Approximate Clustering of Fingerprint Vectors with Missing Values

Andres Figueroa[†] **Avraham Goldstein**[‡] **Tao Jiang**[†] **Maciej Kurowski**[◇]
Andrzej Lingas[⋆] **Mia Persson**[*]

[†]Computer Science Department, University of California Riverside, Riverside, CA 92521.
Email: {andres,jiang}@cs.ucr.edu

[‡]Department of Mathematics, Yeshiva University, New York, NY 10033.
Email: avi_goldstein@netzero.com

[◇]Institute of Informatics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland.
Email: kuros@mimuw.edu.pl

[⋆]Department of Computer Science, Lund University, 22100 Lund, Sweden.
Email: andrzej@cs.lth.se

[*]School of Technology and Society, Malmö University College, 20506 Malmö, Sweden.
Email: mia@cs.lth.se

Abstract

We study the problem of *clustering fingerprints with at most p missing values* (CMV(p) for short) naturally arising in *oligonucleotide fingerprinting*, which is an efficient method for characterizing DNA clone libraries.

We show that already CMV(2) is NP-hard. We also show that a greedy algorithm yields a $\min(1 + \ln n, 2+p\ln l)$ approximation for CMV(p), and can be implemented to run in $O(nl2^p)$ time. Furthermore, we introduce other variants of the problem of clustering fingerprints with at most p missing values based on slightly different optimization criteria and show that they can be approximated in polynomial time with ratios 2^{2p-1} and $2(1 - \frac{1}{2^{2p}})$, respectively.

Keywords: Approximation algorithms, oligonucleotide fingerprinting, clustering, NP-hardness

1 Introduction

In this paper, we study the problem of clustering binarized fingerprints with at most p missing values (CMV(p) for short) which arises very naturally in the problem of characterizing DNA clone libraries, especially in the so called *oligonucleotide fingerprinting* method (Drmanac, Stavropoulos, Labat, Vonau, Hauser, Soares & Drmanac 1996, Herwig, Poustka, Müller, Bull, Lehrach & O'Brien 1999, Meier-Ewert, Lange, Gerts, Herwig, Schmitt, Freund, Elge, Mott, Herrmann & Lehrach 1998, Valinsky, Della Vedova, Jiang & Borneman 2002, Valinsky, Della Vedova, Scupham, Alvey, Figueroa, Yin, Hartin, Chrobak, Crowley, Jiang & Borneman 2002). CMV(p) is a combinatorial optimization problem where one tries to identify clusters and resolve the missing values in the fingerprints simultaneously. The objective is to minimize the cardinality of the partition and the motivation behind is the minimum description length (MDL) principle (or Occam's razor) which makes it natural to consider the problem of partitioning the fingerprints into the smallest number of clusters, each consisting of similar fingerprint vectors. Furthermore, this approach is also consistent with the hypothesis that biomolecular diversity is a precious resource (Figueroa, Borneman & Jiang 2004).

The CMV(p) problem was first considered and motivated in (Figueroa, Borneman & Jiang 2004) where it was shown to be NP-hard for $p \geq 3$ and polynomially solvable for $p = 1$. However, the case $p = 2$ was stated as open in (Figueroa, Borneman & Jiang 2004). In (Figueroa, Borneman & Jiang 2004), also polynomial-time heuristics for CMV(p) were presented. One of them achieves the approximation ratio of 2^p. The other greedy one, iterating building the largest possible cluster, seems more practical; it runs in time $O(p2^pn^2)$, where n is the number of binarized fingerprints (Figueroa, Borneman & Jiang 2004).

In this paper, we show that CMV(2) is NP-hard by a reduction from the *minimum vertex cover problem* on planar, cubic, 3-connected and triangle-free graphs. Furthermore, we show that the aforementioned greedy heuristic yields a $\min(1 + \log n, 2 + p\log l)$ approximation for CMV(p), and can be implemented in $O(nl2^p)$ time, where l denotes the length of a fingerprint vector. We also introduce two other variants of the problem of clustering fingerprint vectors with at most p missing values based on slightly different optimization criteria. The first variant, termed as the problem of *inside compatible clustering with at most p missing values* (IECMV(p) for short) is defined analogously to CMV(p) with the exception that now the number of compatible (i.e., identical on positions not containing N) pairs of vectors within the same clusters is maximized instead of the minimization of the cardinality of the partition. The second variant, denoted as the problem of *outside compatible clustering with at most p missing values* (OECMV(p) for short) is again defined analogously to CMV(p) with the exception that now the number of compatible pairs of vectors belonging to different clusters is minimized. In this paper we show that, for any $p = O(\log n)$, IECMV(p) can be approximated in polynomial time with ratio 2^{2p-1} whereas in case no two compatible vectors have N at the same position OECMV(p) can be approximated in polynomial time with ratio $2(1 - \frac{1}{2^{2p}})$.

