Ontology-driven Co-clustering of Gene Expression Data

F. Cordero, R. G. Pensa, A. Visconti, D. Ienco, M. Botta
Ontology-driven Co-clustering of Gene Expression Data
Editor: SPRINGER-VERLAG
2009
ISBN: 9783642102905

11th International Conference of the Italian Association for Artificial Intelligence: Emergent Perspectives in Artificial Intelligence, AI IA 2009
426 - 435
11th Conference of the Italian Association for Artificial Intelligence AI*IA 2009
Reggio Emilia, Italy
December 9-12, 2009

Ontology-driven Co-clustering of Gene Expression Data

Francesca Cordero[1,2,3], Ruggero G. Pensa[2], Alessia Visconti[2,3], Dino Ienco[2,3], and Marco Botta[2,3]

[1] Department of Clinical and Biological Sciences, University of Torino
[2] Department of Computer Science, University of Torino
[3] Center for Complex Systems in Molecular Biology and Medicine - SysBioM, University of Torino
{fcordero,pensa,visconti,ienco,botta}@di.unito.it

Abstract. The huge volume of gene expression data produced by microarrays and other high-throughput techniques has encouraged the development of new computational techniques to evaluate the data and to formulate new biological hypotheses. To this purpose, co-clustering techniques are widely used: these identify groups of genes that show similar activity patterns under a specific subset of the experimental conditions by measuring the similarity in expression within these groups. However, in many applications, distance metrics based only on expression levels fail in capturing biologically meaningful clusters.
We propose a methodology in which a standard expression-based co-clustering algorithm is enhanced by sets of constraints which take into account the similarity/dissimilarity (inferred by the Gene Ontology, GO) between pairs of genes. Our approach minimizes the intervention of the analyst within the co-clustering process. It provides meaningful co-clusters whose discovery and interpretation is increased by embedding GO annotations.

1 Introduction

Microarrays, and other high-throughput techniques, measure the expression level of thousands of genes in different samples captured at different time points or in different experimental conditions. The volume of data produced by these techniques is huge and grows up day by day. This requires the development of new computational techniques to store and evaluate the data and to formulate new biological hypotheses.

To this purpose, clustering techniques are widely used in microarrays data analysis that enable to discover homogeneous experimental clusters or genes clusters based on a distance measure quantifying the degree of correlation of expression profiles [1]. A limitation of traditional clustering techniques is that they are applied on gene sets or sample sets independently. To exceed this view, *co-clustering* algorithms [2] have been proposed: these identify groups of genes that show similar activity patterns under a specific subset of the experimental conditions.

The goal of co-clustering algorithms emphasizes one of the major target in computational biology: the discovery of regulatory modules that control gene transcription in biological model systems. Approaches based on co-clustering simultaneously cluster genes and conditions, and enable the discovery of more coherent and meaningful groups. The main practical reasons are that biological systems are inherently modular and that grouping genes into modules reduces the effective complexity of a given data set. For instance, the association of these modules with a specific histological cancer class may be exploited within an effective diagnostic tool.

In many applications, distance metrics based only on expression levels fail in capturing biologically meaningful clusters. Moreover, approaches based on clustering that identify gene signatures in specific conditions tend to base the analysis on their signal in the conditions under study. However, a simple list of genes associated with a certain tumor type is far from identifying the regulatory modules in which genes are involved. Several works proposed to define distance metrics based on different sources of information. As an advantage, additional information could help in resolving ambiguities or in avoiding erroneous linking based on spurious similarities.

The pioneer of this stream of works is Hanisch et al. [3]. They proposed a novel approach that allows for an entirely exploratory joint analysis of gene expression data and biological networks. The authors proposed a combined measure derived from gene expression data and metrics based on biological networks into a single distance function that they use as distance measure in a hierarchical average linkage clustering algorithm. Starting from Hanish's work, Steinhauser et al.[4] proposed a new measure that involves operon annotations, intergenic distance and transcriptional co-response data into a distance metric used in hierarchical clustering algorithms. More recently, Brameier et al. [5] presented a co-clustering approach based on self-organizing maps, where center-based clustering of standard SOMs have been combined with a representative-based clustering. The authors developed a two-level cluster selection where the nearest cluster according to GO distance is selected among the best matching clusters w.r.t. gene expression distance. In this work, co-clustering means that the GO-based clustering and expression-based clustering are performed in parallel. None of these methods perform co-clustering on both genes and samples at the same time.

