2 A Survey on Fingerprint Classification Methods for Biological Sequences

BHASKAR DASGUPTA
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607
Email: dasgupta@cs.uic.edu

LAKSHMI KALIGOUNDER
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607
Email: lkalig2@uic.edu

2.1 INTRODUCTION

Since the discovery of the double helical structure of DNA, the molecular biology field has undergone a significant transformation via nucleic acids sequencing to determine genetic information at the most fundamental level. This revolution in biology has created a huge volume of data, estimate by many to grow at an exponential rate, by directly reading DNA sequences. One important reason for this exceptional growth rate of biological data lies in the medical use of such information in the design of therapeutics. Naturally, such a large amount of data poses a serious challenge in storing, retrieving and analyzing biological information.

In this chapter, we provide a survey of a classification problem involving genetic sequences, namely the problem of classifying fingerprint vectors with missing values. Oligonucleotide fingerprinting is a powerful DNA array based method to characterize cDNA and ribosomal RNA (rDNA) gene libraries, and has many applications such as gene expression profiling and DNA clone classification. For example, Herwig *et al.* [18] used oligonucleotide fingerprinting as an efficient and fast approach to extract parallel gene expression information about all genes that are represented in

a cDNA library from a specific tissue under analysis. The information obtained by monitoring gene expression levels in different development stages, tissue types, clinical conditions and different organisms can fuel a understanding of gene function and gene networks, and may assist in diagnostics of disease conditions and effects of medical treatments.

The main focus of this chapter is motivated by the recent development of a *discrete* classification approach by Figueroa, Borneman, and Jiang in 2004 [11], called the Binary Clustering with Missing Values (BCMV) problem, for analyzing oligonucleotide fingerprints, especially in applications such as DNA clone classifications. In this approach, fingerprint data were first normalized and binarized using control DNA clones. Because there may exist unresolved ("missing") values in the binarization process, they formulated the classification of (binary) oligonucleotide fingerprints as a combinatorial optimization problem that attempted to identify clusters and resolve the missing values in the fingerprints *simultaneously*.

The rest of the chapter is organized as follows. In section 2.2 we state some basic mathematical definitions that will be useful in understanding the underlying computational problems more effectively. In Section 2.3 we provide a brief survey of various other classification approaches to provide the reader with a global perspective, and in Section 2.4 we provide a brief overview of several approaches for estimating missing values in the genomic data. In Section 2.5 we survey in more details the BCMV problem and its variations. We assume that the reader is familiar with standard textbook concepts of algorithmic complexity theory such as found in [8, 23].

2.2 BASIC DEFINITIONS AND PROBLEM STATEMENTS

Fingerprint Formally, we define a fingerprint vectors (in short, fingerprint) as a vector with each component (element) from $\Sigma \cup \{N\}$, for some finite alphabet Σ not containing the symbol N, that consists of the hybridization intensity values between the clone and each probe. The value N in a component of the vector corresponds to a component with missing values. The number of elements of a fingerprint is its *length*.

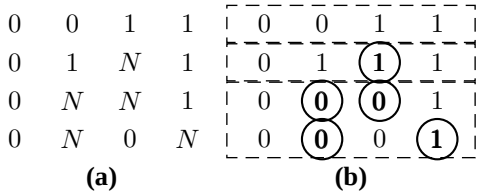


Fig. 2.1 (a) Four fingerprint vectors $\Sigma = \{0, 1\}$. (b) A possible resolution of them. Each compatible fingerprint group is enclosed by a dashed rectangle.

Oligonucleotide probe A short DNA sequence (usually 8–50 bases) which is applied to hybridize with the clones.

Compatible fingerprints Two fingerprint vectors $f_1 = \langle f_1[1], f_1[2], \ldots, f_1[\ell] \rangle$ and $f_2 = \langle f_2[1], f_2[2], \ldots, f_2[\ell] \rangle$ are *compatible* if for any position i where they differ, at least one of $f_1[i]$ and $f_2[i]$ is equal to N. See Fig. 2.1 for an illustration.

Resolved vector A vector r is called *resolved vector* of a fingerprint vector f if it is identical with f on all positions having an alphabet from Σ in f and has a symbol from Σ in each position of f that had the symbol N.

