J Glob Optim (2010) 47:343–354
DOI 10.1007/s10898-008-9349-z

A network flow model for biclustering via optimal re-ordering of data matrices

Peter A. DiMaggio Jr. · Scott R. McAllister · Christodoulos A. Floudas · Xiao-Jiang Feng · Joshua D. Rabinowitz · Herschel A. Rabitz

Received: 26 August 2008 / Accepted: 26 August 2008 / Published online: 13 September 2008
© Springer Science+Business Media, LLC. 2008

Abstract The analysis of large-scale data sets using clustering techniques arises in many different disciplines and has important applications. Most traditional clustering techniques require heuristic methods for finding good solutions and produce suboptimal clusters as a result. In this article, we present a rigorous biclustering approach, OREO, which is based on the Optimal RE-Ordering of the rows and columns of a data matrix. The physical permutations of the rows and columns are accomplished via a network flow model according to a given objective function. This optimal re-ordering model is used in an iterative framework where cluster boundaries in one dimension are used to partition and re-order the other dimensions of the corresponding submatrices. The performance of OREO is demonstrated on metabolite concentration data to validate the ability of the proposed method and compare it to existing clustering methods.

Keywords Biclustering · Mixed-integer linear optimization (MILP)

1 Introduction

Data clustering and data organization is an important problem which arises in many different disciplines. Some examples include pattern recognition [1], image processing [2], information retrieval [3], microarray gene expression [4], and protein structure prediction [5,6]. The goal of clustering is to extract specific patterns or trends from a data set by grouping together "similar" objects, where the definition of similarity is dependent upon the specific types of patterns one hopes to elucidate. Traditional clustering methods, such as hierarchical and partitioning clustering, are typically solved using heuristic methods and, as a result, produce suboptimal clusters since pairwise comparisons are evaluated locally. Several other

P. A. DiMaggio Jr. · S. R. McAllister · C. A. Floudas (✉)
Department of Chemical Engineering, Princeton University, Princeton, NJ 08544-5263, USA
e-mail: floudas@titan.princeton.edu

X.-J. Feng · J. D. Rabinowitz · H. A. Rabitz
Department of Chemistry, Princeton University, Princeton, NJ 08544-5263, USA

clustering techniques have also been developed, including model-based clustering [7,8], neural networks [9], simulated annealing [10], genetic algorithms [11,12], information-based clustering [13], decomposition based approaches [14–16], and data classification [17].

Another effective method for data analysis is to re-order the data based on minimizing the sum of the pair-wise differences between adjacent rows or columns in the final arrangement of the matrix. This problem is known as *rearrangement clustering* and the bond energy algorithm (BEA) was originally proposed as a method for finding "good" solutions. It was pointed out that the rearrangement problem could be cast as a traveling salesman problem [18,19]. The identification of cluster boundaries can be done after the re-ordering using a restricted partitioning approach for a pre-specified number of clusters [20] or during the re-ordering by solving the traveling salesman problem with k additional dummy cities [21]. Rearrangement clustering provides an appealing alternative to traditional clustering techniques since if the traveling salesman problem representation can be solved to optimality using deterministic methods, then one is guaranteed to have an optimal ordering of the data with respect to the proposed objective function.

The concept of biclustering has received considerable attention in the analysis of microarray gene expression data since a gene can be involved in more than one biological process or could belong to a group of genes that are coexpressed under a limited set of conditions [22]. Traditional clustering methods can fail to discover biclusters since they are comprised of submatrices of genes (row) and conditions (columns) of the original matrix. It should be noted that genes can be shared among biclusters (i.e., overlapping biclusters) and it is not a requirement that every gene is assigned to at least one bicluster.

Several different models and algorithms have been proposed for this NP-hard problem [23]. To ensure that biclusters are found in a reasonable amount of time, the existing techniques either utilize heuristic methods for finding good solutions or resort to simplifications in the problem representation (e.g., a simpler model or discretization of the expression level). For instance, the Cheng and Church [23] and cMonkey [24] biclustering algorithms are iterative processes which allow for integration of other data types since they do not transform the data. To solve the optimization problem based on the mean square residue [23], the Cheng and Church algorithm utilizes a greedy heuristic. Other methods for biclustering, such as plaid [22] and spectra models [25], are related to projection methods which regenerate the data matrix by biclusters. The plaid model uses a series of additive layers to capture the underlying structure in the gene expression data [22] and the spectra model uses singular value decomposition to identify eigenvectors that reveal the existence of a checkerboard pattern for the rearranged genes and conditions [25]. The biclustering methods BiMax [26] and SAMBA [27] discretize the expression level which allows them to produce biclusters in less time than more complicated models. To complement the assortment of problem representations for biclustering, there have been a variety of algorithmic approaches developed to solve these models of varying complexity, such as zero-suppressed binary decision diagrams [28], evolutionary algorithms [29,30], Markov chain Monte Carlo [24], bipartite graphs [27], and 0–1 fractional programming [31].

