A Method for Similarity-Based Grouping of Biological Data

Vaida Jakonienė, David Rundqvist, and Patrick Lambrix

Department of Computer and Information Science
Linköpings universitet, SE-581 83 Linköping, Sweden

Abstract. Similarity-based grouping of data entries in one or more data sources is a task underlying many different data management tasks, such as, structuring search results, removal of redundancy in databases and data integration. Similarity-based grouping of data entries is not a trivial task in the context of life science data sources as the stored data is complex, highly correlated and represented at different levels of granularity. The contribution of this paper is two-fold. 1) We propose a method for similarity-based grouping and 2) we show results from test cases. As the main steps the method contains specification of grouping rules, pairwise grouping between entries, actual grouping of similar entries, and evaluation and analysis of the results. Often, different strategies can be used in the different steps. The method enables exploration of the influence of the choices and supports evaluation of the results with respect to given classifications. The grouping method is illustrated by test cases based on different strategies and classifications. The results show the complexity of the similarity-based grouping tasks and give deeper insights in the selected grouping tasks, the analyzed data source, and the influence of different strategies on the results.

1 Introduction

During the last decade an enormous amount of biological data has been generated and techniques and tools to analyze this data have been developed. Many of these tools use data clustering and classification techniques. For instance, these techniques are used to find similar sequences for predicting the functionality of new sequences [GH04], to find correlated genes based on microarray data [SS02], or to classify publications according to an ontology to locate relevant documents faster [DS05]. A basic task underlying these approaches is the computation of a similarity value between objects. Different techniques are developed to compute a similarity value between objects based on the object types. For instance, edit distance [Lev66] and n-gram [PPF95] are well-established techniques to define similarity between strings, while BLAST [AGMML90] can be used to define a similarity measure between DNA or protein sequences. Recently, a number of projects discussed methods to compute semantic similarity over terms in a Gene Ontology (GO) ontology (e.g. [CSC05] and [SFSZ05]). The similarity between GO terms can be used to compute a similarity between data entries that are annotated with these GO terms [LSBG03].

U. Leser, F. Naumann, and B. Eckman (Eds.): DILS 2006, LNBI 4075, pp. 136–151, 2006.

Data entries in biological data sources are often complex and store different types of information. Although most of the research has focused on organizing the data based on aspects, such as sequence similarity and function, we need to analyze data using different aspects and from different points of view to obtain deeper insights in the characteristics of the data and to discover new knowledge. This means that we need to be able to organize the data based on different attributes or different combinations of attributes. [KLKTB04] illustrates how a combination of attributes could be used to find data entries describing the same protein. In this case, search on sequence similarity is complemented with the analysis of sequence length, organism and the data source where the sequence was originally submitted. In this paper we use the term *grouping* to refer to the task of organizing the data according to a certain aspect or a combination of aspects. Further, we concentrate on the task of *similarity-based grouping*. During similarity-based grouping the analyzed data entries are compared with respect to a selected subset of attributes, and similarity functions that are relevant to the attributes are used to compute the similarity of the stored values.

Grouping of data entries in one or more data sources is an operation underlying many different data management tasks. Grouping can be used to structure and visualize search results in a convenient way for the user. This is especially important when large data sources are studied. The possibility to get an overview over the data may lead to the discovery of new knowledge or may allow biologists to locate the information of interest faster. The identification of similar data entries and their grouping are core operations when performing data cleaning activities [HGPWW04]. The identified groups of similar data entries can be further analyzed and merged into a single data entry. In the context of data integration, techniques underlying grouping are important to correlate data entries at different data sources. The grouping task can be narrowed to the duplicate detection task, where it is required that matched data entries represent the same real-world object. Duplicate detection can be both used for data cleaning [KLKTB04] and for data integration [BBBDN05].

A number of aspects influence the quality of the grouping results: the quality of the data sources, the selection of the grouping attributes and the algorithms implementing the grouping procedure. In some cases, given a grouping task, it can be difficult to decide on which attributes to perform grouping. Also, different sets of attributes may seem relevant to the grouping task, but lead to varying quality of the results [KLKTB04]. Further, suitable algorithms need to be selected to compute the similarity between data entries and to organize similar data entries into groups. Many methods exist, but it is often not clear which methods perform best for which grouping tasks. The study of the properties, and the evaluation and the comparison of the different aspects that influence the quality of the grouping results, would give us valuable insight into the best way to use the grouping procedures. It would also lead to recommendations on how to improve the current procedures and develop new procedures. To be able to perform such studies and evaluations we need environments that allow us to compare and evaluate different grouping procedures.

In this paper, as a first step towards the development of an environment to support grouping tasks, we propose a method that covers the main steps and components that should be included in such environments (section 2). The grouping method is illustrated by test cases based on different strategies and classifications (section 3). In subsections 3.1-3.5 we describe the grouping task, the test cases, the implementation approaches and the evaluation approach according to the method. In subsection 3.6 we analyze the evaluation and grouping results and show how we obtain deeper knowledge about the grouping tasks, the analyzed data source, and the influence of different strategies on the results. The paper concludes in section 4.