Our paper is structured as follows. In section 2, we provide more formal definitions of the problems of clustering binary fingerprints vectors that take into account missing values. In section 3, the NP-

Copyright ©2005, Australian Computer Society, Inc. This paper appeared at the Eleventh Computing: The Australasian Theory Symposium (CATS2005), Newcastle, Australia. Conferences in Research and Practice in Information Technology, Vol. 41. Mike Atkinson and Frank Dehne, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

hardness of CMV(2) is proved. The computational complexity of CMV(2) was stated as an open problem in (Figueroa, Borneman & Jiang 2004). In section 4, we consider the greedy algorithm for CMV(p) and prove that it yields an approximation ratio of $\min(1+\log n, 2+p\log l)$. We also show how to implement the greedy algorithm for CMV(p) in order to reduce its running time to $O(nl2^p)$. In Section 5, we prove that for any $p = O(\log n)$, IECMV(p) can be approximated in polynomial time with ratio 2^{2p-1} whereas the aforementioned restriction of OECMV(p) can be approximated in polynomial time with ratio $2(1-\frac{1}{2^{2p}})$.

2 Definitions

We consider binarized fingerprints obtained from hybridization intensity data, which are vectors of 1 (hybridization), 0 (no hybridization) or N (unknown classifications). Let n be the number of fingerprint vectors and let l denote the length of a fingerprint vector. Two fingerprints vectors f_i and f_j are *compatible* if for any position they differ f_i or f_j has N at this position. A $0-1$ vector r is a *resolved vector* of a $0-1-N$ fingerprint vector f if it is identical with f on all positions having 0 or 1 in f. Formally, we define the three different approaches to the problem of clustering fingerprints with at most p missing values as follows.

Definition 1 The problem of *clustering with p missing values* (*CMV(p)* for short) is to partition a set F of $0-1-N$ fingerprint vectors of length l with at most p symbols N into disjoint subsets $F_1, ..., F_k$ such that for each $1 \leq i \leq k$, any two vectors in F_i are compatible and the cardinality of the partition is minimized.

The problem of *inside compatible clustering with p missing values* (*IECMV(p)* for short) is defined analogously 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 problem of *outside compatible clustering with p missing values* (*OECMV(p)* for short) is again defined analogously with the exception that now the number of compatible pairs of vectors belonging to different clusters is minimized.

Note that an exact solution to IECMV(p) is an exact solution to OECMV(p) and *vice versa*.

3 The NP-hardness of CMV(2)

The CMV(p) problem has been proved to be NP-hard for $p \geq 3$ and solvable in polynomial time for $p = 1$ in (Figueroa, Borneman & Jiang 2004). The case $p = 2$ has been stated as an open problem in (Figueroa, Borneman & Jiang 2004). In the following, we prove that even for $p = 2$, CMV(p) is NP-hard.

To prove the NP-hardness of CMV(2), we show a reduction from the *minimum vertex cover problem* (MVC for short) on planar, cubic, 3-connected and triangle-free graphs, which is known to be NP-hard from (Uehara 1996), to the CMV(2) problem. The MVC problem is defined as follows. Given a graph $H = (V, E)$, find a subset $V' \subseteq V$ such that, for each edge$(u, v) \in E$, at least one of u and v belongs to V' and the cardinality of V' is minimized. To show the reduction, consider a planar, cubic, 3-connected and triangle-free graph $G = (V, E)$. Let $e \in E$ be an arbitrary edge incident with faces a and b. Denote the faces incident with the ends of e as a, b, x and a, b, y respectively (See Figure 1). Faces a, b (x, y) are called *close (far) neighbors of* e.

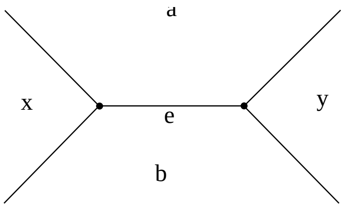

Figure 1: Far and close neighbors of e.

Let F denote the set of faces in G. We construct a set of $0-1-N$ fingerprint vectors as follows. For each edge e in G we define a $0-1-N$ fingerprint vector f_e of length $|F|$ as follows: set the positions a and b to 1, the positions x and y to N, and the remaining ones to 0. To show the NP-hardness of CMV(2), we first need to prove the following lemma.