In this article, we present a biclustering algorithm based on optimally re-ordering data matrices in an iterative fashion. The cluster boundaries identified in the optimal re-ordering of one dimension are used to partition and re-order submatrices in the remaining dimension. We present several objective functions to assess the quality of the re-orderings and a network flow model for performing the physical row and column permutations of the data matrix. It is demonstrated that this global method provides a denser grouping of interrelated entities than other clustering methods and can reconstruct underlying fundamental patterns in the data.

2 Mathematical formulation

In this section, we present each of the components of the mathematical model: (1) the binary variables used to represent the row and column permutations, (2) the objective functions used to assess the quality of a re-ordering, and (3) a network flow model for performing the physical row and column permutations. An iterative framework is then presented which applies the re-ordering algorithm to submatrices to identify biclusters.

2.1 Variable definitions

We define the index pair (i, j) to correspond to the element in row i and column j of the data matrix. The value associated with this element is denoted as $a_{i,j}$. The total number of rows and columns of the matrix are represented by their cardinality, $|I|$ and $|J|$, respectively. It should be noted here that the row and column permutations can be performed independently, so we will only present the mathematical model for permuting the rows of the data matrix since an analogous representation follows for the columns.

The problem representation we adopt is the assignment of neighboring elements in the final arrangement of the rows, which is a boolean decision. To model this, we define the binary {0–1} variables, $y^{\text{row}}_{i,i'}$, to represent the assignment of row i adjacent to and above row i' in the final ordering, as presented below.

$$y^{\text{row}}_{i,i'} = \begin{cases} 1, & \text{if row } i \text{ is adjacent and above row } i' \text{ in the final arrangement} \\ 0, & \text{otherwise} \end{cases}$$

For instance, if the binary variable $y^{\text{row}}_{12,7}$ is equal to one then row 12 is above and adjacent to row 7 in the final ordering of the rows. However, if $y^{\text{row}}_{12,7} = 0$ then this simply implies that row 12 is *not* immediately above row 7 in the final ordering and provides no additional information regarding their relative positions in the re-arranged matrix.

2.2 Objective function

Given the problem representation of assigning neighboring elements in the final ordering, we need to develop an appropriate objective function to measure the cost or similarity associated with each assignment. In this section, we present a generalized objective function for quantifying the pair-wise similarity between two rows that are adjacent in the final ordering and illustrate common expressions that can be utilized. The general form of the objective function is presented in Eq. 1.

$$\sum_i \sum_{i'} y^{\text{row}}_{i,i'} \cdot \sum_j \phi(a_{i,j}, a_{i',j}) \tag{1}$$

In Eq. 1, the term $\phi(a_{i,j}, a_{i',j})$ represents the pair-wise similarity between rows i and i' for each column j. This term can assume a variety of functional forms, such as the relative difference in value for adjacent rows of a matrix, as shown in Eq. 2.

$$\sum_i \sum_{i'} y^{\text{row}}_{i,i'} \cdot \sum_j |a_{i,j} - a_{i',j}| \tag{2}$$

If it is desired to place the emphasis on penalizing large pair-wise differences, then one could use a squared difference form of the objective function, as shown in Eq. 3.

$$\sum_i \sum_{i'} y_{i,i'}^{\text{row}} \cdot \sum_j (a_{i,j} - a_{i',j})^2 \tag{3}$$

An alternative metric of similarity is to compute the root-mean squared deviation between each element j in rows i and i', as shown in Eq. 4.

$$\sum_i \sum_{i'} y_{i,i'}^{\text{row}} \cdot \sqrt{\frac{\sum_j (a_{i,j} - a_{i',j})^2}{|J|}} \tag{4}$$

It is important to note that the aforementioned objective functions are not restricted to these Euclidean forms; we simply presented these since they are commonly used. For instance, if the elements being clustered are amino acid residues, then a PAM or BLOSUM scoring matrix is the appropriate similarity metric since the Euclidean distance is meaningless for these characters. Our objective function can easily accommodate this type of metric. The objective functions can also be tailored to exploit physical trends in the data set. For instance, suppose it is known a priori that the values in the data set have an underlying monotonic landscape and this final ordering is desirable. Then it is straightforward to introduce restrictions for only incorporating those terms when the monotonicity property is violated (i.e., $a_{i,j} > a_{i',j}$). It should be noted that the objective functions presented above are symmetric.