2 Method for Grouping Biological Data

In this section we describe a method that supports similarity-based grouping of biological data and that enables the development of grouping procedures. The components and the main steps of the method are illustrated in figure 1.

The method uses as input the data source on which the grouping is performed. Further, it uses similarity functions that can compute similarity values between data values, and grouping attributes on which we base the computation of the similarity of data entries in the data source. There may also be external sources to support the grouping task. Based on this input the method can generate groupings of data. In addition to this, the method also allows the evaluation and analysis of the grouping results. For this purpose we use a library of known

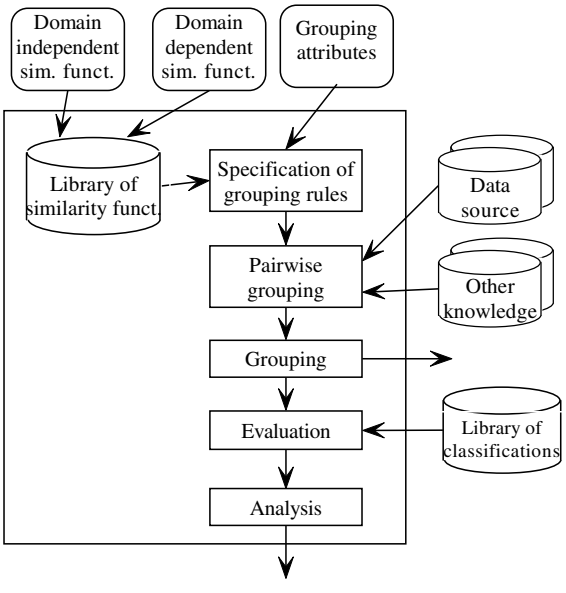

Fig. 1. Method for similarity-based grouping

classifications. The library stores selected sets of data entries organized into classes[1]. The method then returns the generated groups of data entries as well as reports from the evaluation and analysis.

The main steps in the method are: 1) specification of grouping rules that define how to identify similar data entries; 2) pairwise grouping by computing the similarity between pairs of data entries; 3) grouping similar data entries into groups; 4) evaluation of the quality of the generated groups with respect to given classes and 5) analysis of the grouping and evaluation results. We note that the library of similarity functions and the specification of grouping rules are often domain dependent, while the other steps are based on general techniques and approaches. Further, before the grouping procedure can be applied to a data source, the data source usually needs to go through a number of data transformation steps, such as merging of data and data translation from one format to another. In the remainder of this section we briefly describe the different components and steps of our method.

Library of Similarity Functions. This component represents a collection of functions that compute similarity scores between data values:

$SimFunc(v_1, v_2) \rightarrow [0, 1]$. We distinguish between domain independent and domain dependent similarity functions. The former group of similarity functions can be applied to any kind of data values, for instance, string-based functions. In the latter case, similarity functions can only be used to compare values of specific types of data values, for instance, protein sequences. The designer of similarity functions should be aware of and develop approaches for dealing with collection type values and missing values, which are often encountered in biological data.

Specification of Grouping Rules. In our method grouping rules are used to express conditions on which two data entries are compared for similarity. During this step the user defines a grouping rule or selects an already available grouping rule that is deemed to be relevant to the current grouping task. A grouping rule may combine different similarity functions applied to one or more grouping attributes. For instance, a grouping rule may specify that two data entries are similar if the sum of weighted similarity functions applied to certain grouping attributes is higher than a given threshold. In general, the specification of grouping rules is not a trivial task. We may need to do much experimentation and fine-tuning to specify rules that are adequate for a certain grouping task. Our method aims to support this.

Pairwise Grouping. Given a data source and grouping rules, the pairwise grouping step performs pairwise comparisons of data entries. Auxiliary domain knowledge may be used. The result of this step is the identification of the pairs of data entries that are similar. The pairwise grouping includes the following steps: a) selection of pairs of data entries in the data source for the comparison; b) comparison of data values of the selected grouping attributes by applying the

[1] In the rest of the paper we use *classes* to refer to given classifications and *groups* to refer to the results of grouping techniques.

defined similarity functions; and c) comparison of the selected data entries on the basis of given grouping rules. While for small data sources all pairs of data entries can be analyzed, for large data sources pruning techniques may be used to decrease the number of performed comparisons.

Grouping. The step takes as input pairs of similar data entries and organizes the data entries into a set of groups composed of similar data entries. Different techniques can be used to perform grouping and they may vary on a number of aspects. For instance, the groups can be allowed to overlap or may be required to be disjoint. Some approaches may require the transitivity property between similar data entries or they may allow to ignore some similarity relationships, e.g. in order to split a group into smaller groups. Also, there may be a restriction on the total number of groups or on the number of allowed data entries in a group. The generated groups can be seen as the final result of the grouping method or they can be used an input to the evaluation step.

Evaluation. During the evaluation step different measures are computed to evaluate the quality of the grouping results. Two groups of quality measures are distinguished [SKK00]: internal and external quality measures. Internal quality measures compare different groupings only based on information obtained during the grouping (e.g. pairwise similarity between data entries). External quality measures evaluate the grouping results with respect to known classes. As emphasized in [SKK00], to select the best grouping approach for the analyzed task, the approaches have to be compared with respect to a number of measures. In our method, the library of classifications is used to compute external quality measures.

