Efficient Computation of Template Matrices

Asano, T.[(a)]**;Rosselló, F.**[(b)*]**;Valiente, G.** [(c)†]

[(a)] *Japan Advanced Institute of Science and Technology,*
School of Information Science,
Asahidai 1-1, Nomi, Ishikawa 923-1292, Japan
t-asano@jaist.ac.jp

[(b)] *Departament de Ciències Matemàtiques i Informàtica*
Institut Universitari d'Investigació en Ciències de la Salut (IUNICS)
Universitat de les Illes Balears
Campus de la UIB, E-07122 Palma
cesc.rossello@uib.es

[(c)] *Departament de Llenguatges i Sistemes Informàtics*
Universitat Politècnica de Catalunya
Mòdul Omega, Campus Nord, E-08034 Barcelona
valiente@lsi.upc.edu

Abstract. The computation of template matrices is the bottleneck of simple algorithms for perfect phylogeny haplotyping and for perfect phylogeny under mutation and constrained recombination. The fastest algorithms known so far compute them in $O(nm^2)$ time. In this paper, we describe an algorithm for computing template matrices in $O(nm^2/\log(n))$ time. We also present and discuss a conjecture that implies an $O(nm + m^2)$ time algorithm for computing them and, as a consequence, $O(nm + m^2)$ time simple solutions to the perfect phylogeny haplotyping problem and to the perfect phylogeny problem under mutation and constrained recombination, as well as an $O(n^2)$ time solution to the boolean matrix multiplication problem.

Keywords. Template matrix, haplotype, genotype, perfect phylogeny, site consistency, phylogenetic network with recombination, conflict graph, boolean matrix multiplication

*Partially supported by the Spanish DGES project BFM2003-00771 ALBIOM and UE project INTAS IT 04-77-7178.

†Partially supported by the Spanish CICYT TIN2004-07925-C03-01 GRAMMARS, the UE project INTAS IT 04-77-7178, and by the Japan Society for the Promotion of Science through Long-term Invitation Fellowship L05511 for visiting JAIST (Japan Advanced Institute of Science and Technology).

1. Introduction

The problem of inferring haplotype phase from a population of genotypes has received much attention recently, especially on the light of current large-scale efforts to characterize populations in terms of haplotypes. There has also been an increasing interest in the site consistency problem in perfect phylogeny, motivated in part by the characterization of a particular form of phylogenetic networks under mutation and constrained recombination that admit a polynomial-time solution to this problem.

Perfect phylogeny haplotyping (PPH) is the problem of resolving a given set of n diploid sequences into a set of $2n$ haploid sequences that form a perfect phylogeny [6]. The input to the PPH problem is an $n \times m$ *genotype matrix* M over the alphabet $\{0, 1, 2\}$, where the ith row $M[i, *]$ describes the genotype of species s_i, each column $M[*, j]$ represents a polymorphic locus, and each column j for which $M[i, j] = 2$ is a polymorphic site. A solution to the PPH problem is a $2n \times m$ *haplotype matrix* M' over the alphabet $\{0, 1\}$ where each row $M[i, *]$ expands to two rows, say $M'[i, *]$ and $M'[i', *]$, in such a way that: $M'[i, j] = M'[i', j'] = M[i, j]$ for all j such that $M[i, j] \in \{0, 1\}$; $M'[i, j] \neq M'[i', j']$ for all j such that $M[i, j] = 2$; and M' admits a *perfect phylogeny*, which can be characterized for instance as the existence of no pair of columns j_1, j_2 such that the submatrix $M'[*, \{j_1, j_2\}]$ contains each one of the rows $00, 01, 10, 11$.

In a genotype matrix M, columns j, k are called *companion columns* if there is a row i, called a *companion row* for columns j, k, such that $M[i, j] = M[i, k] = 2$. Two companion columns j, k are said to be *forced in-phase* if the expansion of the non-companion row in $M'[*, \{j_1, j_2\}]$ contains $\{00, 11\}$ and *forced out-of-phase* if it contains $\{01, 10\}$. The *genotype graph* $G = (J, E_f \cup E_n)$ for a genotype matrix M has one node for each column in M, one edge in E_f for each pair of companion columns of M that are either forced in-phase or out-of-phase, and one edge in E_n for each pair of companion columns of M that are not forced in-phase or out-of-phase [3]. The obvious algorithm for computing the genotype graph for a given genotype matrix takes $O(nm^2)$ time and, from it, it is very easy either to compute a haplotype matrix or to decide that no such haplotype matrix exists in time $O(nm+m^2)$. Although an $O(nm+m^2)$ solution to the PPH problem has been obtained recently [5], the extreme simplicity of the approach explained above motivates the search for a method to compute the genotype graph in less than $O(nm^2)$ time.

Site consistency in perfect phylogeny (SCPP) is the problem of resolving a given genomic matrix into another one that admits a perfect phylogeny,

by removing the least possible number of loci (columns). While the SCCP problem is NP-hard in phylogenetic networks under mutation alone [4], it becomes polynomial-time solvable in a particular form of phylogenetic networks under mutation and constrained recombination, called *galled-trees* [7]. The key ingredient in the solution is the *conflict graph* for the genomic matrix M, which has one node for each column in M and one edge $\{j, k\}$ for each pair of columns j and k such that there is a row i with $M[i, j] = 0$ and $M[i, k] = 1$, a row i' with $M[i', j] = 1$ and $M[i', k] = 0$, and a row i'' with $M[i'', j] = M[i'', k] = 1$. The obvious algorithm for computing the conflict graph for a given genomic matrix also takes $O(nm^2)$ time, and once the conflict graph for a genomic matrix that can be derived in a galled-tree is known, the site consistency problem can be solved in $O(nm)$ time [1].