2.3 AN OVERVIEW OF VARIOUS CLASSIFICATION APPROACHES

Classification approaches are not very new to biologists; hierarchical classification has been used for a long time to create taxnomic ranks (kingdom, phylum, class, order, family, genus and species) of all living things. However, with the arrival of fast computational tools and large amounts of genetic information, classification and clustering approaches have increased their applicability considerably to efficiently analyze the genomic data. Classification and clustering remains, in general to a certain extent, an art since there are no universally agreed-upon criteria for evaluating solutions, and there is no ultimate algorithm. In this section, we briefly review a few classification approaches that have been used in the past in bioinformatics; for a more comprehensive treatment, see, for example, [24].

Shamir and Sharan in [26] discuss some algorithmic approaches for clustering *gene expression data*. A key step in the analysis of gene expression data is the identification of groups of genes that manifest similar expression patterns. The goal is to partition the elements into subsets, which are called *clusters*, so that two criteria are satisfied: *homogeneity* (elements in the same cluster are highly similar to each other), and *separation* (elements from different clusters have low similarity to each other).

In *hierarchical* classification approach, the solutions are typically represented by a dendogram. Algorithms for generating such solutions often work either in top-down manner, by repeatedly partitioning the set of elements, or in a bottom-up fashion.

k-means [2, 21] is another classical classification approach. It assumes that the number of clusters K is known, and aims to minimize the distance between elements and the centroids of their assigned clusters. The HCS [16, 17] and CLICK [25] algorithms use a similar graph theoretic approach for classification. The input data is represented as a similarity graph. The algorithm recursively partitions the current set of elements into two subsets. Before a partition, the algorithm considers the subgraph induced by the current subset of elements. If the subgraph satisfies a stopping criterion, then it is declared a kernel. Otherwise, a minimum weight cut is computed in that subgraph, and the set is split into the two subsets separated by that cut. The output is a list of kernels that serve as a basis for the eventual clusters. HCS and CLICK differ in the similarity graph they construct, their stopping criteria, and the post-processing of the kernels. In another graph-theoretic approach, Ben-Dor *et al.* [4] developed a polynomial algorithm called CAST (Clustering Affinity Search Technique) for finding true clustering with high probability. The correct cluster structure is represented by a graph that is a disjoint union of cliques, and errors are subsequently introduced in the graph by independently removing and adding edges between pairs of vertices with some probability. If all clusters are of size at least $\Omega(n)$, the algorithm solves the problem to a desired accuracy with high probability.

Self-Organizing Maps (SOM) [20] were developed as a method for fitting a number of ordered discrete reference vectors to the distribution of vectorial input samples. A SOM assumes that the number of clusters is known.

To summarize, the hierarchical method gives an overall view of the structure without an attempt to force a hard classification, whereas the other methods aim to split the universe of elements into clusters, either by geometric approaches that move cluster centers (SOM, k-means) or by graph-theoretic approach. The last approach may take a global view (CLICK) or single out one affinity-stable cluster at a time (CAST).

2.4 MISSING VALUE ESTIMATION METHODS

The value of N in the sequence for fingerprint classification corresponds to some unknown (missing) spots on the sequence during the laboratory process due to various factors (*e.g.*, machine error, gene expression microarray experiments generating data sets with multiple expression values, insufficient resolution in microarray experiments etc.). There are many options for dealing with missing values, each of which reaches drastically different results.

Ignoring missing values is obviously the simplest method and is frequently applied. This approach however has its flaws. Because it is often very costly or time consuming to repeat the experiment, molecular biologists, statisticians and computer scientists have investigated the possibility of recovering the missing gene expression values by *ad-hoc* or systematic methods. Methods like hierarchical clustering and k-*means* clustering are not robust against missing data, and may loose effectiveness even with a few missing values. Other standard supervised statistical microarray analysis techniques such as support vector machine classification, principal component analysis, or singular value component analysis often may not be applicable to data set with missing values. Thus methods for imputing missing data are needed.

One solution to deal with the missing values is to do the same experiment and replicate the data. This extra labor work strategy has been used in many experimental scientists and wet laboratories so far. If the cost of the experiment is not expensive, it may be a practical solution, but certain type of experiments such as patient specific time course experiments are very expensive or may even be impossible to be reproduced. Less labor work and simple tentative solution is to fill the missing values by zeroes, average of the gene expressions, or average of overall expression values.