Analysis. During this step the grouping and evaluation results are analyzed. Different forms and reports are generated providing support for exploring the results from different points of view. For instance, valuable insight could be gained by analyzing and studying the entries belonging to a single group, the correlation between groups and classes, and the influence of external knowledge on the results.

3 Test Cases

In this section we illustrate our method for similarity-based grouping of biological data. Further, we show how the method can be used to gain deeper knowledge about a particular setting, i.e. about the data source, the different steps in the method, and the influence of the various choices on the results. In our test cases we worked on two grouping tasks: grouping of proteins with respect to their biological function and with respect to what classes of isozymes they belong to. Proteins are isozymes (or isoenzymes) if they are enzymes that catalyze the same chemical reaction, but they may differ in their amino acid sequences [BTS02]. Isozymes differ in their kinetic properties, the way they are regulated by other proteins and quantities in which they are expressed in different tissues. For example, the enzyme *Lactate dehydrogenase* is built by two isozymic polypeptide

chains: H and M. The H isozyme functions optimally in aerobic environments and is expressed highly in the heart, while the M isozyme works under anaerobic conditions and is expressed highly in skeletal muscle. For each grouping task we explore the impact of different grouping attributes and grouping rules, which use different types of similarity functions. Also, we study the influence of different grouping algorithms. We describe the setting of the experiments and discuss the grouping and evaluation results.

3.1 Data and Knowledge

Data Source. The data source contains 190 human proteins involved in glycolysis that we retrieved on the 6th of October 2005 from the Entrez retrieval system. The data entries have different origin. They either come from the data sources RefSeq, SWISS-PROT, PRF, PIR and PDB, or are translated from nucleotide sequences at GenBank, EMBL and DDBJ.

Grouping Attributes. The following types of data were selected as relevant to the given grouping tasks:

- DEFINITION is an attribute that describes a protein. It combines information on protein name, synonymous names, isozyme indicator and organism name, for instance, "`ATPase, H+ transporting, lysosomal 31kD, V1 subunit E isoform 1 [Homo sapiens].`".
- PRODUCT is an attribute that holds the name of the gene product and, in some cases, stores the isozyme indicator in a less complex way compared to DEFINITION, for instance, "`liver phosphofructokinase isoform a`".
- SEQUENCE[2] is the attribute where the amino acid sequence of a protein is stored, for instance,

```
  1 malsdadvqk qikhmmafie qeanekaeei dakaeeefni ekgrlvqtqr lkimeyyekk
 61 ekqieqqkki qmsnlmnqar lkvlrarddl itdllneakq rlskvvkdtt ryqvlldglv
121 ...
```

- GO[3] ANNOTATIONS. Some of the data entries are annotated by one or more GO terms, which are denoted by their GO id:s, for instance, "`go:0005524 | go:0004396`". Several attributes in a document may contain GO terms. For our experiments we found GO terms in the attribute DBSOURCE for SWISS-PROT data entries and in the "note" property in the "CDS" field under the FEATURES attribute for the other data entries.

Other Knowledge. Of the 190 data entries, only 71 data entries were originally annotated by GO terms, which we refer to by GO_{ann}. To increase the number of

[2] In the original file this attribute is called ORIGIN, but for the sake of readability we use term SEQUENCE.

[3] The GO ontologies are de facto standard ontologies that describe the roles of genes and proteins in different organisms [GO00]. The three independent publicly available ontologies are: biological process, molecular function and cellular component. Today, many different bio-data sources are annotated with GO terms. The terms in GO are arranged as nodes in a directed acyclic graph, where multiple inheritance is allowed.

data entries with GO terms we used mappings between data values and ontological terms found on the web pages of the GO Consortium. In the experiments, we used the mapping *spkw2go* to translate values of the KEYWORDS attribute into GO terms, which we refer to by GO_{sw}. Also, we used the mapping *ec2go* to translate values of the EC-NUMBER attribute into GO terms, which we refer to by GO_{ec}. During the grouping process, knowledge in the GO ontology was used to compute similarity scores. To explore the quality of the available mappings, we decided to analyze GO_{ann}, GO_{sw} and GO_{ec} in different combinations. All these combinations resulted in variants of our original data source that included only data entries annotated by these terms. As a result, the number of analyzed data entries differed among the test cases. For instance, GO_{ann} and GO_{sw} annotations were available for 75 data entries, while GO_{ann} and GO_{ec} occurred in 92 data entries.

Classifications. The data entries were, for the whole data source, manually classified into 28 disjoint classes according to biological function. For instance, all data entries in one of the classes relate to *Phosphofructokinase* (which is the enzyme responsible for turning Fructose-6-phosphate into Fructose-1, 6-biphosphate). Data entries belonging to the same class may describe the same real-world protein, proteins having similar function, fragments of proteins having the same or similar function, and hypothetical proteins that are strongly believed to have the same or similar function. In the classification the two largest classes consist of 56 and 53 data entries, while 13 classes consist of a single data entry. The largest classes represent the enzymes *Pyruvate kinase* and *Phosphofructokinase*, which are the most prominent regulatory enzymes in glycolysis.