2.3 Re-ordering rows and columns: a network flow model

The physical permutations of the rows and columns of the data matrix can be accomplished using a network flow model [32–37], where the rows correspond to nodes in the graph and an edge between two nodes (rows) indicates that they are neighbors in the final ordering. Recall that the binary variables $y_{i,i'}^{\text{row}}$ represent the assignment of row i' to be below and adjacent to row i in the final arrangement. Thus, in network flow terminology, the binary variable $y_{i,i'}^{\text{row}}$ represents the existence of an edge from row i to row i'. We define additional continuous variables to assign a flow value, $f_{i,i'}^{\text{row}}$, for every edge, $y_{i,i'}^{\text{row}}$.

$$f_{i,i'}^{\text{row}} \equiv \text{the flow from row } i \text{ to row } i'$$

It is important to note that the value of a flow *entering* a node (row) indicates its position in the final ordering. For instance, if $y_{12,7}^{\text{row}} = 1$ and $f_{12,7}^{\text{row}} = 3$, then row 7 is in position 3 in the final arrangement. Since the flows represent positions, we can assign general upper and lower bounds for all flow values since any flow connecting two rows i and i' (i.e., $y_{i,i'}^{\text{row}} = 1$) can never be greater than $|I| - 1$ nor less than 1.

$$y_{i,i'}^{\text{row}} \le f_{i,i'}^{\text{row}} \le (|I| - 1) \cdot y_{i,i'}^{\text{row}} \qquad \forall (i, i') \tag{5}$$

These constraint equations also ensure that if rows i and i' are not adjacent in the final ordering (i.e., $y_{i,i'}^{\text{row}} = 0$) then no flow is assigned (i.e., $f_{i,i'}^{\text{row}} = 0$). Since the variables $y_{i,i'}^{\text{row}}$ and $f_{i,i'}^{\text{row}}$ can only occur between two existing nodes, we need to define a fictitious *source* and *sink* node to indicate the first and final positions in the row orderings, respectively. Thus, we define another set of binary variables, $y\_source_i^{\text{row}}$ and $y\_sink_i^{\text{row}}$, to indicate which row is adjacent to the source (i.e., the top-most row) and which row is adjacent to the sink (i.e., the bottom-most row) in the final rearranged matrix, respectively.

$$y\_source_i^{\text{row}} = \begin{cases} 1, & \text{if row i is the top-most row in the final arrangement} \\ 0, & \text{otherwise} \end{cases}$$

$$y\_sink_i^{\text{row}} = \begin{cases} 1, & \text{if row i is the bottom-most row in the final arrangement} \\ 0, & \text{otherwise} \end{cases}$$

Additional flow variables are defined for the flow leaving the source node (entering the top-most row) and the flow entering the sink node (leaving from the bottom-most row), as shown below.

$$f\_source_i^{\text{row}} \equiv \text{the flow entering the source row } i$$
$$f\_sink_I^{\text{row}} \equiv \text{the flow leaving the sink row } i$$

It is essential that each row i has only one neighboring row above it and one neighboring row below it in the final rearrangement, as modeled by the constraints in Eqs. 6 and 7.

$$\sum_{i' \neq i} y_{i',i}^{\text{row}} + y\_source_i^{\text{row}} = 1 \qquad \forall i \tag{6}$$

$$\sum_{i' \neq i} y_{i,i'}^{\text{row}} + y\_sink_i^{\text{row}} = 1 \qquad \forall i \tag{7}$$

The constraints in Eq. 6 enforce that row i has some row i' adjacent to and above it in the final ordering or it is the top-most row. Equivalently, the constraints in Eq. 7 enforce that row i has some other row i' adjacent to and below it in the final ordering or it is the bottom-most row. Another constraint to consider is that there can be only one top-most and one bottom-most row in the final ordering, as modeled by Eqs. 8 and 9, respectively.

$$\sum_i y\_source_i^{\text{row}} = 1 \tag{8}$$

$$\sum_i y\_sink_i^{\text{row}} = 1 \tag{9}$$

The set of constraints defined by Eqs. 6 through 9 are sufficient for the assignment of unique neighbors for every row in the final ordering. However, *cyclic* arrangements of the rows can mathematically satisfy these constraint equations (i.e., it is possible that $y_{i,i'}^{\text{row}} = y_{i',i''}^{\text{row}} = y_{i'',i}^{\text{row}} = 1$, which results in a cyclic final ordering of $i, i', i'', i \ldots$, etc.). To enforce an acyclic arrangement, we need to ensure that the flow values, $f_{i,i'}$, which denotes the position of row i' in the final ordering, are unique and monotonically decreasing. We first define that the flow leaving the source node and entering the top-most row, f_source^{row}, is equal to the total number of rows ($|I|$), as presented in Eq. 10.

$$f\_source_i^{\text{row}} = |I| \cdot y\_source_i^{\text{row}} \qquad \forall i \tag{10}$$

Given an initial flow of $|I|$ from the source node to the top-most row, we would like the flow values for each subsequent row in the final arrangement to be $|I|-1$, $|I|-2$, and so on. This cascading property of the flow values will ensure a unique final ordering of the rows and can be accomplished using the flow conservation equation presented in Eq. 11.

$$\sum_{i'} (f_{i',i}^{\text{row}} - f_{i,i'}^{\text{row}}) + f\_source_i^{\text{row}} - f\_sink_i^{\text{row}} = 1 \qquad \forall i \tag{11}$$

Note that the positions (or flows) decrement by one at each node and provide a unique final ordering for the rows. Since we have defined the convention that $f\_source_i^{\text{row}}$ starts at $|I|$,

then $f\_sink_i^{\text{row}}$ must have a flow value of *zero* and thus can be eliminated from the above constraint. The set of constraints in Eqs. 6–11 comprise the mixed-integer linear programming (MILP) model for the permutations of the rows of a data matrix, according to any of the aforementioned objective functions. This MILP network flow model can be solved to global optimality using a branch-and-cut algorithm, such as those employed in CPLEX [38].

