JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 12, Number 5, 2005

Pp. 514–533

Multiplexing Schemes for Generic SNP Genotyping Assays[1]

RODED SHARAN,[2,3] JENS GRAMM,[2,4] ZOHAR YAKHINI,[5] and AMIR BEN-DOR[5]

ABSTRACT

Association studies in populations relate genomic variation among individuals with medical condition. Key to these studies is the development of efficient and affordable genotyping techniques. Generic genotyping assays are independent of the target SNPs and offer great flexibility in the genotyping process. Efficient use of such assays calls for identifying sets of SNPs that can be interrogated in parallel under constraints imposed by the genotyping technology. In this paper, we study problems arising in the design of genotyping experiments using generic assays. Our problem formulation deals with two main factors that affect the genotyping cost: the number of assays used and the number of PCR reactions required for sample preparation. We prove that the resulting computational problems are hard, but provide approximate and heuristic solutions to these problems. Our algorithmic approach is based on recasting the multiplexing problems as partitioning and packing problems on a bipartite graph. We tested our algorithmic approaches on an extensive collection of synthetic data and on data that was simulated using real SNP sequences. Our results show that the algorithms achieve near-optimal designs in many cases and demonstrate the applicability of generic assays to SNP genotyping.

Key words: genotyping, multiplexing, experimental design, synchronized matching, graph partitioning, graph packing, approximation algorithms.

1. INTRODUCTION

SINGLE NUCLEOTIDE POLYMORPHISMS (SNPs) are differences in a single base, across the population, within an otherwise conserved genomic sequence. SNPs account for most of the genomic variation in human (Patil *et al.*, 2001) and are often associated with medical condition or specific drug response. Thus, efficient methods for determining SNP variants across a population are of great clinical, scientific, and commercial value.

SNP *genotyping* is the process of determining, for a given set of SNPs and a given individual, the SNP variants, or *alleles*, which are present in the genomic sequence of that individual. Most current SNP

[1]Portions of this paper appeared in Sharan *et al.* (2004).

[2]These authors contributed equally to this work.
[3]School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel.
[4]Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Germany.
[5]Agilent Laboratories, 3500 Deer Creek Road, Palo Alto, CA 94304.

genotyping techniques (Syvanen, 1999; Ross *et al.*, 1998) are problem specific in the sense that at least some of the reagents used in the assay have to be specifically tailored to the set of SNPs under interrogation. *Generic* methods are techniques that defer all problem-specific components to the assay-planning stage and to the result-interpretation stage. For example, Sampson *et al.* (2001) present a method that uses natural and mass-modified generic mixtures of oligonucleotides. A target-mediated enzymatic reaction produces a mixture, whose mass-spectrum is indicative of the sample genotype.

Multiple SNP sites can be genotyped simultaneously in a single assay under certain conditions, a process called *multiplexed genotyping*. Examples include utilizing primer extension and MALDI-TOF mass spectrometry, relying on the natural masses of the extended specifically designed primers (Ross *et al.*, 1998; Aumann *et al.*, 2003). Typically, not all SNPs in a set of interest can be genotyped together; any given genotyping method imposes a set of constraints regarding which SNPs can be assayed together and which cannot. In order to achieve high multiplexing rates and, thus, reduce the genotyping cost, it is necessary to carefully plan the genotyping assays. The design should allow simultaneous genotyping of as many SNPs as possible, on the one hand, while conforming to the constraints, on the other.

Here, we present methods for achieving high multiplexing rates for a family of generic SNP genotyping techniques. We model all the applications in a unified framework in which each allele is assigned a set of features and the multiplexing problem translates to that of finding a minimum partition of the allele set into subsets such that every allele has a feature which is unique with respect to its subset; each of these sets can then be typed using a single assay.

We also study an important variant of this multiplexing problem in which the designed experiments have to meet an additional requirement: The two alleles of every SNP have to be assigned to the same assay. Imposing this requirement is important in practice, since it allows considerable savings in the number of PCR reactions needed for sample preparation, reducing the overall genotyping cost.

We provide both theoretical and practical results for the resulting multiplexing problems. On the theoretical side, we study the partitioning problems and related variants in which one has to find a maximum set of alleles (or SNPs) that can be typed using one assay. We prove that all these problems are NP-hard and devise constant-factor approximation algorithms for them, exploiting the fact that the sequences associated with each SNP have bounded length. Our algorithmic approaches are based on representing the input data using a bipartite graph and solving the resulting partitioning and packing problems using maximum flow techniques. In particular, we give a polynomial time approximation scheme for a problem that is reminiscent of that of packing paths of length 2. In addition, we develop practical heuristic approaches for the multiplexing problem and devise lower bounds for evaluating the performance of the different algorithms.

On the practical side, we conduct extensive benchmarking of the algorithmic approaches on synthetic data and on data that was simulated using real SNP sequences from various sources. Our tests show that one of the heuristic approaches outperforms the other approaches and achieves experimental designs that use an optimal or near-optimal number of assays in many cases. More importantly, when comparing the two variants of the multiplexing problem—with and without the additional requirement on co-assigning the two alleles of an SNP, we discover that imposing this requirement has little effect on the number of assays in the solution, but has great impact on the saving in PCR reactions needed. We also show that for datasets that consist of several hundreds known SNPs in the region of disease-related genes, few assays (up to seven) are sufficient for the genotyping task, demonstrating the applicability of generic assays for SNP genotyping.

The paper is organized as follows: In Section 2, we describe the genotyping process and present the main multiplexing problems that are studied in the paper. In Section 3, we present the main hardness results. In Sections 4 and 5, we provide algorithmic approaches to the two variants of the multiplexing problem. The heuristic approaches are described in Section 6. Section 7 contains our results on synthetic and simulated data.

2. GENOTYPING METHODS AND PROBLEM STATEMENT

2.1. Generic genotyping techniques

Polymerase extension is a widely used technique for interrogating DNA sequences. Typically, all methods based on this technique utilize extension of specifically designed primers and are not generic. For example,

in an array polymerase extension assay (APEX) (Kurg *et al.*, 2000; Syvanen, 1999) the target sample is annealed to array bound probes that are complementary to subsequences upstream from the polymorphic sites. Four fluorescently labeled dideoxynucleotides are used in a primer extension reaction for extending the array probes. The fluorescence signal of the extended probes allows the determination of SNP variants present in the sample. Note that the array needs to be specifically designed to address the input set of SNPs.

In a generic polymerase extension assay, the target sample reacts with a generic set of primers, e.g., all k-mers (DNA sequences of length k). These primers are extended, or not, depending on the target. A detection step follows, wherein the extended primers are determined, based on their altered properties. Information on the target is obtained by an interpretation process. We provide two concrete examples of such techniques below. For convenience, here and in the description of the algorithmic approaches in later sections, we consider a primer sequence s' to hybridize to a target sequence s if s' occurs in s (as a subsequence). In practice, both the sequence s and its Watson–Crick complement are present in the sample, so s' will hybridize to s if it occurs in any of them. Two alleles that correspond to the same SNP are called *mates*.

All *k*-mer Arrays. First, assume that a single site is to be genotyped. The genotyping protocol is as follows:

1. The target region is PCR amplified.
2. The sequence is hybridized to the array and a polymerase reaction is started, in the presence of single labeled dideoxynucleotides.
3. k-mers that occur in nonpolymorphic parts of the amplicon will hybridize to the target, get extended, and produce fluorescence signals.
4. The hybridization signals obtained for k-mers that span the site depend on the alleles of this SNP in the genotyped individual.

The genotype of the sample at the interrogated site can be determined by analyzing the hybridization signature, provided that there is at least one k-mer for each allele that does not appear in the sequence of its mate.

In a multiplexed assay, several targets are jointly interrogated. The set can be jointly interrogated as long as each allele has at least one unique k-mer that does not occur in the sequence of any other allele in the set.

Native/tagged mass-spectrometry. This process involves the following components (Sampson *et al.*, 2001):

1. A mixture of primers is applied to the target in the presence of polymerase and all 4 dideoxynucleotides, allowing for single base extension to occur in a specific, target-mediated manner.
2. Extended primers are separated from the mixture, e.g., by high performance liquid chromatography, and are analyzed by mass spectrometry.

Under complete stringency assumptions, the output mass spectrum will have peaks only at masses that correspond to extended primers that occur in the target sequence. A set of SNPs can be jointly interrogated as long as each of the respective alleles has a corresponding extended primer with a unique mass, different from that potentially arising from any other allele in the set.

A similar genotyping process uses cleavable mass-tags that are attached to the original primers and then cleaved after the separation of the extended products. (Here we assume that the number of available distinguishable tags exceeds the number of primers.) The tags, rather than the extended primers, are analyzed by mass spectrometry. The spectrum will have peaks at masses of tags that correspond to primers that occur in the target sequence. Again, a set of SNPs can be jointly interrogated as long as each of the respective alleles has a corresponding extended primer with a unique tag, different from that potentially arising from any other allele in the set.