The isozyme classification was constructed by further dividing the classes in the function-based classification. For example, the data entries in the *Phosphofructokinase (PFK)* class were distributed into three classes: *Liver-type PFK*, *Platelet-type PFK* and *Muscle-type PFK*. The classification resulted in 52 disjoint classes, where the two largest classes contained 29 and 27 data entries, while 31 classes contained a single data entry.

3.2 Library of Similarity Functions

We used domain independent and domain dependent similarity functions.

$EditDist(v_1, v_2)$ is a function that computes similarity based on the edit distance between strings. The distance between strings v_1 and v_2 is defined by the least number of operations needed to turn v_1 into v_2. The allowed operations are insertions, deletions and replacements. The distance is transformed into a similarity score by the function: $score = 1 - \frac{distance}{MaxLength(v_1, v_2)}$.

$SeqSim(v_1, v_2)$ is a function that performs pairwise sequence alignment and returns a similarity score between sequences. We use a sequence alignment tool implemented in Java, JAligner [JAligner], to compute an alignment between the sequences. The tool implements an improved version of the Smith-Waterman algorithm for producing gapped alignments between sequences. The similarity score is defined as the number of matches in the alignment divided by the length of the alignment.

$SemSim(v_1, v_2)$ is a function that computes the similarity between two sets of GO terms. To evaluate the distance between two GO terms we use an edge-based algorithm that counts the number of edges needed to traverse the GO hierarchy from one term to another. The algorithm counts the number of is_a relationships needed to go up in the hierarchy u, the number of is_a relationships needed to go down in the hierarchy d and the number of other relationships o. The similarity between two GO terms is then defined as $score = e^{-0.5 \cdot ((\frac{u}{p_u})^2 + (\frac{d}{p_d})^2 + (\frac{o}{p_o})^2)}$, where p_u, p_d and p_o are weights for the different types of edges. In the test cases we used p_u=2, p_d=1 and p_o=1. Two sets of GO terms are defined to be similar if each term of one set is similar to a term in the other set.

Table 1. Test cases. Grouping on protein function. n^e - number of analyzed entries, n^g - number of groups, n^c - number of classes, p - purity, E - entropy, F - F-measure, MI - mutual information.

Test case	Grouping rule	n^e	n^g	n^c	p	1-E	F	MI
1	$SemSim(GO_{ann})$> 0.95 GO_{ann} for component, process, function domains	71	23	24	0.90	0.93	0.88	0.86
2	$SemSim(GO_{ann})$> 0.95	67	26	23	1.00	1.00	0.97	0.91
3	$SemSim(GO_{ann} + GO_{sw})$> 0.95	75	23	24	0.80	0.87	0.79	0.79
4	$SemSim(GO_{ann} + GO_{ec})$> 0.95	92	26	25	1.00	1.00	0.99	0.88
5	$SemSim(GO_{ann} + GO_{sw} + GO_{ec})$> 0.95	93	26	25	0.86	0.93	0.88	0.81
6	$SemSim(GO_{ann} + GO_{sw} + GO_{ec})$> 0.95; parent GO terms removed	93	26	25	0.86	0.93	0.88	0.81
7	$SemSim(GO_{ann})$> 0.95 or $SemSim(GO_{sw})$> 0.95 or $SemSim(GO_{ec})$> 0.95	93	14	25	0.48	0.65	0.51	0.59
8	$SemSim(GO_{ann})$> 0.95 or $SemSim(GO_{ec})$> 0.95	92	26	25	1.00	1.00	0.99	0.88
9	$SemSim(GO_{ann} + GO_{ec})$= 1	92	26	25	1.00	1.00	0.99	0.88
10	$SemSim(GO_{ann} + GO_{ec})$> 0.85	92	21	25	0.70	0.78	0.71	0.68
11	$SemSim(GO_{ann} + GO_{ec})$> 0.95 grouping algorithm: cliques	92	29	25	1.00	1.00	0.84	0.88
12	$EditDist(definition)$> 0.9, for $GO_{ann} + GO_{ec}$	92	67	25	1.00	1.00	0.59	0.77
13	$EditDist(definition)$> 0.7, for $GO_{ann} + GO_{ec}$	92	55	25	0.96	0.97	0.66	0.76
14	$SeqSim(sequence)$> 0.85, for $GO_{ann} + GO_{ec}$	92	44	25	1.00	1.00	0.74	0.81
15	$EditDist(definition)$> 0.85	190	94	28	0.97	0.98	0.54	0.57
16	$EditDist(product)$> 0.85	190	105	28	0.99	0.99	0.49	0.57
17	$EditDist(definition)$> 0.7	190	68	28	0.81	0.87	0.56	0.50
18	$EditDist(product)$> 0.7	190	78	28	0.95	0.98	0.64	0.58
19	$EditDist(definition)$> 0.9 or $EditDist(product)$> 0.9 or ($EditDist(definition)$> 0.6 and $EditDist(product)$> 0.6)	190	64	28	0.94	0.96	0.70	0.58
20	$SeqSim(sequence)$> 0.85	190	59	28	0.99	0.99	0.66	0.62

Table 2. Test cases. Grouping on isozymes. n^e - number of analyzed entries, n^g - number of groups, n^c - number of classes, p - purity, E - entropy, F - F-measure, MI - mutual information.