Lemma 1 *Given a planar, cubic,* 3*-connected, triangle-free graph. Edges* e *and* e' *share a common vertex if and only if the vectors* f_e *and* $f_{e'}$ *are compatible.*

Proof: First, suppose that the edges e and e' share a common vertex v. Let the faces incident with v be denoted as p, q, r. The resolved vector $f_{p,q,r}$ has 1 at positions p, q, r and 0 on the remaining positions. Clearly, $f_{p,q,r}$ is compatible with both f_e and $f_{e'}$ and this proves the first part of the lemma. Second, suppose that for some non-incident edges e and e' the vectors f_e and $f_{e'}$ are compatible. Let close and far neighbors of e be a, b and x, y respectively. Similarly, let close and far neighbors of e' be a', b' and x', y'. Observe that $\{a, b\} \subset \{a', b', x', y'\}$ and $\{a', b'\} \subset \{a, b, x, y\}$. Essentially, we need to consider three different cases. First, note that one of these cases, namely when $a = a'$, $b = x'$, $b' = x$, implies that e and e' share a common vertex.

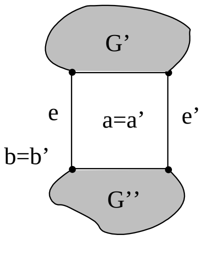

Figure 2: Case $a = a'$, $b = b'$.

Case $a = a'$, $b = b'$ (See Figure 2). Note that one can separate G' from G'' by deleting 2 vertices - a contradiction.

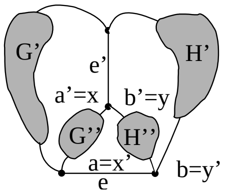

Figure 3: Case $a = x'$, $b = y'$, $a' = x$, $b' = y$.

Case $a = x'$, $b = y'$, $a' = x$, $b' = y$ (See Figure 3). Note that one of G' and G'' can not be empty because otherwise there would be a triangle in G. Let us assume that G' is not empty. It can be separated from the rest of the graph by removal of 2 vertices - a contradiction.

Note that all the other possible cases that may occur are symmetric to either case 1, case 2 or case 3 and therefore they are omitted here.

□

It remains to show that the vectors can be divided into k clusters if and only if G has a vertex cover of cardinality k. First, suppose that the vectors have been divided into clusters $F_1, \ldots, F_k$. As every pair of vectors from cluster F_i share a common vertex and there are no triangles in G all the vectors from F_i share a common vertex. Let us denote it by u_i. Clearly $u_1, \ldots, u_k$ constitute a vertex cover for G. Second, suppose that we have a vertex cover $u_1, \ldots, u_k$ for G. Clearly, we can divide the set of vectors into k clusters selecting to the i-th cluster the vectors corresponding to the edges incident with u_i. We have proved the following theorem.

Theorem 1 *The problem CMV(2) is NP-hard.*

4 Approximation of CMV(p)

In this section we consider the greedy heuristics for CMV(p) and prove that a greedy strategy yields an approximation ratio of $\min(1 + \ln n, 2 + p \ln l)$. We also give some implementation details about how to carefully implement the greedy algorithm for CMV(p) in order to achieve a running time of $O(nl2^p)$. Theorem 2 summarizes these results.

Theorem 2 *CMV(p) can be approximated in time $O(nl2^p)$ with ratio $\min(1 + \ln n, 2 + p \ln l)$. For $p = O(\log n)$ the approximation algorithm runs in polynomial time.*

Proof: Apply the greedy heuristic which iterates the following step while the set of remaining vectors is non-empty: add the largest possible cluster (i.e., the largest possible clique in the graph induced by the compatibility relation) to the current clustering and remove the vectors belonging to the cluster from the set of remaining vectors.

Since the operation of the greedy heuristic can be interpreted as covering the input set of vectors with subsets in one-to-one correspondence with the possible maximal clusters, it yields the $\min(1 + \ln n, 2 + p \ln l)$ approximation ratio (Johnson 1974). This is because the size of each clique is at most $\min(n, \binom{l}{0} + \binom{l}{1} + \ldots + \binom{l}{p})$.

Let us focus now on some details of implementation. For a fingerprint x (set of fingerprints X), let $res(x)$ ($res(X)$) be the set of all resolved fingerprints compatible with x (with some element of X).

