it is found that MDSM performs very well. The match quality is computed for both single match experiments and entire match tasks of the evaluations. Our results show that performance exceeds 70% (measured as the *precison*, *recall*, and *overall* of schema matching process).
- Contrary to the assumption of Do et al. [2], the larger schema does not lead to lower match quality. MDSM can cope with the larger schemas without the loss of match quality. However, the match quality of MDSM is impacted upon by the following factors:

- Given the same input schemas, the match quality varies from user to user and from application to application.
- The configuration and algorithm of the match.

In regard to future research, we will focus on the following issues:

- The similarity function. Even though the similarity results that were obtained from the combination of two similarity measures are effective, some situations might be ambiguous and complicated. This is due to our similarity measure that relies heavily on the development of MGED microarray ontology.
- The larger and more complex schema will be employed to evaluate match quality of the proposed matching approach.
- The *pre-match* and *post-match* effort will be taken into account. In this research work, the manual effort was required to determine similarity values. The machine learning methods will be used to reduce the amount of manual work required in future work.
- The speed of MDSM will be investigated to guarantee that MDSM is fast enough to obtain results in real-world communication.

Even though attempts are made to develop microarray data resources which correspond to MGED ontology, development on these databases has not reached its completion. Furthermore, some existing microarray repositories do not provide the structure of schemas or schemaless. This problem will be considered to enhance our matching method in the future.

Acknowledgement

The work in this paper was partially supported by The Australian Research Council Grant DP0559251.

References

[1] S. Bergamaschi, S. Castano, S. De Capitani di Vimercati, S. Montanari, M. Vincini, An intelligent approach to information integration, International Conference on Formal Ontology in Information Systems (FOIS'98), Trento, Italy, 1998, pp. 253–267.
[2] H.H. Do, S. Melnik, E. Rahm, Comparison of schema matching evaluations, Revised Papers from the NODe 2002 Web and Database-Related Workshops on Web, Web-Services, and Database System, Springer-Verlag, 2003, pp. 221–237.
[3] H.H. Do, E. Rahm, COMA—A system for flexible combination of schema matching approaches, in: Proceedings of the 28th International Conference on Very Large Databases (VLDB), Hong Kong, 2002, pp. 610–621.

[4] A.H. Doan, P. Domingos, A. Levy, Learning source descriptions for data integration, in: Proceedings of 3rd International Workshop on the Web and Databases, Dallas, TX, 2000, pp. 81–86.
[5] N. Gonzalo, A guided tour to approximate string matching, ACM Computing Surveys 33 (1) (2001) 31–88.
[6] Y. Li, Z. Bandar, D. McLean, An approach for measuring semantic similarity between words using multiple information sources, IEEE Transactions on Knowledge and Data Engineering 15 (4) (2003) 871–882.
[7] J. Madhavan, P.A. Bernstein, E. Rahm, Generic schema matching with Cupid, in: Proceedings of the 27th International Conference on Very Large Databases (VLDB), Rome, Italy, September 2001, pp. 49–58.
[8] P. Mitra, G. Wiederhold, J. Jannink, Semi-automatic integration of knowledge sources, in: Proceedings of Fusion'99, Sunnyvale, USA, 1999, pp. 1–9.
[9] D. Murphy, Gene expression studies using microarrays: principles, problems, and prospects, Advances in Physiology Education 26 (1–4) (2002) 256–270.
[10] Nature Genetics Editor (Ed.), Chipping Forecast IINature Genetics Supplement 32 (2003) 461–552.
[11] D.V. Nguyen, A.B. Arpat, N. Wang, R.J. Carroll, DNA microarray experiments: biological and technological aspects, Biometrics 58 (2002) 701–717.
[12] L. Palopoli, D. Sacca, D. Ursino, Semi-automatic semantic discovery of properties from database schemas, in: Proceedings of International Database Engineering and Applications Symp (IDEAS), IEEE Computer, 1998, pp. 244–253.
[13] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall, Englewood Cliffs, NJ, 1998.
[14] A. Pirkola, T. Hedlund, H. Keskustalo, K. Järvelin, Dictionary-based cross-language information retrieval: problems, methods, and research findings, Information Retrieval 4 (3/4) (2001) 209–230.
[15] A. Pirkola, H. Keskustalo, E. Leppänen, A.P. Känsälä, K. Järvelin, Targeted *s*-gram matching: a novel *n*-gram matching technique for cross- and monolingual word form variants, Information Research 7 (2) (2002). Available from: <http://InformationR.net/ir/7-2/paper126.html>.
[16] S. Prompramote, Y. Chen, F. Maire, Information management for microarray experimental data, in: 5th IFAC Symposium on Modelling and Control in Biomedical Systems, IFAC2003, Elsevier Science, 2003, pp. 377–382.
[17] E. Rahm, P.A. Bernstein, A survey of approaches to automatic schema matching, The VDLB Journal 10 (2001) 334–350.
[18] E. Rahm, P.A. Bernstein, On Matching Schema Automatically, Microsoft Research Publications, 2001, pp. 1–22.
[19] A.M. Robertson, P. Willett, Applications of *n*-grams in textual information systems, Journal of Documentation 54 (1) (1998) 48–69.
[20] J. Roddick, K. Hornsby, D. de Vries, A unifying semantic distance model for determining the similarity of attribute values, in: M. Oudshoorn (Ed.), Proceedings of the 26th Australasian Computer Science Conference, Adelaide, Australia, Vol. 16, ACS, 2001, pp. 111–118.
[21] A.M. Rodríguez, M.J. Egenhofe, Determining Semantic Similarity Among Entity Classes From Different Ontologies, IEEE Transactions on Knowledge and Data Engineering 15 (2) (2003) 442–456.
[22] A. Schrijver, Combinatorial Optimization: Polyhedra and Efficiency, Springer-Verlag, New York, LLC, 2003.
[23] D.K. Slonim, From patterns to pathways: gene expression data analysis comes of age, Nature Genetics Supplement 32 (2003) 502–508.

[24] T.P. Speed (Ed.), Statistical Analysis of Gene Expression Microarray Data, Chapman & Hall/CRC, Boca Raton, 2003.
[25] X.L. Sun, E. Rose, Automated Schema Matching Techniques: An Exploratory Study, Research Letters in the Information and Mathematical Sciences 4 (2003) 113–136.