Test case	Grouping rule	n^e	n^g	n^c	p	1-E	F	MI
21	$EditDist(definition) > 0.85$	92	67	47	0.89	0.95	0.73	0.85
22	$SemSim(GO_{ann} + GO_{ec}) > 0.95$	92	26	47	0.59	0.79	0.65	0.79
23	$EditDist(product) > 0.85$	92	56	47	0.83	0.92	0.73	0.84
24	$SeqSim(sequence) > 0.85$	92	44	47	0.91	0.96	0.90	0.91
25	$EditDist(definition) > 0.85$	190	94	52	0.87	0.93	0.63	0.67
26	$EditDist(product) > 0.85$	190	105	52	0.88	0.94	0.58	0.68
27	$SeqSim(sequence) > 0.85$	190	59	52	0.95	0.97	0.91	0.75

3.3 Specification of Grouping Rules

The studied grouping rules are collected in the second column of table 1 and table 2 for the grouping tasks on function and isozymes, respectively. The similarity functions in the tables are shown with one argument representing the type of the compared values. When exploring grouping on function, we developed a number of test cases based on various combinations of GO_{ann}, GO_{sw} and GO_{ec} (test cases 1-11). All these cases, except test case 1, use only function-related terms in the GO ontology. In the test cases 12-20 we analyzed the applicability of the values in attributes DEFINITION, PRODUCT and SEQUENCE for grouping on function. The test cases 8-14 are run on the data entries used in test case 4, since test case 4 had the best results among the test cases run on GO terms. The test cases 15-20 are performed on the whole data source, i.e. in total 190 data entries. The test cases include experiments with different thresholds, complex rules combining similarity functions and an experiment with a different grouping algorithm (test case 11). Similarly as for grouping on function, we tested the applicability of DEFINITION, PRODUCT, GO ANNOTATION and SEQUENCE for grouping on isozymes. Table 2 contains the grouping rules applied on the data entries analyzed in test case 4 and the grouping rules applied on all data entries in the data source.

3.4 Pairwise Grouping and Grouping

To perform the actual grouping of the data entries based on a given rule, first pairwise grouping between the data entries and then, grouping of similar data entries are performed. In our experiments, all pairs of data entries in the data source are compared to each other and are identified as similar or not. To organize the similar data entries into groups we experimented with two approaches: cliques and connected components. *Cliques* require that all data entries in a group are similar to each other. In this approach, the generated groups may overlap as all similarity relationships are taken into account. *Connected components* collect all data entries that are directly or transitively similar to each other into a single group. As a result, the approach generates disjoint groups. From the discussed test cases only test case 11 uses cliques. For the other test cases we used connected components.

3.5 Evaluation

To evaluate the results of the test cases we used external quality measures described in [Str02], purity, F-measure, entropy and mutual information. These measures are defined as follows. Let n be the total number of analyzed data entries, n^g the number of generated groups and n^c the number of given classes. Let n_i^g denote the number of entries in group i and n_j^c denote the number of entries in class j. Let n_{ij} represent the number of entries that are common to group i and class j. For each group i and class j, the precision is defined as $p_{ij} = \frac{n_{ij}}{n_i^g}$ and the recall as $r_{ij} = \frac{n_{ij}}{n_j^c}$.

Purity evaluates the average precision of the groups with respect to their best matching classes. For each group i purity is defined as $p_i = \max_j\{p_{ij}\}$. The purity for the whole grouping is defined as

$$p = \sum_{i=1}^{n^g} \frac{n_i^g}{n} p_i$$

F-measure is the average F-measure of the classes with respect to their best matching groups. The measure combines precision and recall into a single value. For each combination of group i and class j the F-measure is $F_{ij} = \frac{2 \cdot p_{ij} \cdot r_{ij}}{p_{ij} + r_{ij}}$. The F-measure for class j is defined as $F_j = \max_i\{F_{ij}\}$ and the F-measure for the whole grouping is defined by

$$F = \sum_{j=1}^{n^c} \frac{n_j^c}{n} F_j$$

Normalized entropy analyzes how on average the data entries in each group distribute among the classes. $E_i = -\sum_{j=1}^{n^c} p_{ij} \log_{n^c} p_{ij}$ is the normalized entropy for group i and the total normalized entropy for the whole grouping is

$$E = \sum_{i=1}^{n^g} \frac{n_i^g}{n} E_i = -\frac{1}{n} \sum_{i=1}^{n^g} \sum_{j=1}^{n^c} n_{ij} \log_{n^c} p_{ij}$$

Mutual information is the average measure of correspondence between each group and class. The mutual information is calculated as

$$MI = \frac{2}{n} \sum_{i=1}^{n^g} \sum_{j=1}^{n^c} n_{ij} \log_{n^g \cdot n^c} \left(\frac{n_{ij} \cdot n}{n_i^g \cdot n_j^c} \right)$$

The evaluation results for each test case are shown in tables 1 and 2.