First we compute an auxiliary bipartite graph H with set of vertices (A, B) where $A = res(F)$ and $B = F$. For $x \in A$ and $y \in B$, the edge xy is present in H iff x and y are compatible. The algorithm is summarized below.

```
Algorithm    Construction of $H = (A, B, E)$
1        $A := \emptyset$
2        $B := F$
3        $E := \emptyset$
4        for all $x \in B$ do
4.1          for all $y \in res(x)$ do
4.1.1            if $y \notin A$ then
4.1.1.1              Insert($y$,$A$)
                 endif
4.1.2            Insert($E$,$xy$)
             endfor
         endfor
End  Construction of $H = (A, B, E)$
```

There is a technical obstacle we have to take care of. When we go through the vertices from B, for each of them computing its resolved neighbors in A, we have to ensure that no duplicated fingerprints appear in A (see line 4.1.1 in Algorithm Construction of $H = (A, B, E)$). One possible solution to perform the check in $O(l)$ time is to use a hash table to store the elements of A. However if we want to avoid randomization we can use instead a large matrix of size 2^l. To avoid costly initialization we apply a standard backpointer technique. In both cases the construction of H takes $O(nl2^p)$ time.

Observe that each maximal clique corresponds to some resolved fingerprint (possibly not unique). Subsequently the most numerous clique corresponds to the resolved fingerprint $x \in A$ with the largest degree in H.

```
Algorithm    Greedy Clustering
for $i := 1$ to $n$ do
    $Q_i := \emptyset$
endfor
for all $x \in A$ do
    Insert($x$,$Q_{deg(x)}$)
endfor
for $i := n$ to $1$ do
    while $Q_i$ is not empty do
        $x$ :=Delete($Q_i$)
        Begin reporting a new cluster
        for all $y$ neighbor of $x$ do
            Report($y$)
            Delete($y$)
        endfor
        Delete($x$)
    endwhile
endfor
End  Greedy Clustering
```

To look for the vertices of the largest degree in A fast, we store them in an array of ranked queues. For each $i = 1 \ldots n$, the queue Q_i stores all the vertices from A with degree exactly i. At each step of the algorithm we take an arbitrary vertex x from the last nonempty queue, delete all the edges incident with x and their ends together with incident edges. In the algorithm above we assume that the operation Delete(x) removes from H vertex x, all the edges incident with x and moves the neighbors of x to the relevant queues. As every edge and every vertex is deleted from H only once it is clear that the complexity of this phase is linear in the size of H, i.e., $O(n2^p)$. □

5 Approximation of IECMV(p)

In this section, we shall reduce IECMV(p) and a restricted version of OECMV(p) to special variants of maximum and minimum satisfiability problems which will yield polynomial-time constant-factor approximations for both problems.

First we shall show that IECMV(p) can be expressed as a variant of maximum satisfiability problem where the formula is in disjunctive normal form. For each vector $1 \leq i \leq n$ and each position $1 \leq j \leq p$ of N in the vector, we reserve the variable $x_{i,j}$. For the sake of explanation, suppose first that $p = 1$. Then two vectors i and l are in the same cluster iff $x_{i,1}$ and $x_{l,1}$ are set to 0 or 1 so the two vectors become identical. By interpreting the variables as Boolean ones and 0 and 1 as false and true respectively, we can express this as the problem of satisfying a conjunction or disjunction of conjunctions of two literals based on $x_{i,1}$ and $x_{l,1}$. For example, if on the position corresponding to $x_{i,1}$ in the vector l there is 0 and on the position corresponding to $x_{l,1}$ in the vector i there is 1, we obtain the conjunction $\neg x_{i,1}x_{l,1}$. The case when $x_{i,1}$ and $x_{l,1}$ are on corresponding positions is a bit different. Then either both variables have to be set to 0 or both have to be set to 1 which is expressible as a disjunction of $\neg x_{i,1}\neg x_{l,1}$ with $x_{i,1}x_{l,1}$. For larger p, we obtain analogously a disjunction of conjunctions. Each of the conjunctions contains all the variables corresponding to the two strings and thus it has length at most $2p$. The number of the conjunctions is easily seen to be at most 2^p. By forming the disjunction of the conjunctions, we obtain a DNF formula F for which the maximum number of simultaneously satisfiable conjunctions equals the maximum number of compatible pairs of vectors inside the same clusters. An approximative solution with ratio r to the maximum satisfiability problem yields an approximative solution with ratio r to IECMV(p) and *vice versa*. By (Trevisan 1996), the maximum satisfiability problem for DNF formulas with conjunctive clauses of length at most k admits a polynomial-time approximation algorithm with ratio 2^{k-1}. Hence, we obtain an approximation ratio of 2^{2p-1} for the IECMV(p) problem.