It should be noted here the re-ordering of the rows and columns can also be modeled as a traveling salesman problem (TSP), which can be solved using existing TSP solvers, such as Concorde [39]. In the TSP formulation, each row in the matrix is represented as a "city", $i \in |I|$, and there is an associated cost of "traveling" from all cities i to i'. The objective of the TSP problem is to visit each city (or row in the matrix) only once while incurring the minimum total cost. The order in which these rows are visited corresponds to their final positions in the re-ordered matrix. Since the TSP problem requires that the tour start and end at the same city, we introduce a dummy city to connect the top-most and bottom-most row in the final arrangement with an edge that does not have any cost.

2.4 Iterative framework

We apply the network flow model for optimal re-ordering that was presented in the previous section in an iterative fashion to bicluster data matrices, as illustrated by the flow diagram in Fig. 1. This iterative framework initially re-orders the data matrix using the network flow model over a single dimension, which we denote for reference as the "columns" of the matrix. For instance, in gene expression data this dimension would correspond to the set of conditions over which the expression levels were measured. Once the optimal re-ordering of the columns has been identified, we compute the median of the pair-wise objective function values (i.e., the median of $\phi(a_{i,j}, a_{i+1,j})$ over j for every i and $i+1$) between the neighboring columns in the final arrangement. The largest median values are selected as the *cluster boundaries* which divide the data matrix into several submatrices based on this dimension. For each submatrix, we then optimally re-order the rows over its subset of columns using the network flow model and define cluster boundaries for the re-ordered rows based on the largest medians in objective function value. This iterative framework of the network flow model is denoted as OREO, which is stands for the Optimal RE-Ordering of the rows and columns of a data matrix.

3 Computational study: metabolite concentration data

We applied the proposed biclustering method to a set of concentration profiles for 68 metabolites (the rows of the data matrix) that were dynamically recorded using liquid chromatography-tandem mass spectrometry under the conditions of nitrogen and carbon starvation (the columns of the data matrix) for the organisms *E. coli* and *S. cerevisiae* [40]. The starvation conditions (or columns) of the data matrix were optimally re-ordered using CPLEX in 168 seconds on Intel 3.0 GHz Pentium 4 processor for the objective function in Eq. 3. The results are presented in Fig. 2 where the top four cluster boundaries, illustrated by the solid vertical lines, partition the original matrix into the four submatrices A, B, C, D and E.

The most interesting feature of the column re-ordering is that the nitrogen and carbon starvation conditions are perfectly separated. That is, all the nitrogen starvation conditions occupy the left-half of the matrix and all the carbon starvation conditions occupy the right-half of the matrix, as shown in Fig. 2. This suggests that the proposed method has the ability

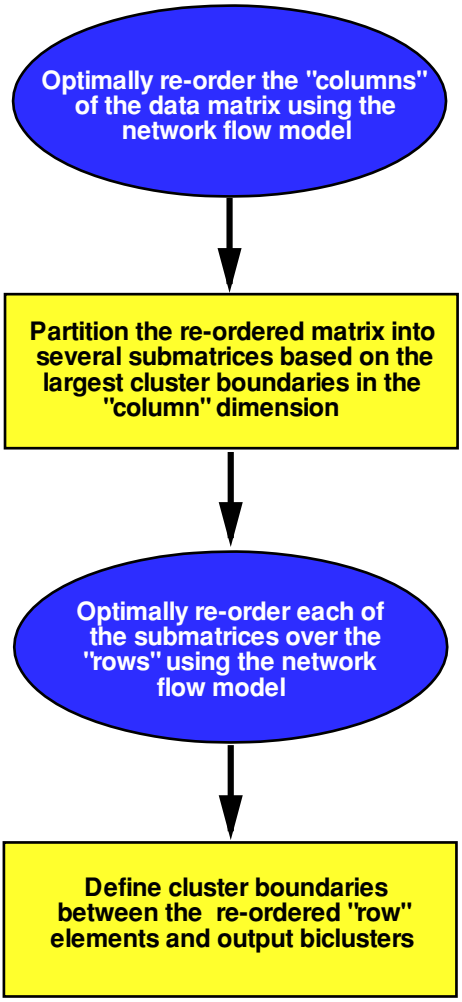

Fig. 1 Flow diagram for the iterative framework for biclustering, OREO. (1) The "columns" of the data matrix are first re-ordered to optimality using the network flow model, as shown in the blue circle at the top of the diagram. (2) Cluster boundaries in the re-ordered column dimension are used to partition the re-ordered matrix into non-overlapping submatrices. (3) Each of these submatrices is then re-ordered over the "row" dimension to optimality using the network flow model, and (4) cluster boundaries in this dimension are used to define the corresponding biclusters

to reconstruct underlying fundamental patterns. The next step in the iterative framework is to re-order over the metabolites for the submatrices A, B, C, D, and E but, for the sake of brevity, we shall only discuss the re-orderings over the regions labeled A and E. The optimal metabolite re-orderings for submatrices A and E were determined in 4085 and 4587 CPU seconds using CPLEX according to the objective function in Eq. 3 and the results are presented in the enlarged regions in Fig. 2.