3.6 Analysis

In this subsection we take a closer look at the grouping and evaluation results for our test cases. We compare different test cases and discuss issues that have

an impact on the results. Further, using 3 examples we discuss interesting cases in some more details.

Best Test Cases. The test cases described in the tables 1 and 2 reveal that grouping on GO ANNOTATIONS combining GO_{ann} and GO_{ec} is best suited for grouping the data entries on function (test cases 4, 8 and 9) and that grouping on SEQUENCE is best suited for grouping on isozymes (test cases 24 and 27). For the test cases 4, 8 and 9, the grouping results were only imprecise in the distribution of data entries of one class between two groups (see example 2 below). The same grouping results for the test cases 4, 8 and 9 could be caused by the type of the compared GO annotations and the type of the used grouping approaches. In the case of grouping on isozymes, grouping on sequence performed reasonably well both on the fragment of the data source and on the whole data source.

Grouping on GO Annotation. Test cases 1 and 2 show that the removal of the component and process terms from GO_{ann} increases the quality of the grouping on function. For instance, each group in test case 2 includes entries from a single class (p=1). However, in some cases valuable information may be removed (see example 2 below). This suggests that a method that assigns different weights to different types of GO terms may improve the results.

From the analysis of test cases 2-8 we conclude that *spkw2go* mappings are not suitable for grouping on function. SWISS-PROT keywords are quite general and are mapped to high level GO terms. For instance, some SWISS-PROT data entries contain 'Glycolysis' as a keyword, while all the data entries in the data source relate to 'Glycolysis'. Therefore, some data entries where grouped together even though they differed in more specific functions, i.e. belonged to different classes. For instance, test case 3 generated 2 groups containing data entries from several classes. In contrast to *spkw2go*, GO terms obtained through *ec2go* mappings were specific enough. This is because EC numbers precisely identify the function of the described sequence. For instance, EC:2.7.1.11 maps to the GO term '6-phosphofructokinase activity', which is a very specific function.

Test cases 5 and 6 show that only using the most specific GO terms in the data entries, does not have an impact on the grouping result. This depends partly on the available GO annotations, which for some data entries match exactly. It also depends on the approach to compare sets of GO terms.

Grouping on Definition and Product. DEFINITION- and PRODUCT-based groupings perform worse than SEQUENCE-based groupings both for function and isozymes. The large number of generated groups in test cases 15-18 shows that the data values in these fields vary a lot. For instance, about 3 times more groups are generated than there are classes in the case of grouping on function for the threshold 0.85. Also, PRODUCT values are not available for some data entries. From the current test cases no definite conclusions can be made about the suitability of these types of data for grouping tasks. Further studies are needed.

Grouping on Sequence. Based on test cases 14 and 20, we can conclude that grouping on sequences is too specific to be used for grouping data entries on

function. There are nearly twice as many groups as there are classes. For instance, the data entries of the *Pyruvate kinase* class were distributed between a group covering muscle-type sequences and a group covering liver-/red blood cell-type sequences. This example confirms the observation that sequence similarity based grouping is better suited for grouping on isozymes.

Impact of Threshold. Test cases 4, 9 and 10 where grouping is performed on GO_{ann} and GO_{ec} with thresholds 0.95, 1 and 0.85, show that the best quality results are returned when all GO terms of one data entry appear among GO terms of the other data entry. For a slightly lower threshold, the quality of the results drops fast. The test cases 15-18 show that PRODUCT performs better for the lower threshold, while for DEFINITION-based grouping, although less groups are generated, the decrease of the threshold produces lower quality results. In general, experiments with different thresholds allow us to explore the correlation of data entries at different levels of similarity.

Complex Rules. Test cases 7, 8 and 19 illustrate the use of complex rules that gave us a better understanding of the data and enabled an increase of the quality of the grouping results. The negative impact of GO_{sw} is shown by test case 7 that performs much worse than test case 8. The grouping rule combining DEFINITION and PRODUCT (test case 19) resulted in increased quality of the results in comparison with test cases 15-18, which perform grouping on a single attribute.

Impact of Grouping Algorithm. Test cases 4 and 11 illustrate the impact of the different grouping approaches, in our case, connected components and cliques. As cliques put stronger requirements on the grouped data entries and allow overlapping groups (see example 3 below), a larger number of groups are generated and a lower F-measure is obtained. Based on the measures, however, we cannot make a decisive claim about which of the two grouping approaches performs better. The nature of the approaches is very different. They complement each other and give different ways of presenting the results. For instance, by comparing the results of the grouping approaches, subgroups of data entries that are interconnected in a stronger way to each other than to the rest of the entries in the group, can be located. Such subgroups could be generated for several reasons, such as the fact that the described sequences may slightly differ in functionality or that the data entries may have incomplete information.

In the remainder of this section we investigate some of the results in more details.

Example 1. A Group Covers Several Classes. In this example we take a closer look at group 2 of test case 1. The group includes data entries from four classes: *Phosphofructokinase* (class 3), *Pyruvate dehydrogenase* (class 11), *Nuclear receptor subfamily 1* (class 16) and *Pyruvate dehydrogenase kinase* (class 27). The data entries belonging to these classes together with their GO annotations are given in table 3.