Instead of combining ontology-based metrics and expression-based metrics within the same distance measure, we propose a methodology in which a standard expression-based co-clustering algorithm is enhanced by sets of constraints which take into account the similarity/dissimilarity (inferred by some background knowledge) between pairs of genes. Using constraints has been proved to be very effective in many applications, including gene expression analysis [6] and sequence analysis [7], since the user can decide which type of biological knowledge leads to the association among gene clusters and condition clusters. In this way the list of genes associated with a set of conditions may assume a specific meaning. Moreover, constraints can be generated by mixing different

semantics, while combining different semantics in a single measure is not an easy task. Defining these constraints by hand is not that simple either. Since the advantage of modularity is crucial in learning biological meaningful clusters from data, we decided to use the expressive power provided by Gene Ontology [8] to construct a set of similarity (must-link) and dissimilarity (cannot-link) constraints automatically.

Furthermore, for a correct usage of the technique presented in [6], similarly to all co-clustering techniques, the user has to specify the desired number of clusters on rows and columns. Deciding an adequate number of clusters is not trivial, and a bad choice may influence negatively the quality of co-clustering results. Thus, we adopt a preprocessing method that automatically determines a congruent number of clusters per rows and columns.

In a nutshell, we propose a new methodology that minimizes the intervention of the analyst within the co-clustering process and that provides meaningful co-clusters whose discovery and interpretation is enhanced by embedding GO annotations. To show the effectiveness of our approach, we apply our methodology on a gene expression dataset consisting on different stress conditions on the *S. Cerevisiae* yeast.

2 Constrained Co-clustering

In this section we briefly describe the constrained co-clustering algorithm presented in [6], and which is central to our methodology.

Let $X \in \mathbb{R}^{m \times n}$ denote a data matrix. Let x_{ij} be the expression level corresponding to gene (row) i and condition (column) j.

A co-clustering $C^{k \times l}$ over X simultaneously produces a set of $k \times l$ co-clusters (a partition C^r into k groups of rows associated to a partition C^c into l groups of columns) which optimize a given objective function. In this work we use the Cheng and Church residue [9] as objective function. Given an element x_{ij} of X, the residue of x_{ij} in the co-cluster defined by the sets of indices I and J, and whose respective cardinalities are $|I|$ and $|J|$, is given by $h_{ij} = x_{ij} - x_{Ij} - x_{iJ} + x_{IJ}$, where $x_{IJ} = \frac{\sum_{i \in I, j \in J} x_{ij}}{|I| \cdot |J|}$, $x_{Ij} = \frac{\sum_{i \in I} x_{ij}}{|I|}$, $x_{iJ} = \frac{\sum_{j \in J} x_{ij}}{|J|}$.

Let $H = [h_{ij}] \in \mathbb{R}^{m \times n}$ denote the matrix of residues computed using the previous definition. The objective function to be minimized is the sum of squared residues [10] computed as follows:

$$||H||^2 = \sum_{I,J} ||h_{IJ}||^2 = \sum_{I,J} \sum_{i \in I, j \in J} h_{ij}^2 \tag{1}$$

The kind of constraints we consider in this work are the two well-known **must-link** (similarity) and **cannot-link** (dissimilarity) constraints, also referred as pairwise constraints, since they involve pairs of objects.

If rows i_a and i_b (resp. columns j_a and j_b) are involved in a **must-link** constraint, denoted $c_=(i_a, i_b)$ (resp. $c_=(j_a, j_b)$), they must be in the same cluster of $C^r = r_1, \ldots, r_k$ (resp $C^c = c_1, \ldots, c_k$). If rows i_a, i_b (resp. columns j_a and j_b)