2.2. Problem formulation

In any of the embodiments, the target is typically a collection of short PCR amplicons, spanning bi-allelic SNP sites. A SNP allele in a target can be determined if and only if the extension event, for one of the k-mers spanning this site and corresponding to this allele, can be uniquely detected under the assay conditions. This requirement can be abstracted as follows: Associate with each target sequence a list of features at which it registers, e.g., all its constituent k-mers, the masses corresponding to all extended primers, etc. This is the set of features potentially *activated* by the given target sequence. Furthermore, the set of activated features can be partitioned into *informative* ones, being all features activated by the amplicons corresponding to *only one* allele of this SNP, and *common* ones, being all features activated by the amplicons corresponding to both alleles of this SNP. Obviously, all informative features span the polymorphic site, while not every feature spanning the polymorphic site is necessarily informative. A set of alleles is *assignable* if each allele in the set has an informative feature that is not potentially activated by any other allele in the set. Efficient genotyping calls for partitioning the given set of SNPs into assignable subsets. This partition constitutes a *multiplexing scheme*.

The objective of the multiplexing scheme can be modeled in two ways. Both formulations reflect the fact that when a specific site is genotyped, both its alleles may activate features (indeed, this will be the case if the sample is heterozygous) and there is no easy way to separate these sets of features one from the other. In the first formulation, we seek a partition of the SNPs into a minimum number of assignable subsets. The basic units here are *allele-pairs* (corresponding to SNPs). In the second variant we seek a partition of the alleles into a minimum number of assignable subsets. The basic units here are *single alleles*, dropping the constraint that two alleles corresponding to one SNP should be put in the same subset in a partition. Solutions to the first variant have the advantage that they require a smaller number of PCR reactions compared to the second variant, since every pair of mates that is split among two assays, requires two amplification reactions instead of one. However, when studying the multiplexing problem in isolation, the single-allele version is the more general one.

We now formulate the arising multiplexing problems. We represent an allele-pair as a pair (s, s') of l-long strings over $\{A,C,G,T\}$, such that s and s' differ in a single position, which is referred to as the *SNP site* of this pair. For a set A of allele-pairs, we call $S_A \equiv \bigcup_{(s,s')\in A}\{s, s'\}$ the *allele set* of A. Given an allele $s \in S_A$, we call a k-long string t a *feature* of s if s contains t as a substring. Given an allele-pair (s, s'), we call a feature t *informative* if only one of s and s' contains t. Thus, all occurrences of t span the SNP site. Otherwise, if both s and s' contain t, t is called *noninformative*. We call an informative feature t of s *unique* with respect to A, if no string $s' \in S_A$, $s' \neq s$, contains t as a substring. A set S of alleles is assignable if every $s \in S$ has a unique (informative) feature. A set A of allele-pairs is assignable if S_A is assignable. The main problems that we study are the following:

- MINIMUM ASSIGNABLE COVER (MAC): Given a set A of allele-pairs and feature length k, find a partition of S_A into a minimum number of assignable subsets.
- MINIMUM ALLELE-PAIR COVER (MAPC): Given a set A of allele-pairs and feature length k, find a partition of A into a minimum number of assignable subsets.

We also study the related problems that call for identifying a maximum subset of alleles or allele-pairs that can be genotyped using one assay:

- MAXIMUM ASSIGNABLE SET (MAS): Given a set A of allele-pairs and feature length k, find an assignable subset of S_A of maximum size.
- MAXIMUM ASSIGNABLE ALLELE-PAIR SET (MAPS): Given a set A of allele-pairs and feature length k, find an assignable subset of A of maximum size.

3. COMPLEXITY ANALYSIS

In this section, we prove that the multiplexing problems MAC, MAPC, MAS, and MAPS are NP-complete and hard to approximate.

Theorem 1. MAPC *and* MAPS *are NP-complete when the feature length k is at least logarithmic in the number of allele-pairs.*

Proof. Clearly, both problems are in NP. Below we prove the hardness of MAPS by reduction from INDEPENDENT SET (Garey and Johnson, 1979). The hardness proof for MAPC employs an analogous reduction from COLORING. Let $(G = (V, E), r)$ be an instance of INDEPENDENT SET, where $V = \{v_1, \ldots, v_n\}$. We transform this instance into an instance (A, k, r) of MAPS, where A denotes the set of allele-pairs, k is the length of the features on the array, and r is the size of a requested assignable subset of A. We set $k \equiv 2\lceil \log n \rceil + 4$. We describe the construction using the alphabet $\{0, 1\}$, which implies hardness also for larger-size alphabets. We use "$\cdot$" to denote the concatenation of strings.

The allele-pair set A is constructed to contain, for every $v_i \in V$, an allele-pair (s_i^0, s_i^1). The building blocks of our construction are strings $\langle v_i \rangle_0$ and $\langle v_i \rangle_1$ that encode vertices of V. We denote by $\langle \log i \rangle_{bin}$ the $\lceil \log n \rceil$-long binary encoding of a positive integer i, $1 \le i \le n$. We denote by $\langle \# \rangle$ the $(\lceil \log n \rceil + 3)$-long string consisting of $(\lceil \log n \rceil + 1)$-many 1's flanked by two 0's. We define

$$\langle v_i \rangle_0 \equiv 0 \cdot \langle \# \rangle \cdot \langle \log i \rangle_{bin},$$
$$\langle v_i \rangle_1 \equiv 1 \cdot \langle \# \rangle \cdot \langle \log i \rangle_{bin}.$$

Given $v_i \in V$, let $N(v_i) = \{v_{j_1}, \ldots, v_{j_i}\}$ denote the set of neighbors of v_i in G. Define the strings

$$t_{neighbors(i),0} \equiv \langle v_{j_1} \rangle_0 \cdot 0 \cdot \langle v_{j_2} \rangle_0 \cdot 0 \cdots 0 \cdot \langle v_{j_i} \rangle_0,$$
$$t_{neighbors(i),1} \equiv \langle v_{j_1} \rangle_1 \cdot 0 \cdot \langle v_{j_2} \rangle_1 \cdot 0 \cdots 0 \cdot \langle v_{j_i} \rangle_1.$$

The allele-pair (s_i^0, s_i^1) is represented by the following two sequences:

$$s_i^0 \equiv \langle v_i \rangle_0 \cdot 0 \cdot t_{neighbors(i),0} \cdot 0 \cdot t_{neighbors(i),1},$$
$$s_i^1 \equiv \langle v_i \rangle_1 \cdot 0 \cdot t_{neighbors(i),0} \cdot 0 \cdot t_{neighbors(i),1}.$$

Consequently, s_i^0 and s_i^1 differ only in their first position. This position represents the SNP site of this allele-pair. Finally, setting $A \equiv \{ (s_i^0, s_i^1) \mid 1 \le i \le n \}$ completes our construction. In the following, we prove that G has an independent set of size at least r if and only if A has an assignable subset of size at least r.

($\Rightarrow$) Let $V' \subseteq V$ be an independent set of size r in G. Then $A' \equiv \{ (s_i^0, s_i^1) \mid v_i \in V' \}$ is an assignable set of size r, since for each i and for each allele $b \in \{0, 1\}$, $b \cdot \langle \# \rangle \cdot \langle \log i \rangle_{bin}$ is a unique feature of s_i^b.

($\Leftarrow$) Let $A' \subseteq A$ be an assignable set of size r. We claim that $V' = \{ v_i \in V \mid (s_i^0, s_i^1) \in A' \}$ is an independent set of size r in G. Since A' is assignable, each of its alleles s_i^b has a unique feature. This feature must span the SNP site or, else, it will be shared also by $s_i^{(1-b)}$. By construction, this feature must be the substring $b \cdot \langle \# \rangle \cdot \langle \log i \rangle_{bin}$. The uniqueness of this feature implies that for any two alleles s_i^b and $s_j^{b'}$, their corresponding vertices are not adjacent in G. To see this, suppose to the contrary that v_i and v_j are adjacent. Then $b \cdot \langle \# \rangle \cdot \langle \log i \rangle_{bin}$ is a substring of $t_{neighbors(j),b}$ and, consequently, it occurs in s_j^0 and in s_j^1. Thus, $\{s_i^b, s_j^{b'}\}$ is not an assignable set, a contradiction. ■

Corollary 1. MAC *and* MAS *are NP-complete.*

Proof. One can use the same reductions as in the proof of Theorem 1. For one direction, note that an assignable subset of allele-pairs implies that its constituent alleles also form an assignable subset. The other direction follows by noting that the construction in the proof of Theorem 1 ensures that if an allele is assigned to a subset S then its mate can also be assigned to S. ■

Since all the above reductions yield 1–1 correspondences between independent set or coloring solutions and solutions to the multiplexing problems, one can use the hardness-of-approximation results of Håstad (1999) to conclude the following:

Corollary 2. *It is NP-hard to approximate* MAC, MAPC, MAS, *and* MAPS *to within a factor of* $|A|^{1-\epsilon}$ *for any* $\epsilon > 0$, *unless NP* $=$ *ZPP.*

We note that the above proofs can be easily extended to the case that the SNP site occupies the middle position of its corresponding sequence.

4. APPROXIMATION ALGORITHMS FOR THE SINGLE-ALLELE CASE

In this section, we present an approximation algorithm for the multiplexing problem under the single-allele variant. First, we devise a graph-theoretic formulation of the problem.