Two recent popular methods of imputing missing values are the *KNNimpute* method [28] and the *LLSimpute* method [3, 19] that uses the k-nearest neighbor clustering, least square and Bayesian optimization. The basic strategy of this type is to find similar expression patterns having missing values by clustering methods, and then to predict the missing value from the corresponding values in the same cluster. In these two methods, the recovery of missing data is done independently, *i.e.*, the estimation of each missing entry does not influence the estimation of other missing entries. Another approach is to use high rank Eigengenes in a hidden concept space to predict the missing value. Representative methods of this type are the

SVDimpute method [1] that uses singular value decomposition, and the *BPCAimpute* method [22] that used principal component analysis and Bayesian optimizations. The basic strategy of this type of approach is to find bases of expression space, and then to reconstruct a matrix with the dominant bases. During the reconstruction process, the missing values are filled. The basis is called *Eigengene*, and Eigengene shows a gene expression fluctuation which is orthogonal to each other in an expression pattern space. Another approach similar to *SVDimpute* and *BPCAimpute* type prediction is the Fixed Rank Approximation Algorithm (FRAA) of Friedland, Niknejad and Chihara [13] to predict missing entries by using Eigengenes. In this approach the estimation of missing entries is done simultaneously, *i.e.*, the estimation of one missing entry influences the estimation of the other missing entries. They showed that FRAA is more accurate than replacing missing values with zeroes or with row means. FRAA by itself is a very useful tool for gene data analysis without using clustering methods. The number of high rank Eigengene should be close to the rank of the perfect matrix, but it is hard to guess the correct number of high rank Eigengenes from a data with missing entries. To find the optimal number, *BPCAimpute* uses Bayesian statistics while *SVDimpute* uses a given fixed number. FRAA also requires the fixed number of major Eigengenes, but the uniqueness of FRAA is that it has an iteration process which can increase the importance of these high rank Eigengenes in a reconstructed matrix on each step. However, it is still difficult to guess the correct number of Eigengne or rank of perfect matrix and therefore even FRAA itself is powerful but not useful in a practical case. The other drawback of FRAA is that the result heavily deepens on initial tentative values for missing entries. Friedland *et al.* [14] suggested a hybrid method IFRAA (Improved FRAA) which is a combination of FRAA and a good clustering algorithm.

There is no general consensus about which type of algorithms is better. Past experiments in [19, 22] suggest that *BPCAimpute* and *LLSimpute* predict generally better than the others, and the performances of these two methods are almost comparable with depending on data sets. Troyanskaya *et al.* [28] observed that *KNNimpute* is more robust and sensitive method for missing value estimation than *SVDimpute* and both *SVDimpute* and *KNNimpute* surpass the commonly used row average method. Gan, Liew and Yan [15] proposed a hybrid approach called POCS (Projection Onto Convex Set), which is the best combination of *SVDimpute* and *KNNimpute*. They experimentally showed that POCS achieves a reduction of 16% to 20% error than *KNNimpute* and *SVDimpute*. The FRAA method has been used by several computational biologists and experimental results on various data sets shows its robustness. To further improve upon the FRAA approach, one needs to combine it with an algorithm for gene clustering. A possible implementation is as follows. First, apply FRAA to the corrupted data set. Next, using this estimated data set, partition the genes into clusters of genes with similar traits. Now apply FRAA again to the missing entries of genes in each cluster.

In the next section, we survey in more details a combinatorial approach to determining missing values originally proposed by Figueroa, Borneman, and Jiang [11], the main focus of this chapter.

2.5 FINGERPRINT CLASSIFICATION: A COMBINATORIAL APPROACH FOR ESTIMATING MISSING VALUES

In this approach, called Binary Clustering with Missing Values (BCMV) approach, fingerprint data are first normalized and binarized using control DNA clones. To resolve the missing values in the binarization process, Figueroa *et al*. formulated the classification of (binary) oligonucleotide fingerprints as several combinatorial optimization problems as described below that attempt to identify clusters and resolve the missing values in the fingerprints *simultaneously*. They studied the computational complexity of these problems and their parameterized versions where the maximum number of N's in a fingerprint vector is bounded by an integer parameter p.

In the following problem formulations, we assume that $\Sigma = \{0, 1\}$.