It is interesting to note that the re-orderings of the metabolites over the different submatrices result in very dense groupings of metabolites corresponding to the same metabolic functions. For instance, the re-ordered metabolites in region A result in a dense grouping of the biosynthetic intermediate metabolites carbamoyl-aspartate, ornithine, dihydrooroate, *N*-acetyl-ornithine, IMP, cystathionine, and orotic acid in the first nine rows of the data matrix. This is consistent with the observation that most biosynthetic intermediates decrease in concentration over all starvation conditions based on the hypothesis that the cells turn off de novo biosynthesis as an early, strong, and consistent response to nutrient deprivation [40].

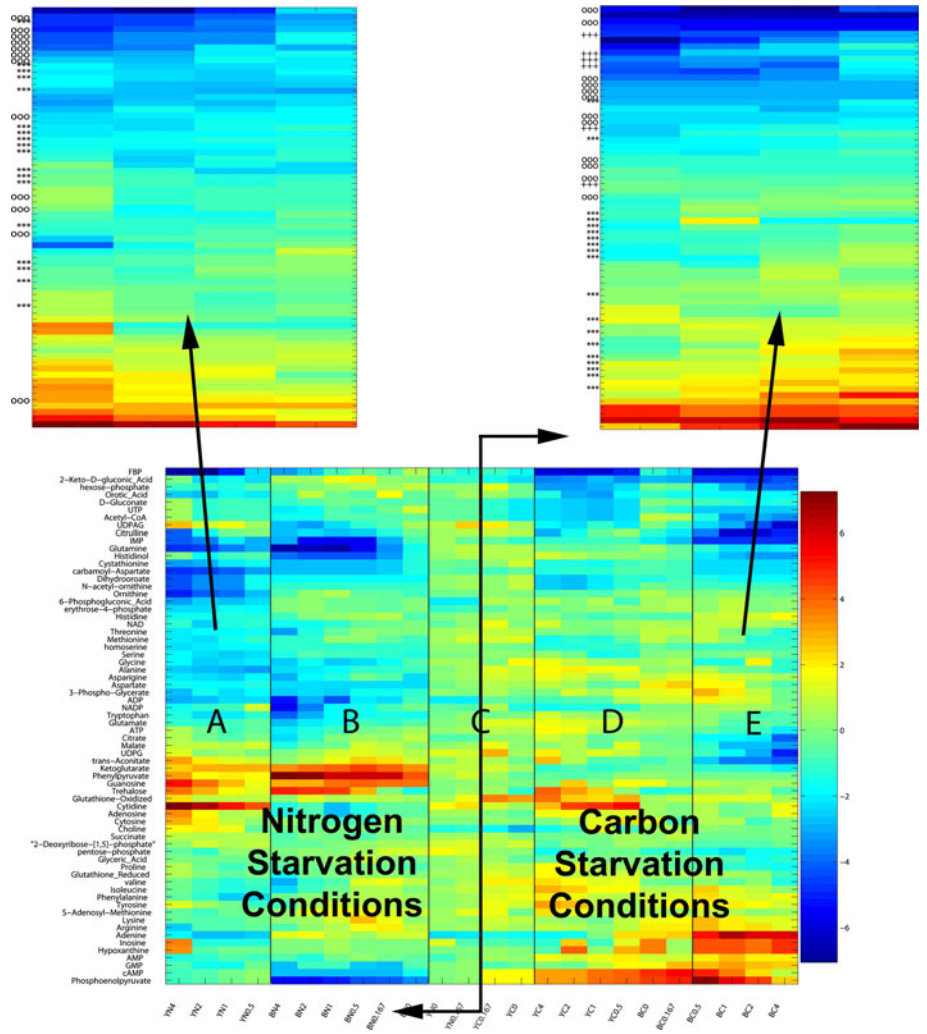

Fig. 2 Partitioning of columns into regions A, B, C, D, and E using cluster boundaries and optimal re-ordering results for metabolites for regions A and E. The relative groupings of metabolites are illustrated using the labels "***" for amino acid metabolites, "ooo" for biosynthetic intermediates, and "+++" for TCA compounds

There is also a strong aggregation of the amino acid metabolites glycine, asparagine, serine, alanine, methionine, threonine, histidine, aspartate, tryptophan, phenylalanine, isoleucine, and valine within a cluster of 26 metabolites. Another interesting arrangement in this submatrix is that the metabolites carbamoyl-aspartate and dihydrooroate are only separated by two positions and are both on the pyrimidine pathway [40].