4.1. Graph-theoretic modeling

We model the input as an edge-colored bipartite graph $G = (V, U, E)$ in which edges carry one of two colors. We translate an allele-pair set A as follows to an edge-colored bipartite graph G_A. Let V be the allele set S_A of A, and let U be the set of all possible features (here we focus on assays whose feature set is the set of all k-mers). Vertices in V are called *allele vertices*, and vertices in U are called *feature vertices*. We construct the set E of edges as follows: For an allele $v \in V$ and a feature $u \in U$, we add a *red*-colored edge $(v, u) \in E$ if u is an informative feature of v. We add a *black*-colored edge $(v, u) \in E$ if u is a noninformative feature of v. If u is not a feature of v then there is no edge between u and v. The edges colored red are called *informative edges* and are denoted by E_{inf}. The edges colored black are called *noninformative edges* and are denoted by E_{ni}. By definition, $E = E_{inf} \cup E_{ni}$. The resulting graph G_A is called an *alleles-features (AF) graph*.

Note that for every allele, k of its features (k-mers; not necessarily distinct) span the polymorphic site. Therefore, in G_A a vertex $v \in V$ has at most k informative edges incident to it; and at most $(l - k + 1)$ noninformative edges, corresponding to $(l - 2k + 1)$ k-mers that do not involve the polymorphic site and k additional k-mers that constitute the informative features of the allele's mate. Also note that not every edge-colored bipartite graph can be realized as an AF graph (i.e., has a corresponding set of alleles and features).

A *matching* in a graph is a subset of independent edges. A matching is said to *cover* a set of vertices if its edges are incident to all those vertices. An *induced matching* is a matching such that no two edges of the matching are connected by an edge.

In an AF graph, consider a subset $V' \subseteq V$ of vertices corresponding to alleles. The set V' is called *assignable* if there exists an induced matching M that covers V' such that $M \subseteq E_{inf}$ (i.e., M consists of informative edges only). An assignable subset of alleles corresponds to an assignable set of allele vertices in the corresponding AF graph, i.e., to an induced matching between the alleles and their unique features. Observe that for a given set V' of allele vertices, one can test in linear time whether V' is assignable, by checking if each allele in V' has an informative edge to a unique feature. However, as we have shown above, the problem of identifying a maximum assignable set is NP-hard. The related problem of finding a maximum induced matching was shown to be NP-hard for general bipartite graphs by Cameron (1989).

We approach MAS and MAC by studying relaxations of the corresponding problems on AF graphs. These relaxations ask for matchings that are not necessarily induced. For the AF graph, this means that we omit the noninformative edges E_{ni} and consider only the informative edges E_{inf}; the resulting graph is a general bipartite graph with no edge colors. The relaxed problem variants can be stated as follows:

- MINIMUM MATCHING COVER (MMC): Given a bipartite graph $G' = (V, U, E)$, find a minimum number of matchings that cover V.
- MAXIMUM BIPARTITE MATCHING: Given a bipartite graph $G' = (V, U, E)$, find a maximum matching in G'.

Given an instance $G = (V, U, E_{inf} \cup E_{ni})$ of MAC, the cardinality of an optimum solution to MMC on $G' = (V, U, E_{inf})$ is a lower bound on the cardinality of any solution to MAC on G and, in particular, a lower bound on the optimum MAC solution. We can compute this lower bound in polynomial time using the algorithm of Aumann *et al.* (2003) for MMC.

4.2. Approximating MAS and MAC

Our approximation algorithms rely on a general technique for partitioning a set of vertex-disjoint subgraphs into subsets such that the subgraphs in each subset are not connected by an edge. We start by describing this technique.

Theorem 2. *Let $G = (V, U, E)$ be a bipartite graph with the degree of every $v \in V$ bounded by d. Let $\mathcal{S}$ be a set of vertex-disjoint induced subgraphs of G such that every $G' \in \mathcal{S}$ contains at most one vertex from V, whose degree in G' is at least k. Then $\mathcal{S}$ can be partitioned into at most $2(d-k)+1$ subsets such that in every subset no two subgraphs are connected by an edge.*

Proof. We generalize the coloring approach of Ben-Dor *et al.* (2003): We build an auxiliary directed graph H, whose vertices correspond to the subgraphs in $\mathcal{S}$. For any two subgraphs $G_1, G_2 \in \mathcal{S}$ whose corresponding vertices in H are h_1 and h_2, we direct an edge in H from h_1 to h_2 if G_1 contains a vertex $v \in V$ that is adjacent to some vertex $u \in U$ in G_2. By assumption, the maximum outdegree in H is $d-k$. Therefore, the undirected graph which underlies H can be colored using smallest-last ordering (SLO) coloring (Matula and Beck, 1983) by at most $2(d-k)+1$ colors. Each color class represents an independent set of vertices, which correspond to a subset of $\mathcal{S}$ in which no two subgraphs are connected by an edge. ■

The following theorem states our approximation result for MAC.

Theorem 3. *There is a $(2l+1)$-approximation algorithm to* MAC, *where l is the length of the input allele sequences.*

Proof. We find the approximate solution in two stages. The reader is referred to the example in Fig. 1 for further explanation and intuition about the algorithm. First, we construct the graph $G' = (V, U, E_{inf})$ by removing the non-informative edges from G. We find a minimum matching cover $E_1, E_2, \ldots, E_r$ of G' using the algorithm of Aumann *et al.* (2003). In the second stage, we use Theorem 2 to partition the alleles covered by each matching E_i into at most $(2l+1)$ assignable sets. This is possible since the degree of each allele vertex is exactly 1 in the matching and at most $l+1$ in G. Overall, the cardinality of our solution is bounded from above by $opt_{MMC}(G') \cdot (2l+1) \leq opt_{MAC}(G) \cdot (2l+1)$. ■

A similar approximation algorithm that is based on solving the maximum bipartite matching problem can be devised for MAS, yielding the following result:

Theorem 4. *There is a $(2l+1)$-approximation algorithm to* MAS *on G, where l is the length of the input allele sequences.*

5. APPROXIMATION ALGORITHMS FOR THE ALLELE-PAIR CASE

In this section, we provide approximation algorithms to MAPS and MAPC. Similarly to the single allele case, we recast the multiplexing problems as partition and packing problems on a bipartite graph and derive the approximation algorithms by studying relaxations of the problems, in which one looks for sets of disjoint triples consisting of an allele-pair and two informative features for each of its alleles, rather than requiring the uniqueness of the features assigned to the alleles. The solutions to the relaxed problems are then further modified to satisfy the uniqueness constraints.

5.1. Graph-theoretic modeling

We model the input as an edge-colored bipartite graph $G = (V, U, E)$, similar to the AF graph in Section 4.1 while here the edges carry one of *three* colors. We translate an allele-pair set A to such an edge-colored bipartite graph G_A as follows. The set V consists of the allele-pairs in the input set A, and these vertices are consequently called *allele-pair vertices*. The set U of feature vertices consists of all