A combination of reasons caused the data entries to be organized into the same group. 1) The algorithm that compares sets of GO terms considers the

Table 3. Test case 1. Data entries in group 2.

Accession #	Class #	GO Annotations
Q01813	3	**GO:0005945**, GO:0003872, GO:0006096
NP_000280	3	**GO:0005945**, GO:0005524, GO:0016301 GO:0000166, GO:0016740, GO:0000287 GO:0003872, GO:0006096, GO:0006006 GO:0005977, GO:0006110
NP_002618	3	**GO:0005945**, GO:0005524, GO:0016301 GO:0000166, GO:0016740, GO:0000287 GO:0003872, GO:0006096
P11177	11	*GO:0004739*, GO:0006099
P29803	11	*GO:0004739*
P10515	11	**GO:0005967**, GO:0004742, GO:0006085
NP_000275	11	**GO:0005739**, GO:0016491, *GO:0004739* GO:0016624, GO:0006096, GO:0008152 GO:0006084
P08559	11	**GO:0005739**
NP_005114	16	**GO:0005634**, GO:0046872, GO:0003707 GO:0003700, GO:0003713, GO:0003714 GO:0006350, GO:0008203, GO:0007165 GO:0008206, GO:0006355
NP_002603	27	**GO:0005739**, GO:0005524, GO:0004672 GO:0004740, GO:0006006, GO:0006468

data entries Q01813, NP_000280, NP_002618, P10515, NP_000275, NP_005114 and NP_002603 to be similar to P08559 since the only GO term in the data entry P08559, GO:0005739 - the component term "Mitochondrion", has high similarity with other component terms in the other data entries (marked by bold in table 3). Similarly, the data entries P11177 and NP_000275 are similar to P29803 because of the included GO:0004739. The comparison algorithm ignores the fact that some data entries have more GO terms assigned than the others. 2) The test case uses a grouping approach that assumes that similarity between data entries is transitive. As a result, two groups of similar data entries identified previously are connected by NP_000275 into a single group of similar data entries. 3) The fact that some GO terms in the annotation were general and that some of the data entries contained very few GO terms also caused the classes to be grouped into a single group.

Example 2. A Class Distributed Among Several Groups. In test case 4 only class 11, describing *Pyruvate dehydrogenase complex*, is not matched perfectly by a group. The data entries are divided into two groups: P11177, NP_000275, P08559 and P29803 are grouped together, while P10515 appears in a separate group. The grouping result can be explained by the fact that class 11 describes an enzyme complex that consists of multiple copies of the three types of enzymes E_1, E_2 and E_3. The goal of the whole enzyme complex is to build the molecule acetyl-CoA from Pyruvate and CoA, but different types of enzymes

belonging to the complex vary in their function. For instance, E_1 has the major catalytic function, namely the pyruvate dehydrogenase function [BTS02]. In our case, P11177, NP_000275, P08559 and P29803 describe E_1, while P10515 describes E_2. The difference between functions is also reflected in the available GO annotations: P11177, NP_000275, P08559 and P29803 are annotated with "pyruvate dehydrogenase (acetyl-transferring) activity", while P10515 has "dihydrolipoyllysine-residue acetyltransferase activity". These two terms are far from each other in GO. Test case 1 is the only one that organizes all the data entries into a single group. This illustrates how the knowledge from the component and process GO ontologies may positively contribute to the grouping on function. For the other test cases, where only GO terms from the function ontology were used, the available GO annotations are too specific to identify the whole enzyme complex.

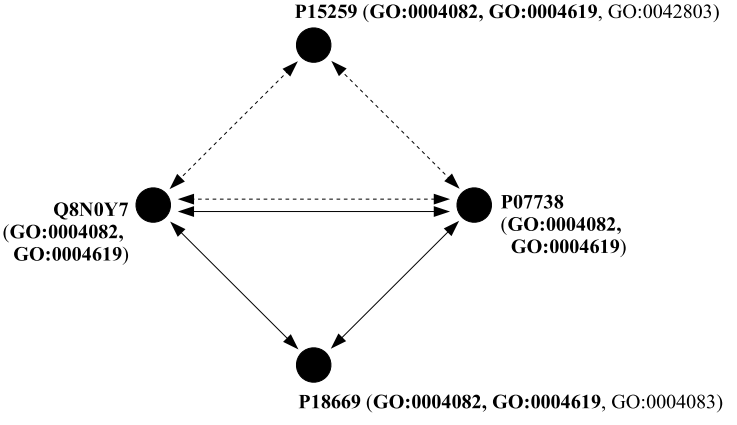

Fig. 2. Clique-based grouping. Class 8.

Example 3. Clique-Based Grouping. In test case 11 we used a clique-based algorithm for grouping similar data entries, which requires that all data entries in a group are similar to each other. The grouping result had a few cases where data entries in a class are distributed between 2 groups. For instance, the data entries of class 8, which describes *Phosphoglycerate mutase*, were distributed between 2 overlapping groups: one group contained P07738, Q8N0Y7 and P15259, while the other group included P07738, Q8N0Y7 and P18669 (figure 2). P15259 and P18669 are not found to be similar as the GO annotations differ from each other by GO:0042803 and GO:0004083, respectively. As a result, the data entries are moved to separate groups. P07738 and Q8N0Y7 are included in both of the groups as they are found to be similar to P15259 and P18669. We have checked that lowering the threshold, e.g. to 0.8, would combine all four data entries to a single group. This example illustrates the high impact of the different aspects in the grouping procedures on the grouping result; in this case the threshold, the

approach for comparing sets of GO terms and the approach for grouping similar data entries.