Binary Clustering with Missing Values (BCMV) The problem of clustering with p missing values (CMV(p) for short) is to partition a set F of n fingerprint vectors, each of of length ℓ with at most p symbols that are N, into *disjoint* subsets $F_1, F_2, \ldots, F_k$ such that, for each $1 \leq i \leq k$, any two fingerprints in F_i are compatible. The objective is to *minimize* the number of partitions. Intuitively, the CMV problem aims to resolve the fingerprints using the minimum number of resolved vectors.

Inside Edge Binary Clustering with Missing Values (IEBCMV) The problem of inside compatible clustering with p missing values (IEBCMV(p) for short) is defined analogously except that the number of compatible pairs of vectors within the same partition is maximized instead of the minimization of the cardinality of the partition. That is, the objective now is to *maximize* the number of co-clustered pairs of fingerprints.

Outside Edge Binary Clustering with Missing Values (OEBCMV) The problem of outside compatible clustering with p missing values (OEBCMV(p) for short) is again defined analogously except that now the number of compatible pairs of vectors belonging to different clusters is minimized. That is, the new objective is to *minimize* the number of pairs of compatible fingerprints assigned to different clusters.

2.5.1 Algorithmic Complexity Results

BCMV(p)

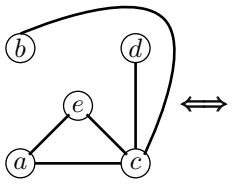
BCMV(p) was first considered and motivated in [11]. For arbitrary p, the following strong inapproximability result can be shown.

Theorem 2.1 [9] *For any constant $0 < \varepsilon < 1$ and unrestricted p, BCMV(p) cannot be approximated to within a ratio of $n^{1-\varepsilon}$ unless* NP $\subseteq$ ZPP.

Sketch of Proof. In the standard graph coloring problem, the goal is to produce an assignment of colors to vertices of a given graph $G = (V, E)$ such that no two

adjacent vertices have the same color and the number of colors is *minimized*. Let $\chi^*(G)$ denote the minimum number of colors in a coloring of G. The following inapproximability result is known [10]: for any constant $0 < \varepsilon < 1$, $\chi^*(G)$ cannot be approximated to within a factor of $|V|^{1-\varepsilon}$ unless $\mathsf{NP} \subseteq \mathsf{ZPP}$.

Given an instance $G = (V, E)$ with n vertices and m edges, one can construct an instance of BCMV(p) in the following manner. There is a sequence f_v of length m for every node v of G. Consider any arbitrary ordering of the m edges of G. For the i^{th} edge in the order, say $\{u, v\}$, we have $f_u[i] = 0$, $f_v[i] = 1$, and $f_x[i] = N$ for every $x \in V \setminus \{u, v\}$. See Fig. 2.2 for an illustration. The proof can then be completed by showing that G can be colored with y colors if and only if BCMV(p) outputs a solution with y partitions. □



	$\{a,c\}$	$\{a,e\}$	$\{c,e\}$	$\{c,d\}$	$\{b,c\}$
a	**0**	**0**	***N***	***N***	***N***
b	***N***	***N***	***N***	***N***	**0**
c	**1**	***N***	**0**	**0**	**1**
d	***N***	***N***	***N***	**1**	***N***
e	***N***	**1**	**1**	***N***	***N***

Fig. 2.2 An illustration of the reduction in Theorem 2.1.

However, in practice the number of N's in a binarized fingerprint vector is often upper bounded by a small constant, depending on the quality of hybridization intensity values and choice of control clones. Thus, it behooves to look at the problem with restricted values of p. Figueroa, Borneman and Jiang [11] showed the problem to be NP-hard even when $p = 3$, and polynomial-time solvable when $p = 1$. The polynomial-time solvability for BCMV(1) was shown by reducing it to vertex cover problem on *bipartite graphs*, and observing that the later problem is well-known to be solvable in polynomial time by matching techniques. Figueroa *et al.* in 2005 [12] further showed that BCMV(2) is NP-hard by giving a reduction from the minimum vertex cover problem on planar, cubic, 3-connected and triangle-free graphs, which is known to be NP-hard [29], to the BCMV(2) problem.