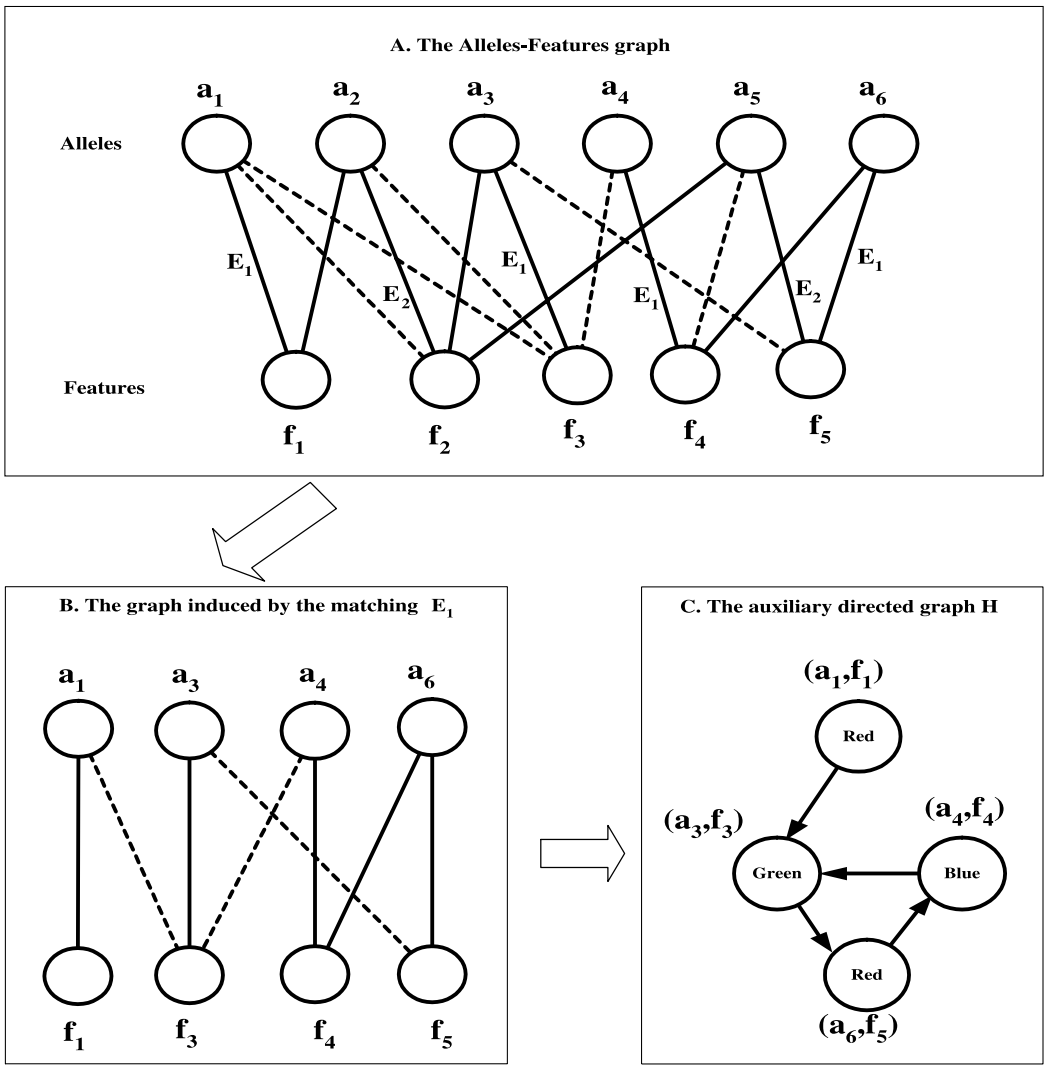

FIG. 1. An example demonstrating the approximation algorithm for MAC. (**A**) The alleles–features graph. Informative edges are solid; noninformative edges are dashed. E_1 and E_2 represent one possible optimal matching cover. (**B**) The graph induced by the matching E_1 in part A. (**C**) The auxiliary directed graph H constructed from the graph in part B. The outdegree in H is at most 1, so H can be colored with 3 colors. Each color class corresponds to a set of independent edges in E_1. For example, the red color corresponds to the edges $(a_1, f_1), (a_6, f_5)$. These edges corresponds to the assignable allele set $\{a_1, a_6\}$.

possible features (k-mers). The set E of edges connects allele-pairs to features that are present in their sequences. There are three kinds of features and, correspondingly, E is the union of edges of three types, or colors: For every allele-pair $v = (s, s')$, there is a *red* edge $(v, u) \in E$ if and only if $u \in U$ is an informative feature of s, and there is a *blue* edge $(v, u) \in E_b$ if and only if $u \in U$ is an informative feature of s'. The set E_r of red edges and the set E_b of blue edges comprise the *informative edges*. In addition, there is a *black* edge $(v, u) \in E$ if and only if u is a noninformative feature of s (and s'); these edges form the set E_{ni} and are called *noninformative*. A graph being generated from an allele-pair set in this way is called a *SNPs-features (SF)* graph.

Given an SF graph $G = (V, U, E)$, a triple of vertices (v, u, u') with $v \in V$ and $u, u' \in U$ is called a *synchronized triple* if $(v, u) \in E_r$ and $(v, u') \in E_b$. In words, a synchronized triple consists of an allele-pair and two informative features, one for each of its alleles. A set of synchronized triples forms a *synchronized matching* if all its triples are vertex-disjoint. We call a synchronized matching *induced* if no two of these triples are joined by an edge in $E = E_r \cup E_b \cup E_{ni}$. A subset $V' \subseteq V$ is called *assignable* if and only

if there exists an induced synchronized matching that covers the vertices of V'. Note that any assignable subset of the allele-pair set A corresponds to an assignable set of allele-pair vertices in the corresponding SF graph G_A.

Much as we did in Section 4.1, we approach MAPS and MAPC by studying relaxations of the corresponding problems on SF graphs. We relax the requirements for an assignable set by considering synchronized matchings that are not necessarily induced. For the SF graph, this means that we can ignore the noninformative edges E_{ni} and restrict attention to the informative edges in E_r and E_b. For defining the relaxations, it will be convenient for us to generalize the notion of synchronized triples to any bipartite graph (V', U', E') whose edges are colored with two colors. In this case, we define a synchronized triple to be a triple $(v, u, u') \in V' \times U' \times U'$ such that (v, u) and (v, u') are edges with different colors. The relaxations that we study are defined as follows:

- SYNCHRONIZED MATCHING COVER (SMC): Given a bipartite graph $G = (V, U, E)$ with edges of two colors, find a partition of V into a minimum number of subsets, each of which can be covered by a synchronized matching in G.
- MAXIMUM SYNCHRONIZED MATCHING (MSM): Given a bipartite graph $G = (V, U, E)$ with edges of two colors, find a maximum synchronized matching in G.

Clearly, the cardinality of an optimum solution to MAPC for an allele-pair set A is bounded from below by the cardinality of an optimum solution to SMC on the corresponding SF subgraph $G_A = (U, V, E_r \cup E_b)$. Similarly, the cardinality of an optimum solution to MAPS is bounded from above by the cardinality of an optimum solution to MSM on the same instance.

Theorem 5. SMC *is NP-complete, even on SF graphs.*

Proof. Clearly, the problem is in NP. The NP-hardness of SMC is shown by a reduction from EDGE COLORING (Holyer, 1981). Let $(G = (V, E), c)$ be an instance of EDGE COLORING, asking whether the edges of graph G can be colored by c colors such that no two adjacent edges are assigned the same color. First, we show that SMC is NP-hard for general bipartite graphs with edges of two colors, and then we prove hardness on SF graphs.

We construct an instance $G_A = (V_A, U_A, E_A)$ of SMC as follows: $V_A \equiv \{ v_e \mid e \in E \}$, $U_A \equiv \{ u_v \mid v \in V \}$, and E_A contains, for every edge $e = (w_1, w_2) \in E$, a red-colored edge (v_e, u_{w_1}) and a blue-colored edge (v_e, u_{w_2}). The correctness of the hardness proof is based on a 1–1 correspondence between edge colorings for G and synchronized matching covers for G_A: For all edges $e = (w_1, w_2) \in E$ that are assigned the same color in an edge coloring for G, the synchronized triples (v_e, u_{w_1}, u_{w_2}) form a synchronized matching in G_A.

We now construct a corresponding allele-pair set over the alphabet $\{0, 1\}$. The allele-pair set A is composed of two sets of alleles: A set A' that contains an allele-pair for every edge in E and a set A'' of "blocking" allele pairs. Let $n = |V|$. The building blocks of our construction are strings $\langle v_i \rangle$ that encode vertices of V by a length-n binary string with 1 at position i and 0 otherwise. For every edge $e = (v_i, v_j)$, we have the allele-pair (s_e^0, s_e^1):

$$s_e^0 \equiv 010 \cdot \langle v_i \rangle \cdot 01 \cdot 0 \cdot 10 \cdot \langle v_j \rangle \cdot 011$$

$$s_e^1 \equiv 010 \cdot \langle v_i \rangle \cdot 01 \cdot 1 \cdot 10 \cdot \langle v_j \rangle \cdot 011$$

where s_e^0 and s_e^1 are of length $(2n+11)$ with the SNP site located in their middle position. We set the feature length $k \equiv n + 6$. For constructing the additional blocking allele-pairs, consider the set B that contains, for every $e \in E$, all length-k substrings of s_e^0 and s_e^1 that do not include the last position of s_e^0 and the first position of s_e^1, respectively. Define the mate of every substring in B to be the same substring with the middle position flipped. For every pair of substring and its mate in B, we remove one of these arbitrarily. Set A'' is now defined to contain c allele-pairs for each substring in B, composed of the substring and its mate. In particular, for any $v_i \in V$, $010 \cdot \langle v_i \rangle \cdot 011$ is not an allele in $S_{A''}$.

By construction, two copies of an allele-pair in A'' cannot be part of the same assignable set. Also, there is a clear 1–1 correspondence between the edges of G and allele-pairs in A'. Hence, given an

edge coloring of G, it induces a partition into assignable subsets. Conversely, in a partition to c assignable subsets, no subset can contain two allele-pairs that correspond to adjacent edges. This holds, since for every $e = (v_i, v_j) \in E$, the unique features that correspond to s_e^0 or s_e^1 are $010 \cdot \langle v_j \rangle \cdot 011$ and $010 \cdot \langle v_i \rangle \cdot 011$, respectively, as all their other features are covered by the blocking alleles. ■

The following hardness-of-approximation result follows from Holyer (1981):

Corollary 3. *It is NP-hard to approximate* SMC *to within a factor of* $(4/3 - \epsilon)$*, for any* $\epsilon > 0$.

The complexity of MSM is currently open. However, we can show that a natural extension of the problem to SNPs with at most three alleles is NP-hard. In this extension, the input instance is an SF-like graph, and the goal is to find a maximum number of vertex-disjoint synchronized quadruples, consisting of a SNP and three informative features, one for each of its alleles. We do not report the result here as it is out of scope.

5.2. Approximating MAPS

We now devise an approximation algorithm for MAPS. Let G be an input SF graph. The algorithm has two stages. In its first stage, we compute a maximal synchronized matching in G; in the second stage, we transform this matching into a collection of induced synchronized submatchings, choosing the largest as the output assignable set. For the first stage we use an approximation algorithm to MSM, which we present next.

Theorem 6. MAXIMUM SYNCHRONIZED MATCHING *admits a polynomial-time approximation scheme.*

Proof. *Algorithm.* We use a greedy algorithm to find a maximal synchronized matching. In its first phase, we successively try to extend the current synchronized matching by adding a disjoint synchronized triple. The first phase ends when no extension is possible. In the second phase, for a positive integer r, we repeat the following step until no improvement is possible: For every set of r synchronized triples in the current solution, we try to improve the solution by replacing the r synchronized triples with $r+1$ triples. We denote the resulting two-phase greedy algorithm by H_r. Clearly, H_r is polynomial for constant r. Denote the resulting synchronized matching by M_{app}, and let V_{app} be the set of allele-pair vertices included in M_{app}.