By taking the negation of F, and applying de Morgan laws, we obtain a CNF formula G with clauses of length at most $2p$. Now, the problem of finding the minimum number of clauses in G that can be simultaneously satisfied is easily seen to be equivalent to OECMV(p). Furthermore, if no two compatible vectors contain N at the same position, there is one-to-one correspondence between satisfied clauses and compatible pairs outside clusters. Since the problem of minimum k-satisfiability admits a polynomial-time approximation algorithm with ratio $2(1-\frac{1}{2^k})$ (Bertsimas, Teo & Vohra 1996), we obtain an approximation ratio of $2(1-\frac{1}{2^{2p}})$ for the so restricted OECMV(p) problem. Thus, we have proved the following theorem.

Theorem 3 *For any $p = O(\log n)$, IECMV(p) can be approximated in polynomial time with ratio 2^{2p-1} whereas in case no two compatible vectors have N at the same position OECMV(p) can be approximated in polynomial time with ratio $2(1-\frac{1}{2^{2p}})$.*

6 Concluding remarks and open problems

We designed polynomial-time approximation algorithms for three different variants, CMV(p), IECMV(p) and restricted OECMV(p), of the problem of clustering fingerprints with p missing values. In particular, we presented a greedy approximation algorithm for CMV(p), running in polynomial time for $p = O(\log n)$. We also proved the NP-hardness of CMV(2) and by this, the complexity status of CMV(p) for all possible values of the parameter p is resolved. Several open problems remain. Several of them have been raised during joint discussions with Leszek Gasieniec and Peter Damaschke.

1. Does OECMV(p) in the general case admit an $O(1)$-approximation polynomial-time algorithm?

2. Does the greedy heuristic yield also a non-trivial approximation ratio for IECMV(p)? Some experimental work would be helpful in order to develop intuitions about this question.

3. Is there any non-trivial approximation relationship between CMV(p), IECMV(p) and OECMV(p). Again, some experimental work would be useful here.

4. For a set of $0-1-N$ fingerprint vectors, one can consider several problems related to the construction of phylogenetic trees. For instance, find an assignment to the N-positions which, for the resulting vectors, yields perfect phylogeny or a phylogenetic tree of minimum size or a phylogenetic tree with minimum number of mutations *etc.*

References

Bertsimas, D., Teo, C-P. & Vohra, R. (1996), On dependent randomized rounding algorithms, *in* 'Proc. 5th International Conference on Integer Programming and Combinatorial Optimization', pp. 330–344.

Drmanac, S., Stavropoulos, N.A., Labat, I., Vonau, J., Hauser, B., Soares, M.B. & Drmanac, R. (1996), 'Gene representing cDNA clusters defined by hybridization of 57,419 clones from infant brain libraries with short oligonucleotide probes', *Genomics* **37**, 29–40.

Figueroa, A., Borneman, J. & Jiang, T. (2004), Clustering binary fingerprint vectors with missing values, to appear in Journal of Computational Biology.

Herwig, R., Poustka, A.J., Müller, C., Bull, C., Lehrach, H. & O'Brien, J. (1999), 'Large-scale clustering of cDNA-fingerprinting data', *Genome research* **9**, 1093–1105.

Johnson, D.S. (1974), 'Approximation algorithms for combinatorial problems', *Journal of Computer and System Sciences* **9**, 256–278.

Meier-Ewert, S., Lange, J., Gerts, H., Herwig, R., Schmitt, A., Freund, J., Elge, T., Mott, R., Hermann, B. & Lehrach, H. (1998), 'Comparative gene expression profiling by oligonucleotide fingerprinting.', *Nucleic Acids Research* **26**, 2216–2223.

Trevisan, L. (1996), Positive linear programming, parallel approximation and pcps, *in* 'Proc. 4th Annual European Symposium on Algorithms', pp. 62–75.

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

Valinsky, L., Della Vedova, G., Jiang, T. & Borneman, J. (2002), 'Oligonucleotide fingerprinting of ribosomal RNA genes for analysis of fungal community composition', *Applied and Environmental Microbiology* **68**, 5999–6004.

Valinsky, L., Della Vedova, G., Scupham, A., Alvey, S., Figueroa, A., Yin, B., Hartin, R., Chrobak, M., Crowley, D., Jiang, T. & Borneman, J. (2002), 'Analysis of bacterial community composition by oligonucleotide fingerprinting of rRNA genes', *Applied and Environmental Microbiology* **68**, 3243–3250.