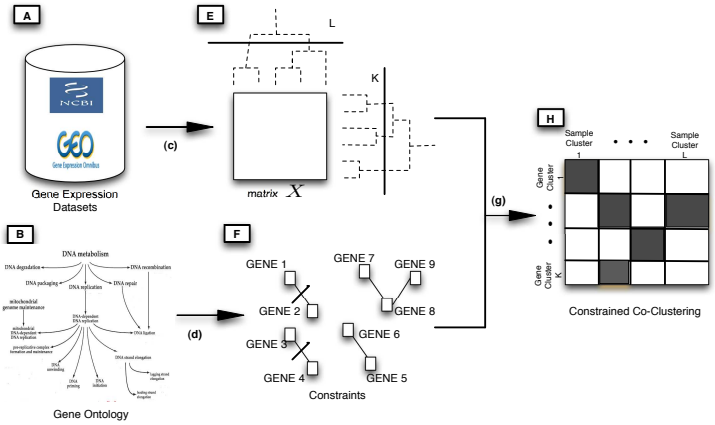

Fig. 1. Methodology overview

are involved in a **cannot-link** constraint, denoted $c_{\neq}(i_a, i_b)$ (resp. $c_{\neq}(j_a, j_b)$), they cannot be in the same cluster of $C^r = r_1, \ldots, r_k$ (resp $C^c = c_1, \ldots, c_k$).

We can then transform a set of must-link constraints over rows into a collection $\mathcal{M}_r = M_1, \ldots, M_N$, where each M_i is a set of rows involved by the same transitive closure of must-link constraints. Let us denote $\mathcal{M}_c$ the same set built for columns and let $\mathcal{C}_r$ and $\mathcal{C}_c$ be the sets of cannot-link constraints for rows and columns respectively. The co-clustering algorithm builds a $k \times l$ co-clustering over X, trying to minimize the objective function (1), and satisfying constraints $\mathcal{M}_r$, $\mathcal{M}_c$, $\mathcal{C}_r$, and $\mathcal{C}_c$. We skip the algorithmic details of this approach (see [6] for the complete algorithm).

3 Methodology Overview

Our framework is motivated by the necessity of using the previously described co-clustering algorithm on gene expression data, by limiting the number of user-defined parameters. So far, the user has to provide the following parameters: (i) number of row clusters; (ii) number of column clusters; (iii) a set of pairwise constraints (optional); (iv) convergence criterion (optional).

Providing a correct number of clusters is crucial for every clustering algorithm, since a wrong number might considerably alter the quality of the results. Unfortunately this is not an easy task, and some heuristics should be used to determine a correct number of clusters.

Providing a coherent and useful set of constraints is also a hard task. In classic semi-supervised applications, constraints are automatically selected from labeled samples, by selecting random pairs of labeled objects and setting a must-link or a cannot-link constraint depending on their class label. We will show how to extend this setting to gene expression data analysis.

Figure 1 shows an overview of the crucial steps of our methodology. The first two steps are performed independently. From one side, microarray experiments (A) are preprocessed to build a gene expression matrix consisting of normalized expression values. Other preprocessing techniques, such as missing value replacement, gene or sample filtering, are performed (c). The resulting matrix is then processed using the method described in [11], in order to determine a congruent number of row/column clusters (E). From the other side, the Gene Ontology graph (B) is processed in order to obtain nodes consisting of multiple regulated GO terms, linked by similarity relationships (d). The retained set of genes is mapped into the obtained GO graph to identify groups of similar and dissimilar genes, that contributes to the definition of must-link and cannot-link constraints (F). The central step is the constrained co-clustering algorithm (g) performed over the gene expression matrix obtained at step (c), using the number of row/column clusters discovered at step (E), and embedding the set of constraints built during step (F). The results (H) are a set of row clusters described by GO annotations, and a set of columns clusters described by experiment information.

4 Methodology Detailed Description

In this section we describe in full detail steps E and F of the methodology described beforehand.

4.1 Determining a suitable number of co-clusters

To estimate parameters k and l (i.e., the number of row and column clusters) we adopt the method described in [11], namely *L-method*, which aims at selecting the number of clusters that provides the best result in a hierarchical clustering setting. It consists in the following steps:

1. Generation of a hierarchical clustering using a distance matrix over the set of objects to be clustered;
2. Analysis of the resulting dendrogram, and assessment of the goodness (homogeneity) of the obtained clusters for each dendrogram level;
3. Analysis of the previously performed evaluations to identify the most suitable number of clusters.

These steps are first performed on the original matrix $X \in \Re^{m \times n}$ to identify the number of row clusters. They are then applied on a reduced matrix X^R, to obtain a suitable number of column clusters (we will motivate this choice later).