Approximation ratio. Let M_{opt} denote an optimum solution, and let V_{opt} denote the allele-pair vertices occurring in M_{opt}. In the following, we will show that $|V_{opt}| \leq (1 + \frac{2}{r+1})|V_{app}|$.

An allele-pair vertex in $V_{opt} \setminus V_{app}$ is called *blocked*. An allele-pair vertex in $V_{app} \setminus V_{opt}$ is called *greedy-only*. For a blocked vertex v, participating in a synchronized triple $(v, u_1, u_2) \in M_{opt}$, define its set $\mathrm{dom}(v)$ of *dominating* feature vertices as follows: For $i = 1, 2$, feature vertex u_i is in $\mathrm{dom}(v)$ if u_i occurs in a triple of M_{app}; inductively, $u_i' \in \mathrm{dom}(v)$ if there exist $u_i'' \in \mathrm{dom}(v)$ and $v' \in V$ such that the following three conditions are satisfied: (a) M_{opt} contains a triple (v', u_1', u_2'); (b) M_{app} contains a triple (v', u_1'', u_2''); and (c) u_i' occurs in a triple of M_{app}. An example is given in Fig. 2. Note that a feature vertex can dominate

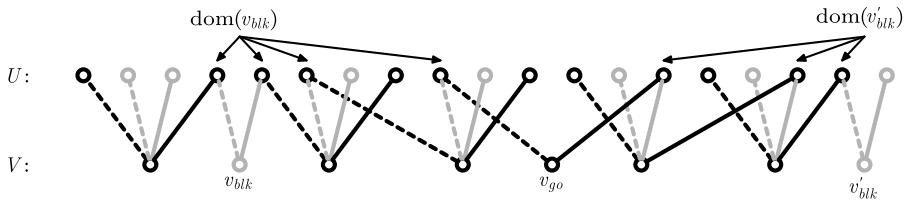

FIG. 2. Dominating features and dominating components. An example of an SF graph $G = (V, U, E_r \cup E_b)$, where blue edges appear dashed and red edges appear solid. Triples of an optimal solution are shown in gray; triples in the solution found by the greedy algorithm appear in black. Vertices v_{blk}, v'_{blk} are blocked; vertex v_{go} is greedy-only. The sets $\mathrm{dom}(v_{blk})$ and $\mathrm{dom}(v'_{blk})$ are indicated; the corresponding dominating components are $U_{dom}(v_{blk}) = U_{dom}(v'_{blk}) = \mathrm{dom}(v_{blk}) \cup \mathrm{dom}(v'_{blk})$.

at most one blocked allele-pair. Finally, we define the *dominating component* $U_{dom}(v) \subseteq U$ of a blocked allele-pair vertex v as follows: Consider a graph G_{dc} on all blocked vertices in which two blocked vertices v and v' share an edge if there is a greedy-only allele-pair vertex v'' with $(v'', u''_1, u''_2) \in M_{app}$ such that u''_1 is a dominating vertex of v and u''_2 is a dominating vertex of v', or vice versa. Let $C(v)$ be the connected component containing v in G_{dc}. Then $U_{dom}(v) \equiv \bigcup_{v' \in C(v)} \mathrm{dom}(v')$. Note that the maximum degree of a vertex in G_{dc} is at most two. Therefore, the connected components of G_{dc} are either paths or cycles.

For a dominating component $U_{dom}(v)$, we let $V_{blk}(v) \subseteq V$ denote the set of blocked vertices v' for which $(v', u'_1, u'_2) \in M_{opt}$ while $u'_1 \in U_{dom}(v)$ or $u'_2 \in U_{dom}(v)$. We denote by $V_{go}(v) \subseteq V$ the set of greedy-only vertices v'' for which $(v'', u''_1, u''_2) \in M_{app}$ while $u''_1 \in U_{dom}(v)$ or $u''_2 \in U_{dom}(v)$. Thus, within $C(v)$, $V_{blk}(v)$ are the allele-pair vertices that are "gained," and $V_{go}(v)$ the allele-pair vertices that are "lost" in M_{opt} compared to M_{app}. The following two claims are essential in proving the approximation ratio:

1. $|V_{blk}(v)| = |V_{go}(v)|$ or $|V_{blk}(v)| = |V_{go}(v)| + 1$. Clearly, $|V_{blk}(v)| \geq |V_{go}(v)|$ or, else, M_{opt} could be improved by (a) replacing the triples involving vertices of $V_{blk}(v)$ with those in M_{app} that involve $V_{go}(v)$, and (b) replacing each triple $(v, u, u') \in M_{opt}$ that involves features from $U_{dom}(v)$ with a triple from M_{app} that includes v. Since the elements of $V_{blk}(v)$ are the vertices of a connected component in G_{dc}, whose edges are in 1–1 correspondence with $V_{go}(v)$, we must have $|V_{blk}(v)| \leq |V_{go}(v)| + 1$.
2. If $|V_{blk}(v)| = |V_{go}(v)|+1$ then $|U_{dom}(v)| \geq r+1$. Suppose to the contrary that $|V_{blk}(v)| = |V_{go}(v)|+1$ but $|U_{dom}(v)| \leq r$. Hence, H_r can remove all triples (at most r) from M_{app} that contain a feature from $U_{dom}(v)$ and replace them with the $r+1$ triples in M_{opt} that involve the vertices in $V_{blk}(v)$, a contradiction.

Assume now that G_{dc} has d components, d' of which correspond to allele-pair vertices v such that $|V_{blk}(v)| = |V_{go}(v)|+1$. By claim (2), $|V_{app}| \geq (d-d')+\frac{d'(r+1)}{2}$. Claim (1) implies that $|V_{opt}| \leq |V_{app}|+d'$. Therefore, the approximation factor can be bounded as follows:

$$\frac{|V_{opt}|}{|V_{app}|} \leq \frac{d + \dfrac{d'(r+1)}{2}}{d - d' + \dfrac{d'(r+1)}{2}} \leq \frac{d + \dfrac{d(r+1)}{2}}{\dfrac{d(r+1)}{2}} \leq 1 + \frac{2}{r+1}. \qquad \blacksquare$$

Theorem 6 shows that the approximation ratio achieved by H_r is $1 + \frac{2}{r+1}$, which can be easily shown to be tight. It is interesting to compare this result to that obtained by De Bontridder *et al.* (2003). They analyzed the same algorithm for the problem of packing paths of length 2 in a given graph (PP2). In contrast to MSM, instances of PP2 are not necessarily bipartite, and their edges are not colored. De Bontridder *et al.* show that for PP2, H_r achieves an approximation factor of 3 for $r = 0$ and of 2 for $r = 1$. For these values of r, their results coincide with ours. However, they also show that PP2 is APX-hard, implying that the approximation factor achieved by H_r converges to a constant greater than 1. In particular, they analyzed the performance of H_r for $r = 2, 3, 4$ and prove that the factors achieved are $\frac{9}{5}$, $\frac{11}{7}$, and $\frac{3}{2}$, respectively, different than our results for MSM. We note that for PP2 it remains open to determine the exact constant to which the approximation ratio of the greedy algorithm converges with growing values of r.

We are now also ready to state our approximation algorithm for MAPS.

Theorem 7. *There is a polynomial approximation algorithm to* MAPS *with ratio* $(2l-1)(1+\frac{2}{r+1})$, *where* l *is the length of the input sequences.*

Proof. Our approximation algorithm has two stages. In the first stage, we compute an approximate synchronized matching in G, using the greedy algorithm H_r given in the proof of Theorem 6. In the second stage, we partition the allele-pairs in this synchronized matching into assignable subsets using the method of Theorem 2. Since each allele-pair has at most $l+1$ distinct features (at most $l-2k+1$ noninformative and at most $2k$ informative features), two of which participate in a synchronized triple with that allele-pair, we can partition the allele-pairs into at most $2l-1$ assignable subsets. Finally, we pick the largest assignable subset to be the output solution. In combination with the result of Theorem 6, we get the stated ratio. $\blacksquare$

It is interesting to note that MAPS is fixed-parameter tractable with respect to fixed feature length k, i.e., solvable in $f(k) \cdot |A|^{O(1)}$ time for some function f: The algorithm iterates over all possible feature sets. For each set F, the algorithm checks whether there exists a corresponding assignable allele-pair set, whose unique features are exactly the features in F. This is done by building a graph in which the vertices are the features in F, and the edges connect two features f and f' for which there exists an allele-pair v such that (v, f, f') is a synchronized triple and v is not adjacent to any other feature in F. An assignable set exists if and only if the latter graph has a perfect matching. The largest feature set identified by the algorithm yields the maximum allele-pair set.

5.3. *Approximating MAPC*

In this section, we present our results for MAPC and the related SYNCHRONIZED MATCHING COVER problem. Our main result is a polynomial 2-approximation algorithm for SYNCHRONIZED MATCHING COVER, leading to a polynomial-time $(4l - 2)$-approximation algorithm for MAPC, where l denotes the length of the input sequences.

Theorem 8. *There is a polynomial 2-approximation algorithm to* SYNCHRONIZED MATCHING COVER.