In a subsequent paper, Bonizzoni, Della Vedova, Dondi and Mauri [7] showed some improvements in closing the gaps between the known lower bounds and upper bounds on the approximability of variants of the original problem proposed in [12]. They showed that, even when each fingerprint contains only two unknown positions, BCMV(2) is APX-hard[1] by giving an L-reduction from minimum vertex cover on cubic graphs which is known to be APX-hard. In particular, to prove that BCMV(2) is APX-hard, they combined two L-reductions: the first one from the minimum vertex cover problem on a graph G to the minimum vertex cover on a graph gadget G' and the second one from the minimum vertex cover problem on a graph gadget G' to BCMV(2).

As the proof of Theorem 2.1 suggests, BCMV(p) can be easily formulated as one of finding a *minimum clique partition* of a graph in the following manner:

[1]A problem that is APX-hard cannot be approximated within a factor of $(1+\varepsilon)$, for some positive constant $\varepsilon > 0$, in polynomial-time unless $\mathsf{P} = \mathsf{NP}$.

> Given a set of fingerprint vectors F, define a graph $G_F = (F, E_F)$ where two nodes (fingerprints) are adjacent if and only if they are compatible.

The graph G_F is known as the *compatibility graph* of F. Hence BCMV of F is equivalent to the problem of finding a minimum clique partition (MCP) on G_F. However, it is also well-known that finding MCP of a graph is in general an NP-hard problem. Nonetheless, based on such a reformulation, The authors in [11, 12] presented efficient algorithmic approaches for BCMV(p) by taking advantage of some unique properties of the graph G_F, resulting in several results such as the following.

- There exists a greedy algorithm with an approximation ratio of $\min\{1 + \ln n,\, 2 + p\,\ln \ell\}$ that can be implemented to run in $O\left(n\ell\, 2^p\right)$ time. For $p = O(\log n)$ this approximation algorithm runs in polynomial time.
- There exists a polynomial-time heuristics that achieves an approximation ratio of 2^p.
- They provide a practical greedy heuristics based on iterating on building the largest possible cluster that has a worst-case running time of $O\left(p^{2^p} n^2\right)$. Since p is usually small compared to n in practice, the running time of the algorithm is practically efficient. To find a small clique partition of G_F one keeps on finding unique *maximal* cliques and removing it from the graph (and updates the graph accordingly) until no unique maximal cliques can be found. Then, a greedy action takes place by removing a maximum clique from the graph and the same process is repeated until all vertices of G_F have been included in some clique.

IEBCMV(p) and OEBCMV(p)

These two variants of the original optimization problems in [12], introduced in [7], aim to solve the fingerprint classification problem based on slightly different optimization criteria. The first variant, termed as the problem of inside compatible clustering with at most p missing values (IECBMV(p) for short) is defined analogously to BCMV(p) with the exception that the number of compatible pairs of vectors within the same clusters is maximized instead of the minimization of the cardinality of the partition. The second variant, termed as the problem of outside compatible clustering with at most p missing values (OECBMV(p) for short) is again defined analogously to BCMV(p) with the exception that now the number of compatible pairs of vectors belonging to different clusters is minimized. Various results on these problems that were reported in the papers [7, 12] include the following.

- It was shown in [12] that, when $p = O(\log n)$, IECMV(p) can be approximated in polynomial time within a factor of 2^{2p-1}, whereas in the special case when no two compatible vectors have N at the same position OECMV(p) can be approximated in polynomial time within a ratio of $2\left(1 - 2^{-2p}\right)$. To obtain these results, they reduced IECBMV(p) and the restricted version of OECBMV(p)

to special variants of maximum and minimum satisfiability problems which yielded polynomial-time constant-factor approximations for both problems.

They showed that IECBMV(p) can be expressed as a variant of the maximum satisfiability problem where the Boolean formula is in disjunctive normal form (DNF). By a result of Trevisan [27], the maximum satisfiability problem for DNF formulas with conjunctive clauses of length at most k admits a polynomial-time approximation algorithm with an approximation ratio of 2^{k-1} which leads to an approximation ratio of 2^{2p-1} for the IECBMV(p) problem.

By taking the negation of the above DNF formula and applying De Morgan's laws, they obtained a formula Φ in conjunctive normal form (CNF) with clauses of length at most $2p$. Now, the problem of finding the minimum number of clauses in Φ that can be simultaneously satisfied is easily seen to be equivalent to the OECBMV(p) problem. Furthermore, if no two compatible vectors contain N at the same position, then there is 1-1 correspondence between satisfied clauses and compatible pairs of fingerprints that are in different clusters. Using the fact that the problem of minimum k-satisfiability admits a polynomial time approximation algorithm with an approximation ratio of $2\left(1-2^{-k}\right)$ [6], they obtained an approximation ratio of $2\left(1-2^{-2p}\right)$ for the so restricted version of the OECBMV(p) problem.