Step 1. Hierarchical clustering To build a cluster hierarchy over rows, we must first compute a distance matrix. The chosen distance metric is the one described in [9], which has been modified to enable the comparison between two rows. In particular, we consider Equation 1 for the submatrices of X consisting

in each pair of rows (considered as singleton clusters), and n singleton column clusters. The resulting distance matrix is then processed using a standard hierarchical clustering algorithm.

For columns, we do not process the transposed matrix directly, but we first reduce its dimensions using PCA. This choice is motivated by the fact that in gene expression data analysis usually the number of conditions is much lesser than the number of genes. On the resulting matrix we apply a standard hierarchical clustering algorithm using Euclidean distance.

Step 2. Analysis of the dendrograms Once the dendrogram has been computed, we analyze its levels. This analysis is performed in three steps:

1. starting from the bottom of the hierarchy, for each level we identify the two clusters that are joined at the next level;
2. we compute a representative for each of the two clusters, by averaging the features of the members belonging to each of them;
3. we compute a distance between the two representatives using the above mentioned metrics (Cheng and Church for rows, and Euclidean distance for columns).

This process results in pairs of values (number of clusters and distance) which identify a series of points.

Step 3. Determining the number of clusters The crucial phase of the *L-method* presented in [11] consists in determining a suitable number of clusters. It is performed in four steps:

1. We consider the $n-1$ points (where n is the number of rows or columns) generated at the previous step. We choose a point c (which, in the first iteration, is equal to 2). This point divides the whole set of points in two subsets that, graphically represents two intervals: the left interval L_c, containing points $[1, c]$, the right interval R_c, containing points $[c+1, n-1]$.
2. For each of these subsets, we computed the line approximating them, using linear regression.
3. For each line, we compute the Root Mean Square Error (RMSE). We obtain a value for the line built on the left subset ($RMSE(L_c)$) and another one for the line built on the right subset ($RMSE(R_c)$).
4. Those values are then combined using the following formula [11]:

$$RMSE_c = \frac{c-1}{n-1} * RMSE(L_c) + \frac{n-c}{n-1} * RMSE(R_c)$$

We iterate these steps until $c = n-1$. The estimated number of clusters is then given by $\hat{c} = \min_c RMSE_c$. Clearly, this step is performed for both rows and columns.

4.2 Definition of constraints

To create sets of must-link and cannot-link constraints, we decided to use the information stored in Gene Ontology [8].

The Gene Ontology (GO) is a controlled vocabulary for the consistent description of attributes of genes and gene products maintained by the Gene Ontology Consortium. The ontologies are in the form of direct acyclic graphs whose nodes represent GO terms and edges represent the relationships between them. The nodes can be associated by five types of relationships: *is_a*, *part_of*, *regulates*, *positively_regulates* and *negatively_regulates*. This ontology is organized in three key domains that are shared by all organisms: **molecular function**, **biological process** and **cellular component**. These domains are represented by separate disconnected sub-graphs of the root node.

The *is_a* relationship is a class-subclass relationship, where A is_a B means that A is a subclass of B. Instead, it is defined C *part_of* D if whenever C is present, it is always a part of D, but C does not always have to be present. In other words, a child class is either a *part_of* the parent class or *is_a* more specific variant. The *regulates*, *positively_regulates* and *negatively_regulates* relationships describe interactions between biological processes and other biological processes, molecular functions or biological qualities. When a biological process E regulates a function or a process F, it modulates the occurrence of F. If F is a biological quality, then E modulates the value of F. In this work, we do not consider the cellular component domain.

We reformat the regulative information contained in GO in order to obtain a more concise representation of regulative relationships between the GO classes.

We built a weighted graph, where each node is a set of GO term linked together by *regulative* relationships. If there is at least one *is_a* relationship among GO terms in two different nodes we put an edge among these two vertices. The resulting graph contains 1537 GO macro-nodes. The weight associated to each edge is given by the number of *is_a* relationships existing among two nodes. By extracting the cliques in that graph, we obtain 202 cliques that represent strongly connected *regulative modules*.

Then, each gene of X is mapped into GO cliques following its GO annotation. Clearly, genes that are involved in multiple biological process/molecular functions, are likely to belong to more than one clique. To construct the set of must-link constraints, we perform the following steps: first, since we perform hard (non overlapping) co-clustering, we do not consider genes that belong to more than one clique; then those genes (among the remaining ones) belonging to one clique are associated to a unique transitive closure of must-link constraints.