Proof. The algorithm works in two phases: In the first phase we use maximum network flow to find a lower bound c_{opt} on the optimum solution and to choose for each allele-pair two representative (informative) features. In the second phase, we partition the resulting triples into assignable sets using graph coloring. In the following, we describe the two phases in detail.

In the first phase, we restrict attention to the subgraph $G' = (V, U, E_r \cup E_b)$ of the SF graph G, spanning only the informative edges, and convert it into a network graph $N = (U', E')$, as illustrated in Fig. 3. The vertex set U' consists of a source vertex s, two vertices v' and v'' for each vertex $v \in V$, the vertices in U, and a sink vertex t. The edge set E' contains capacity-1 edges (s, v') and (s, v''), for every $v \in V$; capacity-1 edges (v', u) for all $(v, u) \in E_r$; capacity-1 edges (v'', u) for all $(v, u) \in E_b$; and capacity-c edges (u, t), for all $u \in U$. For $c = 1, 2, \ldots$ we compute a maximum network flow in N from s to t and stop when the value of the flow reaches $2|V|$. The value of c at that point, denoted c_{opt}, serves as a lower bound on the size of a synchronized matching cover of G.

In the second phase, we use the flow computed in the previous phase to construct a conflict graph $C = (V, E'')$. With each allele-pair, we associate a synchronized triple that contains this pair, as implied by the computed flow. There is an edge in C between two allele-pairs if and only if their associated triples overlap (i.e., share some feature). We now color this graph using SLO coloring. Clearly, each color class corresponds to an assignable set of allele-pairs. Due to the capacity constraints on the network flow, the maximum degree of a vertex in C is $2(c_{opt} - 1)$. Hence, C can be colored using at most $2c_{opt} - 1$ colors. The approximation ratio follows. ■

Corollary 1. *There is a polynomial approximation algorithm to* MAPC *with ratio* $4l - 2$, *where* l *is the length of the allele-pair sequences.*

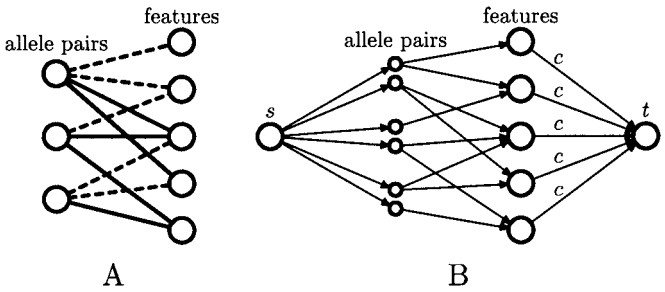

FIG. 3. Network graph construction. (**A**) The subgraph $G' = (V, U, E_r \cup E_b)$ of the allele-pair graph (edges in E_r are dashed and edges in E_b are solid). (**B**) The corresponding network graph N (unlabeled edges have capacity 1).

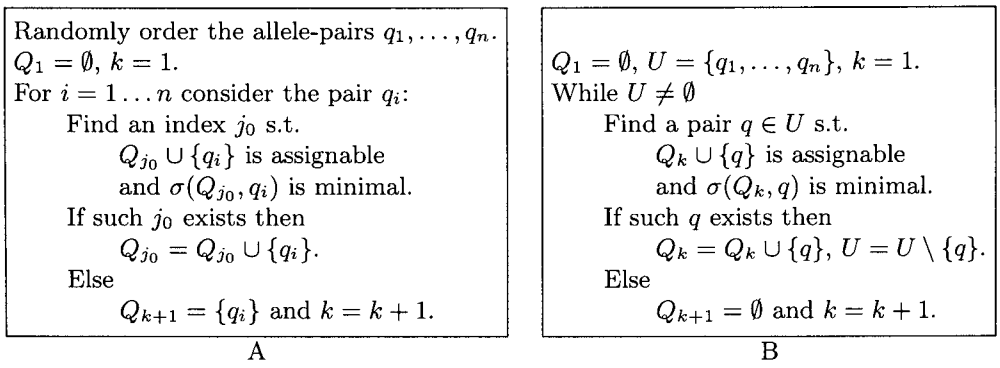

FIG. 4. The minimal partition (**A**) and maximal set (**B**) algorithmic schemes. The input allele-pair set is $A = \{q_1, \ldots, q_n\}$.

Proof. We apply the approximation algorithm to SMC on the input graph G. We then use the method described in the proof of Theorem 7 to transform each synchronized matching in the cover into a collection of at most $(2l - 1)$ assignable subsets. ■

We note that a similar algorithmic problem has been studied by Kivija *et al.* (2002).

6. HEURISTIC APPROACHES

In addition to the approximation algorithm, we devised two heuristic approaches to MAC and MAPC. For simplicity, we present these approaches for the allele-pair case, but similar procedures are applied also in the single-allele case.

The first heuristic is called *minimal partition (MP)*. We allocate one SNP at a time, inserting it into the subset of the given allele-pair set A that "best" accommodates it. Subsets are ranked by a potential function $\sigma : 2^A \times A \rightarrow \mathcal{R}$ which assigns every pair of $A' \subseteq A$ and $q \in A$ a real-valued "potential" $\sigma(A', q)$. We start a new subset only when the target cannot be accommodated in an existing subset. The second heuristic is called *maximal set (MS)*. We attempt to construct a large assignable subset of SNPs, each time adding to the current set a SNP whose ranking according to the potential function is the lowest. When this set cannot be extended anymore, we iteratively call the process on the remaining SNPs. These schemes are given in Fig. 4.

We experimented with several potential functions $\sigma(A', q)$, described below. For a subset of allele-pairs A', denote by $N(A')$ the features that are covered by the alleles in $S_{A'}$. The potential functions are (σ_1), the number of features of q that are not covered by the allele-pairs in A', i.e., $\sigma_1(A', q) = |N(\{q\}) \setminus N(A')|$; ($\sigma_2$), the number of allele-pairs q' that were not assigned yet and have features that are not covered by A' but are covered by q, i.e., $N(\{q'\}) \cap (N(\{q\}) \setminus N(A')) \neq \emptyset$; and ($\sigma_3$), the number of allele-pairs q' that were not assigned yet and for which $A' \cup \{q'\}$ is assignable, but $A' \cup \{q, q'\}$ is not. We obtained similar results using the three functions; in the following, we present only results that were obtained using σ_1, whose computation is the most efficient (by a factor of n).

7. EXPERIMENTAL RESULTS

We implemented the algorithmic approaches for MAPC and MAC presented in the previous sections, including the approximation algorithm for MAPC (Apx) and the heuristic schemes (MP and MS). The approximation algorithm was slightly modified compared to Theorem 8 and Corollary 1: After having computed a network flow in its first phase, we "improve" the chosen synchronized triples without altering the total flow in the network. The improvement aims at minimizing the sum over all features included in the synchronized triples defined by the flow of the number of allele-pairs that have that feature. The minimization is done in a greedy way: For a feature u, let $N(u)$ denote the set of allele-pairs containing u.

We determine the allele v for which its currently assigned feature u can be replaced by a new feature u' such that $|N(u') \setminus N(u)|$ is maximum. We successively do these replacements until no improvement is possible. Notably, this step does not affect the approximation guarantee of the algorithm but compensates for the fact that the first phase ignores the noninformative edges of the allele-pair graph and, therefore, often yields unfavorable assignments in this respect. Intuitively, this improvement decreases the number of edges of the conflict graph which is colored in the second phase of the approximation algorithm.

We applied the approaches to synthetic SNP data and to data that was simulated based on real SNP sequences. We evaluated the quality of the solutions obtained by computing a lower bound on the size of an optimum solution. The lower bound, denoted LB, is based on the size s of a maximal clique in a conflict graph whose vertices are allele-pairs and whose edges connect two allele-pairs that cannot be assigned to the same assay. Maximal cliques in this graph are computed using a greedy search. If $s > 1$, then LB is set to s. Otherwise, we check (in polynomial time) whether the entire set of alleles is assignable. If the answer is positive, then LB is set to 1 and is tight. If the answer is negative, then we set LB to 2.

We also evaluated an alternative lower bound arising from the theoretical analysis of SMC: It is the minimum capacity c_{opt} that is computed by the network flow algorithm of Theorem 8. In the experiments, however, this bound performed worse than LB and, consequently, we do not report results on this.

The results that we present here cover both generic technologies that were introduced in Section 2, namely, all-k-mer arrays and mass-spectrometry assays. For each target sequence, we include features from both the target and its Watson–Crick complement. When using real sequence data, we omit close SNPs if one lies in the flanking sequence of the other. We also omit SNPs for which one allele does not have an informative feature. We report on results for both perfect and noisy models of hybridization.

7.1. Synthetic data

Our first goal was to evaluate the performance of the three algorithmic approaches on synthetic data. We generated at random 41-long sequences for varying number of SNPs. For each sequence, we chose at random two distinct nucleotides, representing two alleles, to occupy the 21st base of the sequence. We simulated experiments with an all-k-mer array, where k ranged from 6 to 8.