The optimally re-ordered metabolites in region E result in a closer grouping of the amino acid and TCA cycle metabolites. Within a cluster of 27 metabolites, there are 16 amino acids (out of a possible 19 total in the data set) and 8 of these are grouped consecutively: serine, glycine, valine, glutamate, tryptophan, alanine, threonine, and methionine (see the "***" symbols in Fig. 2). This strong grouping of amino acid metabolites is consistent with

the observation that amino acids tend to accumulate during carbon starvation [40], which are the exactly the starvation conditions in this submatrix. The four TCA cycle metabolites trans-aconitate, citrate, malate, and acetyl-coA (out of the six total that are present in the data set) are within 6 positions of each other in the final arrangement. We again observe a strong grouping of the biosynthetic intermediates in the top half of the data matrix, which is consistent with the observation that most biosynthetic intermediates decrease in concentration under starvation conditions, but note that this clustering is not as strong as what was observed for region A. An interesting arrangement in the re-ordered metabolites for region E is that the metabolites FBP and phosphoenolpyruvate (PEP) are placed on opposite ends of the matrix (i.e., one is adjacent to the source row and the other is adjacent to the sink row). This relative placement makes sense since FBP is a positive regulator of pyruvate kinase, which is the major enzyme consuming PEP [40]. Since carbon-starvation resulted in a decrease of FBP, this presumably down-regulates the activity of pyruvate kinase, which in turn results in an accumulation of PEP.

3.1 Comparison with other clustering methods

To compare our results with those obtained from traditional clustering methods, we applied hierarchical clustering to the metabolite concentration data [40]. It was observed that the re-orderings obtained by OREO resulted in a closer grouping of metabolites of similar known metabolic function than hierarchical clustering. For instance, the largest consecutive grouping of amino acids for hierarchical clustering is alanine, glutamate, threonine, methionine, and serine, which are 3 less than the 8 total found by OREO in region E. The proposed method also results in a closer clustering of the 6 TCA cycle compounds. In general, OREO arranges the metabolites in an order which more closely reflects their known metabolic functions than does hierarchical clustering.

In order to quantify the actual deviation from the optimal ordering, we evaluated the objective function in Eq. 3 for the ordering reported by hierarchical clustering for both the rows (metabolites) and columns (starvation conditions), as shown in Table 1. The "Percent Gap" column in Table 1 is a standard measure in optimization for quantifying the deviation of given solution from optimality. One can see in Table 1 that the final ordering provided by the hierarchical results is suboptimal with respect to the squared difference objective function in Eq. 3.

Since the rearranged data appears to naturally form biclusters, we applied the biclustering algorithms ISA [41], Cheng and Church's [23], OPSM [42], BiMax [26], and SAMBA [27] to this metabolite concentration data set. Each algorithm was run using the default parameter values, which were only adjusted in the event that no biclusters were reported. The resulting biclusters were visualized using the BiVoc algorithm [43]. It was surprising to find that none of the biclustering methods were able to produce biologically insightful groupings of metabolites. The bicluster of highest annotated enrichment was found by Cheng and

Table 1 Comparison between optimal objective value and hierarchical objective value for metabolite concentration data for squared difference objective function

Problem	Optimal objective value	Hierarchical objective value	Percent gap (%)
Rows	4,415.8	6,377.6	30.8
Columns	1,753.0	2,677.9	34.5

Church's Algorithm [23], which assigned 15 amino acid metabolites to a large bicluster of 30 metabolites. The longest consecutive ordering of amino acids within this bicluster are serine, methionine, threonine, glutamate, and alanine, which is exactly the same as that reported in the hierarchical clustering results. The majority of the remaining metabolites in this bicluster are biosynthetic intermediates. The OPSM Algorithm [42] also produced a loosely correlated bicluster that contained the amino acid metabolites valine, isoleucine, alanine, phosphoenolpyruvate, ATP, proline, asparigine, and glutamate under the conditions of nitrogen and carbon starvation in *S. cerevisiae* and *E. coli*.

4 Discussion

In this article we presented a rigorous method for biclustering based on iteratively re-ordering the rows and columns of a data matrix. This algorithm, OREO, utilizes a network flow model to perform the row and column permutations according to a given objective function, which can assume a variety of functional forms and is a convenient option for the user to specify. The proposed approach was applied to metabolite concentration data and it was shown that our method results in a closer grouping of related metabolites than hierarchical clustering and other biclustering algorithms, which suggests that the optimal re-ordering has distinct advantages over a local re-ordering. It was also shown that OREO has the ability to separate objects into distinct groups, as was illustrated with the separation of the starvation conditions in the metabolite concentration data. It is noteworthy that OREO could be applied to clustering ensembles of conformers resulting from free energy calculations of oligopeptides [44–46] or proteins [47,48], de novo sequences generated in protein design [49,50], as well as design and scheduling of batch processes [51,52].