Finally, to provide a set of cannot-link constraints, we consider all pair of cliques associated to the transitive closures of must-link constraints generated before. If they do not share any gene, then we set a cannot-link constraint between an arbitrary pair of genes belonging to the associated transitive closures.

5 Application

To evaluate the performance of our method, we used gene expression experiments for the organism *S. Cerevisiae* (yeast). This data set consists in 5 different microarray experiments (GEO accession series: GSE1312, GSE5301, GSE4660, GSE2224, GSE1723), downloaded from Gene Expression Omnibus (GEO[4]). In these experiments, yeast is treated with different types of stress. Gene expression levels are given as log (base 2) ratios of the measured level and a reference (control) level. All these experiments are hybridised on the same platform GPL90, Affymetrix Yeast Genome S98 Array. From GEO site we also extracted the gene ontology annotations files of each experiment.

5.1 Instantiation of the methodology

Following the steps described in the work-flow reported in Figure 1, we describe in details the instantiation of our approach:

- *Construction of matrix X* From the gene expression dataset, we build a matrix X with 9335 rows and 29 columns.
- *Selection of a suitable number of row/column clusters* We process matrix X using the method described in Section 4.1. We obtained a suggested number of 1677 row clusters ($k = 1677$) and 9 column clusters ($l = 9$).
- *Generation of a collection of constraints* We generated a collection of must-link constraints $\mathcal{M}$ and a set of cannot-link constraints $\mathcal{C}$ as described in Section 4.2. A total number of 2151 genes were constrained in 52 transitive closures of must-link constraints, and 39 cannot-link constraints.
- *Constrained Co-Clustering* Using the previously discovered k and l parameters and the collections $\mathcal{M}$ and $\mathcal{C}$ of constraints, we performed 40 trials of the co-clustering algorithm. The co-clustering process stops when $||X||^2_{t-1} - ||X||^2_t < 10^{-5}$, where $||X||^2_t$ and $||X||^2_{t-1}$ are the values of the objective function at iteration t and iteration $t-1$.

5.2 Validation of the results

To be able to assess the quality of the clustering results, we evaluated how accurate the partition over columns is w.r.t. the reference partition given by the GEO accession ID. To measure the accuracy we used the Normalized Mutual Information [12]. We denote by $\mathbf{C} = \{C_1 \dots C_J\}$ the partition built by the clustering algorithm on objects, and by $\mathbf{P} = \{P_1 \dots P_I\}$ the partition inferred by the original classification. J and I are respectively the number of clusters $|\mathbf{C}|$ and the number of classes $|\mathbf{P}|$. We denote by n the total number of objects. The Normalized Mutual Information (NMI) provides an information that is impartial with respect to the number of clusters [12]. It measures how clustering results

[4] http://www.ncbi.nlm.nih.gov/geo/

share the information with the true class assignment. NMI is computed as the average mutual information between every pair of clusters and classes:

$$\mathbf{NMI} = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} p_{ij} \log \frac{np_{ij}}{p_i p_j}}{\sqrt{\sum_{i=1}^{I} p_i \log \frac{p_i}{n} \sum_{j=1}^{J} p_j \log \frac{p_j}{n}}}$$

where p_{ij} is the cardinality of the set of objects that occur both in cluster C_j and in class P_i; p_j is the number of objects in cluster C_j; p_i is the number of objects in class P_i. Its values range between 0 and 1.

In our experiments, the column partitioning was quite stable: we achieved an average NMI of 0.8514 with a standard deviation of 0.0661. The best trial corresponds to a NMI value of 0.9199. We selected this trial for a detailed analysis on gene partition. The selected row partition contains 4 column clusters and 1310 row clusters. Among them, 4 clusters contain only genes involved by constraints, 1259 clusters contain only genes not involved by any constraints and 47 clusters contain a mix of constrained and unconstrained genes.

To assess the homogeneity of the discovered clusters, we use an *homogeneity* score, defined as the ratio of each involved GO term (for both biological process domain and molecular function domain) in each cluster. It takes values between 0 and 1, where a value of 1 means that all genes are involved in at least one common GO term. We consider that a cluster is consistent if the *homogeneity* of at least one domain is high, as a consequence we retained the maximum between the two domain values. The average score is 0.6105 (with a standard deviation of 0.3966), and in 70% of the cases it is greater than 0.50.