To model noise in the hybridizations, we prepared two versions of each dataset. In the first version, a k-mer was considered a feature of a sequence if it occurred in it. In the second version, we used a simple noise model based on the fact that an enzymatic reaction (such as extension) on a k-mer will not occur when there is no stable binding close to the reactive (3′, in case of polymerase extension) end of the primer (Jeff Sampson, unpublished data). Thus, in our model, a k-mer was considered a feature of a sequence if the k-mer appeared in the sequence with up to one mismatch, where the mismatch was constrained to occur in one of the nucleotides that are located at least 6 bp from the extension end of the primer. The results, averaged over 10 runs, are presented in Table 1 and are depicted in Fig. 5. We also simulated datasets in which the features of each sequence were taken to be the molecular weights of its constituent k-mers. These results, which measure the performance of the mass-spectrometry genotyping technique, are given in Table 2. In both sets of results, the heuristic approaches outperform the approximation algorithm.

Next, we applied the MS and MP heuristics to the perfect hybridization data in their single-allele version, allowing alleles from the same allele-pair to be assigned to different sets. This experiment shows what we gain or lose by coupling the alleles of each SNP together. The results are summarized in Table 3. Since the single-allele version is less constrained, we have expected that the solutions will contain fewer arrays. In practice, it turned out that the heuristics do not exploit well the added freedom in the single-allele version, and the solution sizes were often slightly higher than in the allele-pair case. Moreover, in terms of amplification reactions, we see a significant difference between the two versions. In the single-allele case, a considerable fraction of the allele-pairs are split between the assays, implying a significant increase in the number of amplification reactions that are required for the genotyping process.

7.2. Simulations using real sequence data

In order to generate data that is representative of real genotyping experiments, we used real SNP sequence data from various sources. First, we extracted from the public SNP database (Sherry *et al.*, 2001) 41 bp-long sequences flanking the first 1,000 reference SNPs of each chromosome, omitting those SNPs not showing a sufficiently long flanking sequence, or not having a unique informative feature for each allele in the pair.

TABLE 1. PERFORMANCE ON SYNTHETIC DATA FOR ALL k-MER ARRAYS[a]

	6-mers				*7-mers*				*8-mers*			
# SNPs	*MP*	*MS*	*Apx*	*LB*	*MP*	*MS*	*Apx*	*LB*	*MP*	*MS*	*Apx*	*LB*
200	4.1	4.4	5.2	1.9	2.0	2.0	2.0	1.0	1.1	1.1	1.1	1.0
200*	—	—	—	—	4.0	4.0	4.9	1.5	2.0	2.2	2.1	1.0
400	6.8	6.9	9.8	2.0	3.0	3.0	3.0	1.6	2.0	2.0	2.0	1.0
400*	—	—	—	—	6.0	6.0	9.0	2.0	3.0	3.3	4.0	1.1
600	9.0	9.0	14.1	2.1	3.6	3.9	4.0	1.8	2.0	2.0	2.0	1.2
600*	—	—	—	—	8.0	8.1	12.3	2.0	4.0	4.0	6.0	1.2
800	11.1	11.1	17.9	2.1	4.1	4.1	5.0	1.9	2.0	2.0	2.0	1.3
800*	—	—	—	—	10.0	10.0	16.1	2.1	5.0	5.0	7.9	1.7
1000	13.2	13.1	21.7	2.3	5.0	5.0	6.7	2.0	2.1	2.2	2.1	1.1
1000*	—	—	—	—	12.0	12.0	19.7	2.0	5.8	6.0	9.0	2.0

[a]Datasets that were produced using the noisy model of hybridization are denoted by an asterisk. Note that for 6-mers the perfect and noisy model are identical.

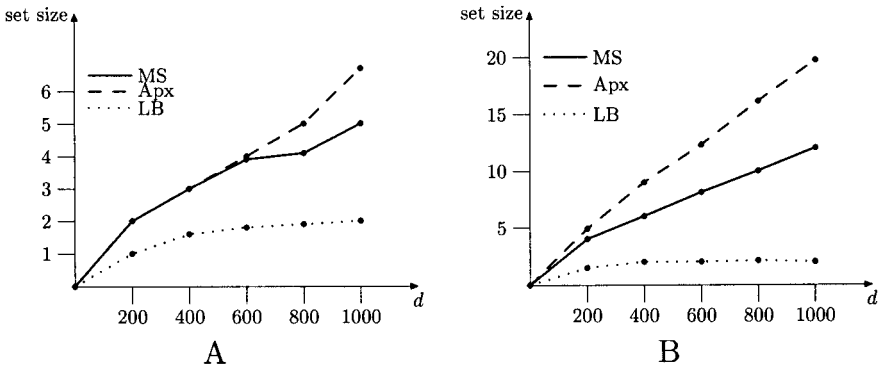

FIG. 5. Performance on synthetic data in terms of the number of required arrays, $k = 7$. (**A**) Perfect hybridization data. (**B**) Data produced using the one-mismatch noise model of hybridization.

TABLE 2. PERFORMANCE ON SYNTHETIC DATA FOR MASS-SPECTROMETRY ASSAYS[a]

	6-mers				*7-mers*				*8-mers*			
# SNPs	*MP*	*MS*	*Apx*	*LB*	*MP*	*MS*	*Apx*	*LB*	*MP*	*MS*	*Apx*	*LB*
200	38.4	36.1	51.5	9.2	29.6	27.2	36.2	5.7	22.8	21.3	26.8	4.0
200*	—	—	—	—	83.7	80.9	88.3	48.9	89.6	83.9	92.3	58.3
400	70.1	66.5	91.7	12.0	52.2	48.6	67.1	8.2	40.6	37.7	48.5	5.3
400*	—	—	—	—	153.1	145.2	173	79.1	162.8	152.9	164.8	93.8
600	99.7	93.2	125.3	16.0	73.0	68.8	85.6	10.1	56.9	52.8	66.7	6.1
600*	—	—	—	—	217.3	206.7	230.7	102.5	230.7	219.4	236.4	128.6
800	128.7	119.6	159.6	17.2	94.1	87.7	113.3	11.8	72.4	67.5	85.3	6.8
800*	—	—	—	—	284.8	270.4	309.7	134.5	301.6	284.8	304.3	163.7
1000	157.6	146.8	192.7	20.8	114.6	106.2	144.3	12.7	87.6	81.8	104.7	7.5
1000*	—	—	—	—	349.6	330.3	391.7	154.3	365.1	342.6	374.8	189.7

[a]Datasets that were produced using the molecular weight of its constituent k-mers as features, the datasets under the noise model of hybridization are denoted by an asterisk.

Table 3. Performance of the Single-Allele Version on Synthetic Data, in Terms of the Number of Required Arrays (# Sets) and the Number of Required Amplification Reactions[a]

	6-mers				*7-mers*				*8-mers*			
	MP		*MS*		*MP*		*MS*		*MP*		*MS*	
# SNPs	*# Sets*	*ΔAR*	*# Sets*	*ΔAR*	*# Sets*	*ΔAR*	*# Sets*	*ΔAR*	*# Sets*	*ΔAR*	*# Sets*	*ΔAR*
200	4.2 (0)	24.4	4.1 (−0.1)	39.7	2.0 (0)	4.5	2.0 (0)	8.8	1.1 (0)	0.0	1.1 (0)	0.0
200*	—	—	—	—	4.1 (+0.1)	30.0	4.0 (0)	42.5	2.2 (+0.2)	11.1	2.4 (+0.4)	19.2
400	6.9 (0)	69.3	7.0 (+0.2)	97.2	3.0 (0)	25.8	3.0 (0)	47.1	2.0 (0)	1.2	2.0 (0)	1.9
400*	—	—	—	—	6.4 (+0.3)	94.1	6.4 (+0.3)	113.1	3.1 (+0.1)	51.5	3.4 (+0.4)	69.2
600	9.0 (−0.1)	143.5	9.0 (0)	170.3	3.7 (−0.1)	59.7	3.8 (+0.1)	86.3	2.0 (0)	5.3	2.0 (0)	9.0
600*	—	—	—	—	8.9 (+0.9)	173.1	9.0 (+1.0)	197.4	4.0 (0)	98.5	4.1 (+0.1)	132.6
800	11 (0)	198.7	11.1 (0)	252.7	4.1 (+0.1)	97.9	4.1 (0)	136.3	2.0 (0)	17.7	2.0 (−0.1)	33.4
800*	—	—	—	—	10.7 (+0.7)	246.1	10.6 (+0.6)	284.1	5.0 (0)	157.3	5.0 (0)	210.9
1000	13.2 (0)	275.0	13.2 (0)	324.3	5.0 (0)	144.8	5.0 (−0.1)	200.5	2.0 (0)	37.4	2.1 (+0.1)	63.3
1000*	—	—	—	—	12.2 (0)	341.8	12.6 (−0.4)	358.0	5.9 (+0.1)	227.0	6.0 (0)	287.2

[a]The difference in solution cardinalities compared to the allele-pair results is given in parentheses. ΔAR denotes the number of the additional amplification reactions implied by the single-allele case, i.e., the number of allele-pairs that are split between two arrays. The results are averaged over 10 runs.