4 Conclusion

In this paper we motivated the need for environments that support the development and evaluation of similarity-based grouping procedures. We proposed a method that identifies the main components and steps that are important for such environments. Further, we illustrated the method by analyzing test cases for grouping of protein data entries with respect to their function and with respect to what classes of isozymes they belong to. The test cases illustrate the complexity of similarity-based grouping tasks. The choices made at the different steps in a grouping procedure have a large impact on the quality of the grouping results.

The test cases gave us also insights in different issues as well as interesting topics for future work. For instance, for the analyzed data source grouping on GO ANNOTATIONS combining GO_{ann} and GO_{ec} is best suited for grouping function and grouping on SEQUENCE is best suited for grouping on isozymes. Further studies are needed to investigate how other attributes can be useful for grouping tasks in life sciences. When grouping based on GO annotations, it is important to be aware of the fact that the annotations may be incomplete. In this paper we illustrated the possibility of partially compensating the lacking information by using mappings available at the GO Consortium. We observed that different mappings can be useful to different degrees. For instance, the mappings translating EC-NUMBER into GO terms (*ec2go*) gave good results, while mappings translating KEYWORDS in GO terms (*spkw2go*) were too general. When working with GO terms it is important to distinguish between general and more specific terms as they contribute differently to our knowledge. A number of test cases showed the importance for deeper studies to develop suitable methods to compare sets of GO terms and to explore their impact on the results.

The analysis and evaluation of the test cases was a time-consuming process. Tools are needed to support the different steps in our method. For instance, we need support at different levels of detail for the generation and analysis of the grouping results, the visualization and analysis of related data entries, and the analysis of the influence of external knowledge.

Acknowledgements

This research work was funded by CUGS (the Swedish national graduate school in computer science) and CENIIT (Center for Industrial Information Technology). The first and third authors are also members of the EU Network of Excellence REWERSE (Sixth Framework Programme project 506779, working group on a Semantic Web for Bioinformatics).

References

[AGMML90] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. *Journal of Molecular Biology*, 215:403-410.

[BTS02] Berg JM, Tymoczko JL, Stryer L (2002) *Biochemistry*. W.H. Freeman and Company, New York.

[BBBDN05] Bilke A, Bleiholder J, Böhm C, Draba K, Naumann F (2005) Automatic Data Fusion with HumMer. Demo at *VLDB Conference*, pp 1251-1254.

[CSC05] Couto FM, Silva MJ, Coutinho P (2005) Semantic similarity over the gene ontology: family correlation and selecting disjunctive ancestors. *Conference on Information and Knowledge Management*, pp 343-344.

[DS05] Doms A, Schroeder M (2005) GoPubMed: Exploring PubMed with the GeneOntology. *Nucleic Acids Research*, 33:W783-W786.

[GH04] Gabaldon T, Huynen MA (2004) Prediction of protein function and pathways in the genome era. *Cellular and molecular life sciences : CMLS*, 61(7-8):930-944.

[GO00] The Gene Ontology Consortium (2000) Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25-29. http://www.geneontology.org/.

[HGPWW04] Herbert KG, Gehani NH, Piel WH, Wang J, Wu CH (2004) BIO-AJAX: An Extensible Framework for Biological Data Cleaning. *SIGMOD Record*, 33(2):51-57.

[JAligner] Java implementation of the Smith-Waterman algorithm for biological sequence alignment. http://jaligner.sourceforge.net/

[KLKTB04] Koh JLY, Lee ML, Khan AM, Tan PTJ, Brusic V (2004) Duplicate Detection in Biological Data using Association Rule Mining. *ECML/PKDD Workshop on Data Mining and Text Mining for Bioinformatics*, pp 31-37.

[Lev66] Levenshtein VI (1966) Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707-710.

[LSBG03] Lord PW, Stevens R, Brass A, Goble CA (2003) Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. *Bioinformatics*, 19(10):1275-1283.

[PPF95] Pfeifer U, Poersch T, Fuhr N (1995) Searching Proper Names in Databases. *Conference on Hypertext - Information Retrieval - Multimedia*, pp 259-275.

[SS02] Shamir R, Sharan R (2002) Algorithmic Approaches to Clustering Gene Expression Data. Chapter in *Current Topics in Computational Biology*, Jiang T, Smith T, Xu Y, Zhang MQ editors, MIT Press, pp 269-299.

[SFSZ05] Speer N, Fröhlich H, Spieth C, Zell A (2005) Functional Distances for Genes Based on GO Feature Maps and their Application to Clustering. *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pp 142-149.

[SKK00] Steinbach M, Karypis G, Kumar V (2000) A comparison of document clustering techniques. *KDD Workshop on Text Mining*.

[Str02] Strehl A (2002) *Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining*. PhD thesis, University of Texas at Austin.