Low homogeneity clusters contain genes spread in multiple GO terms, but these terms might still be strongly connected each other. To verify this hypothesis, we performed a in-depth analysis on the 229 clusters which contain more than 5 and less than 50 genes. We found 4 clusters containing only genes involved by constraints, 191 clusters containing only genes not involved by any constraints and 34 clusters containing a mix of constrained and unconstrained genes. In most cases, the GO terms are involved in the same biological process: for instance in one cluster (containing 7 unconstrained genes; Gene IDs: UTP14, ERB1, RNT1, DBP6, RIX1, URB2, RPA12, UTP13), whose homogeneity value for the biological process domain is low, all GO terms are related to *ribosomal processing*. Moreover, another cluster (containing 6 unconstrained genes; Gene IDs: ALD6, DLD1, SNA2, MPM1, OPI3, MCR1 and 5 constrained genes; Gene IDs: SDH2, COQ10, SDH4, SDH3, SDH1) has a low homogeneity value and all its genes are related to *cellular respiration*.

Gene expression regulation is controlled by a complex network of interactions involving DNA cis-regulatory elements and transcription factors (TF). TFs control, by promoting or blocking, the transcription of genetic information from DNA to mRNA. Therefore, as we built (using our approach in Section 4.2) constraint sets from the GO regulation relationship, we expect the obtained clusters to contain genes belonging to specific transcriptional modules. Since in literature a strong dependency between the transcriptional units and the experimental conditions is proved, we checked how many clusters contain at least one TF. From

the YTF website[5], we extracted a list of yeast TFs, and we found that 101 of the 229 analyzed clusters contain at least one TF.

6 Conclusion

In this paper we presented an ontology-driven co-clustering approach for the identification of gene clusters characterized by similar expression profiles and involved in similar biological processes or functions. This leads to discovery more biological coherent and meaningful gene groups. Therefore, we provided a methodology to cluster genes following their expression signature in specific conditions with the help of constraints built over the Gene Ontology. The methodology is based on a constrained co-clustering algorithm, and automatically suggest a number of column/row clusters, as well as a congruent set of constraints.

Acknowledgments Francesca Cordero and Ruggero G. Pensa are co-funded by Regione Piemonte.

References

1. Eisen, M., Spellman, P., Botstein, P.B.D.: Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. USA **95** (1998) 14863–14868
2. Madeira, S., Oliveira, A.: Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Trans Comput Biol Bioinform. **1** (2004) 24–45
3. Hanisch, D., Zien, A., Zimmer, R., Lengauer, T.: Co-clustering of biological networks and gene expression data. Bioinformatics **18** (2002) S145–S154
4. Steinhauser, D., Junker, B., Luedemann, A., Selbig, J., Kopka, J.: Hypothesis-driven approach to predict transcriptional units from gene expression data. Bioinformatics **20** (2004) 1928–1939
5. Brameier, M., Wiuf, C.: Co-clustering and visualization of gene expression data and gene ontology terms for saccharomyces cerevisiae using self-organizing maps. J Biomed Inform. **40** (2007) 160–173
6. Pensa, R., Boulicaut, J.: Constrained co-clustering of gene expression data. In: Proceedings of SIAM SDM. (2008) 25–36
7. Cordero, F., Visconti, A., Botta, M.: A new protein motif extraction framework based on constrained co-clustering. In: Proceedings of the 24th Annual ACM Symposium on Applied Computing. (2009) 776–781
8. Ashburner, M. *et al.*: Gene ontology: tool for the unification of biology. the gene ontology consortium. Nat Genet. **25** (2000) 25–29
9. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proceedings ISMB 2000. (2000) 93–103
10. Cho, H., Dhillon, I.S., Guan, Y., Sra, S.: Minimum sum-squared residue co-clustering of gene expression data. In: Proceedings of the Fourth SIAM International Conference on Data Mining. (2004) 114–125
11. Salvador, S., Chan, P.: Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In: Proceedings of the 16th IEEE International Conference on Tools with AI. (2004) 576–584
12. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research **3** (2002) 583–617

[5] http://biochemie.web.med.uni-muenchen.de/YTFD/index.htm