Table 4. Performance on Simulated Data, in Terms of the Number of Required Arrays, Considering a Noisy Hybridization Model[a]

Chr	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*X*	*Y*
Apx	10	10	10	11	11	10	10	10	10	10	10	10	10	10	10	10	9	10	10	10	10	9	10	10
MP	8	9	10	9	10	8	10	8	8	9	8	8	9	8	10	9	7	10	9	9	8	9	9	10
MS	7	7	7	7	7	7	7	7	7	7	7	7	7	8	8	7	7	8	8	8	9	8	8	8
LB	3	3	6	3	5	2	4	3	3	3	3	3	4	4	4	5	3	3	4	6	5	5	5	4

[a]The data were simulated based on known SNPs from human chromosomes. Each dataset corresponds to a chromosome and contains its first 1,000 reference SNPs, $k = 8$.

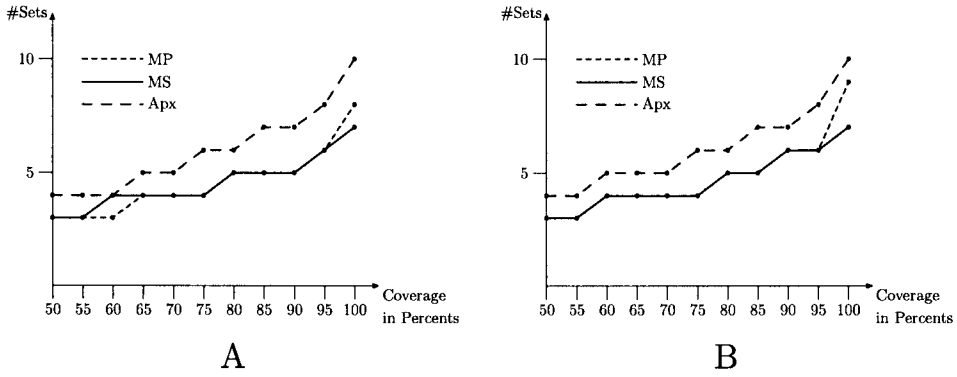

FIG. 6. Percentage of coverage for data simulated using known SNPs from human chromosomes. For percentages p ranging from 50% to 100%, depicted are the number of sets required to cover at least p percents of the allele-pairs in the data. (**A**) Performance on chromosome 1. (**B**) Performance on chromosome 16.

The results of applying our algorithms to this data, assuming experiments with an all-8-mer array, are summarized in Table 4. It can be seen that the approximation algorithm gives in practice an approximation ratio of 2.6 on average (with respect to LB), which is much better than its performance guarantee. However, both heuristics outperform the approximation, with the MS heuristic yielding better results in most cases. As can be observed, the results on real sequence data are slightly worse than their random data counterparts (i.e., higher-cardinality solutions are obtained for the same number of SNPs).

Studying the distribution of the sizes of the assignable sets, we discovered that, typically, about half the sets in a solution cover over 75% of the allele-pairs. The remaining sets are small and contain few SNPs each. Figure 6 depicts the number of sets in the solutions of chromosomes 1 and 16 as a function of the required coverage percentage.

Second, we compiled several datasets spanning the SNPs in the regions of specific disease-related genes, using data available from the NIEHS Environmental Genome Project (*www.egp.gs.washington.edu/data/*). For each dataset we used 41-long SNP sequences. The results using all k-mer arrays are summarized in Table 5. In many of the shown cases, one of the algorithms finds an optimal solution. Note that, in Table 5, LB also incorporates checking whether one array is sufficient, which can be done in polynomial time. In all these experiments, the MS algorithm outperformed the approximation algorithm and produced on average covers within a factor of 1.4 of the lower bound, when considering 8-mers.

8. CONCLUSION

In this paper, we have studied the problem of designing genotyping experiments so as to maximize the multiplexing rate of several generic SNP genotyping schemes. We devised approximation algorithms and practical heuristics for the multiplexing problems. We tested our approaches on an extensive collection of synthetic and simulated datasets and showed that a heuristic that is based on a set-cover approach achieves optimal or near-optimal designs in many cases. On realistic examples that involve SNPs in the regions of disease-related genes, our designs could be applied with as few as four arrays. This demonstrates the applicability of the approach in enabling flexible, region-specific SNP genotyping.

From the computational point of view, we note two interesting observations derived from analyzing our experimental results. First, the approximation ratio achieved by our approximation algorithm is, in practice, significantly better than its guaranteed ratio. Second, as often happens with other NP-complete problems, a practical heuristic with no approximation guarantee out-performs a theoretically proven approximation algorithm. An open problem is to determine the complexity of MAXIMUM SYNCHRONIZED MATCHING. Also, exact algorithms for MAPC are of interest.

While our work has given initial promising results, several extensions and refinements to our method can be explored. For example, tighter lower bounds for the multiplexing problem are of interest, as are extensions of the algorithms to handle SNPs with more than two alleles.

TABLE 5. PERFORMANCE ON SIMULATED DATA IN TERMS OF THE NUMBER OF REQUIRED ARRAYS, USING THE NOISY MODEL OF HYBRIDIZATION[a]

	6-mers								*8-mers*							
	ace	*apoA1*	*apoB*	*atm*	*brca1*	*gata3*	*tnfap1*	*wt1*	*ace*	*apoA1*	*apoB*	*atm*	*brca1*	*gata3*	*tnfap1*	*wt1*
# SNPs	88	26	143	309	220	96	40	130	88	26	145	310	221	96	40	130
MP	3	2	4	8	7	3	2	4	2	1	2	4	4	2	2	2
MS	3	2	4	7	6	4	3	4	2	1	2	4	4	2	2	2
Apx	3	2	4	9	7	3	2	4	2	1	2	4	5	2	2	2
LB	2	2	2	3	5	2	2	2	2	2	2	3	4	2	2	2

[a]The data were simulated based on known SNPs from disease-related genes. Each dataset corresponds to a gene and contains known SNPs in its region.

ACKNOWLEDGMENTS

We thank Jeff Sampson, Anya Tsalenko, and Bo Curry of Agilent Technologies for discussion and early valuable insight on the heuristics. We also thank Ron Shamir and Richard M. Karp for helpful discussions. Part of this work was done while RS and JG were doing their postdoctoral work at the International Computer Science Institute, Berkeley. JG was supported through a postdoctoral fellowship by the DAAD (German Academic Exchange Service) and partially by DFG grant NI 369/2. This research was supported in part by NSF ITR Grant CCR-0121555.

REFERENCES

Aumann, Y., Manisterski, E., and Yakhini, Z. 2003. Designing optimally multiplexed SNP genotyping assays. *Proc. 3rd Int. Workshop on Algorithms in Bioinformatics (WABI)*, 320–338.

Ben-Dor, A., Hartman, T., Schwikowski, B., Sharan, R., and Yakhini, Z. 2003. Towards optimally multiplexed applications of universal DNA tag systems. *Proc. 7th Ann. Conf. on Research in Computational Molecular Biology (RECOMB)*, 48–56.

Cameron, K. 1989. Induced matchings. *Disc. Appl. Math.* 24, 97–102.

De Bontridder, K.M.J., Halldorsson, B.V., Halldorsson, M.M., Hurkens, C.A.J., Lenstra, J.K., Ravi, R., and Stougie, L. 2003. Approximation algorithms for the test cover problem. *Mathematical Programming* 98, 477–491.

Garey, M.R., and Johnson, D.S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, San Francisco.

Håstad, J. 1999. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica* 182, 105–142.

Holyer, I. 1981. The NP-completeness of edge colorings. *SIAM J. Comput.* 10, 718–720.

Kivija, T., Arvas, M., Kataja, K., Penttilä, M., Söderlund, H., and Ukkonen, E. 2002. Assigning probes into a small number of pools separable by electrophoresis. *Bioinformatics* 18(1), S199–S206.

Kurg, A., Tõnisson, N., Tollett, J., Georgiou, I., Shumaker, J., and Metspalu, A. 2000. Arrayed primer extension: Solid phase four-color DNA resequencing and mutation detection technology. *Genetic Testing* 4(1), 1–7. Also see *www.asperbio.com/APEX.htm*.

Matula, D., and Beck, L. 1983. Smallest-last ordering and clustering and graph coloring algorithms. *J. ACM* 30, 417–427.

Patil, N., Berno, A.J., Hinds, D.A., *et al.* 2001. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21, *Science* 294(5547), 1719–1723.

Ross, P., *et al.* 1998. High level multiplex genotyping by MALDI-TOF mass-spectometry. *Nature Biotechnol.* 16, 1347–1351.

Sampson, J.R., *et al.* 2001. *Method and mixture reagents for analyzing the nucleotide sequence of nucleic acids by mass spectrometry*. US patent 6,218,118.

Sharan, R., Ben-Dor, A., and Yakhini Z. 2004. Multiplexing schemes for generic SNP genotyping assays. *Proc. 10th Pac. Symp. on Biocomputing*, 140–151.

Sherry, S.T., Ward, M.H., Kholodov, M., Baker, J., Phan, L., Smigielski, E.M., and Sirotkin, K. 2001. dbSNP: The NCBI database of genetic variation. *Nucl. Acids Res.* 29, 308–311. *www.ncbi.nlm.nih.gov/SNP*.

Syvanen, A.C. 1999. From gels to chips: "Minisequencing" primer extension for analysis of point mutations and single nucleotide polymorphisms. *Human Mutatation* 13(1), 1–10.

Address correspondence to:
Roded Sharan
School of Computer Science
Tel-Aviv University
Tel-Aviv 69978, Israel

E-mail: roded@post.tau.ac.il