- Further improvements of the results of [12] are reported in a subsequent paper [7]. Here, they proved that both of these problems are APX-hard. The APX-hardness of IECBMV(2) is obtained via an L-reduction from maximum independent set on 3-regular graphs which is known to be APX-hard [5], and their results show that it is NP-hard to approximate IECBMV(2) with a ratio better than $1+\frac{1}{3479}$.

  On the positive side, these authors presented a fixed-parameter tractable approximation algorithm whose running time is $O\left(2^p n^3 \ell\right)$, and achieved an approximation ratio of 2. Despite the hardness of these restricted versions of the problem, they also showed that the general clustering problem on an unbounded number of missing values such that these missing values occur for every fixed position of at most one input fingerprint vector can be solved in polynomial time. Finally, they gave a polynomial-time algorithm for solving the BCMV(p) problem for the special case where, for each position of a fingerprint vector, there is at most one fingerprint with an N symbol in such position. They denoted such a restriction by 1-BCMV and showed that their proposed algorithm run in $O\left(n^2 \ell\right)$ time.

2.5.2 Experimental Results

The experimental results on simulation and real data demonstrated that the greedy heuristics in [11] run faster and perform better (in the context of DNA clone classification) than popular clustering methods such as UPGMA, CLUSTER and CLICK. If the ratio between the largest and the smallest intensity values is above some pre-

specified threshold, the intensity of the clone was considered as a missing value, and the reliability of hybridization intensities were evaluated using clones spotted twice. The results on real data from the classification of microbial rDNA clones suggested that this discrete approach is more accurate than clustering methods based on real intensity values in terms of separating clones that have different characteristics with respect to the given oligonucleotide probes. An important advantage of the discrete approach was that binarized fingerprints were essentially reproducible whereas (normalized) real intensity values were generally not.

2.5.3 Open Problems for Future Research

As observed in [7, 12], several open problems remains for future on the algorithmic complexity side. For example:

- Is there a constant factor approximation algorithm for OECBMV(p) in the general case, and a non-trivial approximation ratio for greedy heuristics for IECMV(p)? Can we discover any non-trivial relationship between the various problem BCMV(p), IECBMV(p) and OECBMV(p) in terms of their hardness of approximation? Some experimental works could be helpful for this purpose to develop intuitions about this the special structure of the input data.

- Naturally, one could relate resolving the fingerprint vectors with construction of the phylogenetic trees of the corresponding resolved sequences. For instance, a natural objective could be to find an assignment to the N-positions which will yields phylogenetic trees optimizing a specific evolutionary objective (*e.g.*, perfect phylogeny, phylogenetic tree of minimum size or a minimum number of mutations etc.). After the rDNA clone libraries are constructed, the clones can classified by individual hybridization experiments on DNA microarrays with a series of short DNA oligonucleotides into clone types or operational taxonomic units (OTUs), where an OTU is a set of DNA clones sharing the same set of oligonucleotides that have successfully hybridized. Once classified, the nucleotide sequence of representative clones from each OTU can then be obtained by DNA sequencing to provide phylogenetic descriptions of the microorganisms.

Acknowledgments

We thank Paola Bonizzoni and Riccardo Dondi for useful discussions. This work was partially supported by NSF grant DBI-1062328.

REFERENCES

1. O. Alter, P. O. Brown and D. Botstein. *Singular value decomposition for genome-wide expression data processing and modeling*. proceedings of the National Academy of Sciences USA, 97, 10101-10106, 2000.

2. C. H. Ball and D.J. Hall. *A clustering technique for summarizing multivariate data*, Behavioral Sciences, 12, 153-l55, 1967.

3. T. H. Be, B. Dysvik and I. Jonassen. *LSimpute: accurate estimation of missing values in microarray data with least squares methods*, Nuclear Acids Research, 32 (3), e34, 2004.

4. A. Ben-Dor, R. Shamir and Z. Yakhini. *Clustering Gene Expression Patterns*, Journal of Computational Biology, 6 (3-4), 281-297, 1999.