References

1. Anderberg, M.R.: Cluster Analysis for Applications. Academic Press, New York (1973)
2. Jain, A.K., Flynn, P.J.: Image segmentation using clustering. In: Ahuja, N., Bowyer, K. (eds.) Advances in Image Understanding: A Festschrift for Azriel Rosenfeld, pp. 65–83. IEEE Press, Piscataway (1996)
3. Salton, G.: Developments in automatic text retrieval. Science **253**, 974–980 (1991)
4. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. USA **95**, 14863–14868 (1998)
5. Zhang, Y., Skolnick, J.: SPICKER: a clustering approach to identify near-native protein folds. J. Comput. Chem. **25**, 865–871 (2004)
6. Mönnigmann, M., Floudas, C.A.: Protein loop structure prediction with flexible stem geometries. Protein: Struct. Funct. Bioinform. **61**, 748–762 (2005)
7. Edwards, A.W.F., Cavalli-Sforza, L.L.: A method for cluster analysis. Biometrics **21**, 362–375 (1965)
8. Wolfe, J.H.: Pattern clustering by multivariate mixture analysis. Multivariate Behav. Res. **5**, 329–350 (1970)
9. Jain, A.K., Mao, J.: Artificial neural networks: a tutorial. IEEE Comput. **29**, 31–44 (1996)
10. Klein, R.W., Dubes, R.C.: Experiments in projection and clustering by simulated annealing. Pattern Recognit. **22**, 213–220 (1989)
11. Raghavan, V.V., Birchand, K.: A clustering strategy based on a formalism of the reproductive process in a natural system. In: Proceedings of the Second International Conference on Information Storage and Retrieval, pp. 10–22 (1979)
12. Bhuyan, J.N., Raghavan, V.V., Venkatesh, K.E.: Genetic algorithm for clustering with an ordered representation. In: Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 408–415 (1991)
13. Slonim, N., Atwal, G.S., Tkacik, G., Bialek, W.: Information-based clustering. Proc. Natl. Acad. Sci. USA **102**(51), 18297–18302 (2005)

14. Tan, M.P., Broach, J.R., Floudas, C.A.: A novel clustering approach and prediction of optimal number of clusters: global optimum search with enhanced positioning. J. Glob. Optim. **39**(3), 323–346 (2007)
15. Tan, M.P., Broach, J.R., Floudas, C.A.: Evaluation of normalization and pre-clustering issues in a novel clustering approach: global optimum search with enhanced positioning. J. Bioinform. Comput. Biol. **5**(4), 895–913 (2007)
16. Tan, M.P., Smith, E.R., Broach, J.R., Floudas, C.A.: Microarray data mining: a novel optimization-based approach to uncover biologically coherent structures. BMC Biol. **9**, 268–283 (2008)
17. Busygin, S., Prokopyev, O.A., Pardalos, P.M.: An optimization based approach for data classification. Optim. Methods Softw. **22**(1), 3–9 (2007)
18. Lenstra, J.K.: Clustering a data array and the traveling-salesman problem. Oper. Res. **22**(2), 413–414 (1974)
19. Lenstra, J.K., Rinnooy Kan, A.H.G.: Some simple applications of the traveling-salesman problem. Oper. Res. Q **26**(4), 717–733 (1975)
20. Alpert, C.J., Kahng, A.B.: Splitting an ordering into a partition to minimize diameter. J. Classif. **14**, 51–74 (1997)
21. Climer, S., Zhang, W.: Rearrangement clustering: pitfalls, remedies, and applications. J. Mach. Learn. Res. **7**, 919–943 (2006)
22. Turner, H.L., Bailey, T.C., Krzanowski, W.J., Hemingway, C.A.: Biclustering models for structured microarray data. IEEE/ACM Trans. Comput. Biol. Bioinform. **2**(4), 316–329 (2005)
23. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proc. ISMB 2000, pp. 93–103 (2000)
24. Reiss, D.J., Baliga, N.S., Bonneau, R.: Integrated biclustering of heterogeneous genome-wide datasets for the inference of global regulatory networks. BMC Bioinform. **7**, 280–302 (2006)
25. Kluger, Y., Basri, R., Chang, J.T., Gerstein, M.: Spectral biclustering of microarray data: coclustering genes and conditions. Genome Res. **13**, 703–716 (2003)
26. Prelic, A., Bleuler, S., Zimmermann, P., Wille, A., Buhlmann, P., Gruissem, W., Hennig, L., Thiele, L., Zitzler, E.: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics **22**(9), 1122–1129 (2006)
27. Tanay, A., Sharan, R., Shamir, R.: Discovering statistically significant biclusters in gene expression data. Bioinformatics **18**, S136–S144 (2002)
28. Yoon, S., Nardini, C., Benini, L., De Micheli, G.: Discovering coherent biclusters from gene expression data using zero-suppressed binary decision diagrams. IEEE/ACM Trans. Comput. Biol. Bioinform. **2**(4), 339–354 (2005)
29. Bleuler, S., Prelic, A., Zitzler, E.: An EA framework for biclustering of gene expression data. In: IEEE Congress on Evolutionary Computation, pp. 166–173 (2004)
30. Divina, F., Aguilar, J.: Biclustering of expression data with evolutionary computation. Trans. Knowl. Data Eng. **18**(5), 590–602 (2006)
31. Busygin, S., Prokopyev, O.A., Pardalos, P.M.: Feature selection for consistent biclustering via fractional 0–1 programming. J. Comb. Optim. **10**, 7–21 (2005)
32. Ford, L.R., Fulkerson, D.R.: Flows in Networks. Princeton University Press, Princeton (1962)
33. Floudas, C.A., Grossmann, I.E.: Synthesis of flexible heat exchanger networks with uncertain flowrates and temperatures. Comput. Chem. Eng. **11**(4), 319–336 (1987)
34. Ciric, A.R., Floudas, C.A.: A retrofit approach for heat-exchanger networks. Comput. Chem. Eng. **13**(6), 703–715 (1989)
35. Floudas, C.A., Anastasiadis, S.H.: Synthesis of distillation sequences with several multicomponent feed and product streams. Chem. Eng. Sci. **43**(9), 2407–2419 (1988)
36. Kokossis, A.C., Floudas, C.A.: Optimization of complex reactor networks-II: nonisothermal operation. Chem. Eng. Sci. **49**(7), 1037–1051 (1994)
37. Aggarwal, A., Floudas, C.A.: Synthesis of general separation sequences—nonsharp separations. Comput. Chem. Eng. **14**(6), 631–653 (1990)
38. CPLEX.: ILOG CPLEX 9.0 User's Manual (2005)
39. Applegate, D.L., Bixby, R.E., Chvatal, V., Cook, W.J.: The traveling salesman problem: a computational study. Princeton University Press, Princeton (2007)
40. Brauer, M.J., Yuan, J., Bennett, B., Lu, W., Kimball, E., Bostein, D., Rabinowitz, J.D.: Conservation of the metabolomic response to starvation across two divergent microbes. Proc. Natl. Acad. Sci. USA **103**, 19302–19307 (2006)
41. Ihmels, J., Friedlander, G., Bergmann, S., Sarig, O., Ziv, Y., Barkai, N.: Revealing modular organization in the yeast transcriptional network. Nat. Genet. **31**, 370–377 (2002)
42. Ben-Dor, A., Chor, B., Karp, R., Yakhini, Z.: Discovering local structure in gene expression data: the order-preserving submatrix problem. In: Proceedings of the Sixth Annual International Conference on Computational Biology (RECOMB 2002), Washington, DC, USA, pp. 49–57 (2002)

43. Grothaus, G.A., Mufti, A., Murali, T.M.: Automatic layout and visualization of biclusters. Algorithms Mol. Biol. **1**, 1–15 (2006)
44. Androulakis, I.P., Maranas, C.D., Floudas, C.A.: Prediction of oligopeptide conformations via deterministic global optimization. J. Glob. Optim. **11**, 1–34 (1997)
45. Klepeis, J.L., Floudas, C.A.: Free energy calculations for peptides via deterministic global optimization. J. Chem. Phys. **110**, 7491–7512 (1999)
46. Klepeis, J.L., Floudas, C.A., Morikis, D., Lambris, J.D.: Predicting peptide structures using NMR data and deterministic global optimization. J. Comput. Chem. **20**(13), 1354–1370 (1999)
47. Klepeis, J.L., Floudas, C.A.: Ab initio tertiary structure prediction of proteins. J. Glob. Optim. **25**, 113–140 (2003)
48. Klepeis, J.L., Floudas, C.A.: ASTRO-FOLD: a combinatorial and global optimization framework for ab initio prediction of three-dimensional structures of proteins from the amino acid sequence. Biophys. J. **85**, 2119–2146 (2003)
49. Klepeis, J.L., Floudas, C.A., Morikis, D., Tsokos, C.G., Argyropoulos, E., Spruce, L., Lambris, J.D.: Integrated computational and experimental approach for lead optimization and design of compstatin variants with improved activity. J. Am. Chem. Soc. **125**(28), 8422–8423 (2003)
50. Fung, H.K., Floudas, C.A., Taylor, M.S., Zhang, L., Morikis, D.: Towards full sequence de novo protein design with flexible templates for human beta-defensin-2. Biophys. J. **94**, 584–599 (2008)
51. Lin, X., Floudas, C.A.: Design, synthesis and scheduling of multipurpose batch plants via an effective continuous-time formulation. Comput. Chem. Eng. **25**, 665–674 (2001)
52. Janak, S.L., Lin, X., Floudas, C.A.: Enhanced continuous-time unit-specific event based formulation for short-term scheduling of multipurpose batch processes: resource constraints and mixed storage policies. Ind. Eng. Chem. Res. **43**, 2516–2533 (2004)