5. P. Berman and M. Karpinski. *On some tighter inapproximability results*, 26th International Colloquium on Automata, Languages, and Programming, 200-209, 1999.

6. D. Bertsimas, C-P. Teo, and R. Vohra. *On dependent randomized rounding algorithms*, in proceedings of the 5th International Conference on Integer Programming and Combinatorial Optimization, 330-344, 1996.

7. P. Bonizzoni, G. Della Vedova, R. Dondi and G. Mauri. *Fingerprint Clustering with Bounded Number of Missing Values*, Algorithmica, 58 (2), 282-303, 2010.

8. T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein. *Introduction to Algorithms*, The MIT Press, 2001.

9. B. DasGupta and R. Dondi. *Some improved inapproximability results for fingerprint classification*, manuscript, 2011.

10. U. Feige and J. Kilian. *Zero Knowledge and the Chromatic Number*, Journal of Computers & System Sciences, 57 (2), 187-199, 1998.

11. A. Figueroa, J. Borneman, and T. Jiang. *Clustering binary fingerprint vectors with missing values for DNA array data analysis*, Journal of Computational Biology, 11 (5), 887-901, 2004.

12. A. Figueroa, A. Goldstein, T. Jiang, M. Kurowski, A. Lingas and M. Paterson. *Approximate clustering of fingerprint vectors with missing values*, in proceedings of the 2005 Australasian symposium on Theory of computing, 41, 57-60, 2005.

13. S. Friedland, A. Niknejad and L. Chihara. *A simultaneous reconstruction of missing data in DNA microarrays*, Linear Algebra and its Applications, 416 (1), 8-28, 2006.

14. S. Friedland, A. Niknejad, M. Kaveh and H. Zare. *An algorithm for missing value estimation for DNA microarray data*, Computer Engineering, 1-9, 2005.

15. X. Gan, A. W.-C. Liew and H. Yan. *Missing microarray data estimation bsed on projection onto convex set method*, proceedings of the 17th International Conference on Pattern Recognition, 3, 782-785, 2004.

16. E. Hartuv and R. Shamir. *A clustering algorithm based on graph connectivity*, Information Processing Letters, 76 (4-6), 175-181, 2000.

17. E. Hartuv, A. O. Schmitt, J. Lange, S. Meier-Ewert, H. Lehrach and R. Shamir. *An algorithm for clustering cDNAs for gene expression analysis*, Genomics, 66 (3), 249-256, 2000.

18. R. Herwig, A. J. Poustka, C. Miller, C. Bull, H. Lehrach, and J. O'Brien. *Large-scale clustering of cDNA fingerprint data*,

19. H. Kim, G. H. Golub and H. Park. *Missing Value Estimation for DNA microarray gene expression data: local least squares imputation*, Bioinformatics, 21 (2), 187-198, 2005.

20. T. Kohonen. *Self-Organizing Maps*, Third, extended edition. Springer, 2001.

21. J. MacQueen. *Some methods for classification and analysis of multivariate observations*, in proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability, L. M. Le Cam & J. Neyman (eds), 1, 281-297, University of California Press, 1967.

22. S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara and S. Ishii. *A Bayesian missing value estimation method for gene expression profile data*, Bioinformatics, 19 (16), 2088-2096, 2003.

23. C. H. Papadimitriou. *Computational Complexity*, Addison-Wesley; reading, MA, 1994.

24. C. Romesburg. *Cluster Analysis for Researchers*, Lulu Press, 2004.

25. R. Sharan and R. Shamir. *CLICK: a clustering algorithm with applications to gene expression analysis*, in proceedings of the International Conference on Intelligent Systems in Molecular Biology, 8, 307-316, 2000.

26. R. Shamir and R. Sharan. *Algorithmic Approaches to clustering Gene Expression Data*, in Current Topics in Computational Biology, 269-300, MIT Press, 2001.

27. L. Trevisan. *Positive linear programming, parallel approximation and pcps*, in proceedings of the 4th Annual European Symposium on Algorithms, 62-75, 1996.

28. O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein and R. B. Alman. *Missing value estimation methods for DNA microarray*, Bioinformatics, 17 (6), 520-525, 2001.

29. R. Uehara. *NP-complete problems on a 3-connected cubic planar graph and their applications*, Technical Report TWCU-M-0004, Tokyo Woman's Christian University, 1996.