Both the genotype graph in the PPH problem and the conflict graph in the SCPP problem can be obtained as combinations of template matrices. The *template matrix* derived from a matrix M for the template ab is the boolean matrix $\mathcal{M}_{ab}$ with $\mathcal{M}_{ab}[j, k] = 1$ if and only if there exists some row i such that $M[i, j] = a$ and $M[i, k] = b$. It can be proved that the forced edges E_f in the genotype graph for a given genotype matrix M are given by $\mathcal{M}_{22} \cap (((\mathcal{M}_{00} \cup \mathcal{M}_{20} \cup \mathcal{M}_{02}) \cap (\mathcal{M}_{11} \cup \mathcal{M}_{21} \cup \mathcal{M}_{12})) \cup ((\mathcal{M}_{10} \cup \mathcal{M}_{20}) \cap (\mathcal{M}_{01} \cup \mathcal{M}_{02})))$, and the non-forced edges E_n are given by $\mathcal{M}_{22} \setminus E_f$ [3, Lemma 2]. On the other hand, the edges in the conflict graph for a given genomic matrix are clearly given by $\mathcal{M}_{01} \cap \mathcal{M}_{10} \cap \mathcal{M}_{11}$.

The obvious algorithm to compute a template matrix from given matrix M takes $O(nm^2)$ time, and thus any improvement in this cost would yield an improvement in the algorithms for the PPH and the SCPP problems described above. This paper presents two contributions to the problem of the efficient computation of template matrices: an $O(nm^2/\log(n))$ time algorithm, and a conjecture that would imply an $O(nm + m^2)$ algorithm. This conjecture also implies a linear time algorithm for boolean matrix multiplication. In the talk we shall provide detailed analytical and computational arguments supporting our conjecture, which we do not include in this extended abstract because of the lack of space. Part of this work was presented in a poster at the RECOMB 2006 conference [2].

2. Template Matrices

The notion of template matrix was introduced in [3].

Definition 1. Let M be an $n \times m$ matrix over an alphabet Σ. The *template matrix* derived from M for the template $ab \in \Sigma^2$ is the $m \times m$ boolean

matrix $\mathcal{M}_{ab}$ defined as follows: for all $1 \leqslant j, k \leqslant m$, $\mathcal{M}_{ab}[j,k] = 1$ if and only if $M[i,j] = a$ and $M[i,k] = b$ for some $1 \leqslant i \leqslant n$.

The obvious algorithm for computing a template matrix $\mathcal{M}_{ab}$ derived from an $n \times m$ matrix M with no duplicate rows or columns, consists in traversing each pair j, k of columns with $1 \leqslant j, k \leqslant m$ and $j \neq k$, until either finding a row i such that $M[i,j] = a$ and $M[i,k] = b$, in which case $\mathcal{M}_{ab}[j,k] = 1$, or exhausting the columns, in which case $\mathcal{M}_{ab}[j,k] = 0$. The total exact cost of this procedure is given by the value $T_{ab}(M)$ defined next.

Definition 2. Let M be an $n \times m$ matrix over an alphabet Σ. For every $1 \leqslant j, k \leqslant m$ with $j \neq k$ and for every $a, b \in \Sigma$, let

$$T_{ab}^{(j,k)}(M) = \begin{cases} \min\{i \mid 1 \leqslant i \leqslant n, M[i,j] = a, M[i,k] = b\} & \text{if it exists} \\ n & \text{otherwise} \end{cases}$$

The *template number* for M and ab is, then,

$$T_{ab}(M) = \sum_{\substack{1 \leqslant j,k \leqslant m \\ j \neq k}} T_{ab}^{(j,k)}(M).$$

Unfortunately, $T_{ab}(M)$ need no be linear in the size of M plus the size of $\mathcal{M}_{ab}$, as the following two simple examples of families of binary $m \times m$ matrices show:

- Let Id_m be the $m \times m$ diagonal matrix. Since there are no $1 \leqslant i, j, k \leqslant m$ with $j \neq k$ such that $\mathrm{Id}_m[i,j] = \mathrm{Id}_m[i,k] = 1$, we have that $T_{11}(\mathrm{Id}_m) = m(m-1)m \notin O(m^2)$.
- Let Tr_m be the upper triangular $m \times m$ matrix defined by $\mathrm{Tr}_m[i,j] = 1$ if $i + j < m$ and $\mathrm{Tr}_m[i,j] = 0$ otherwise. There are $\binom{m}{2}$ pairs of columns (j,k) such that there does not exist any $1 \leqslant i \leqslant m$ with $\mathrm{Tr}_m[i,j] = 1$ and $\mathrm{Tr}_m[i,k] = 0$: namely, those with $j > k$. Each such pair contributes m to $T_{10}(\mathrm{Tr}_m)$, which shows that $T_{10}(\mathrm{Tr}_m) \geqslant \binom{m}{2} \cdot m$ and hence this value does not belong to $O(m^2)$.

3. Efficient Computation of Template Matrices

A more efficient computation of template matrices can be done by compressing the columns of the input matrix. Given an $n \times m$ matrix M over an alphabet Σ, for any $a, b \in \Sigma$, the set P_{ab} of pairs (j,k) for which

there exists a row i such that $M[i,j] = a$ and $M[i,k] = b$, can be enumerated in $O(nm^2/\log(n))$ time, as described below. Since $\mathcal{M}_{ab}[j,k] = 1$ if $(j,k) \in P_{ab}$ and $\mathcal{M}_{ab}[j,k] = 0$ otherwise, this computes this template matrix in $O(nm^2/\log(n))$ time.

The algorithm for the case $a \neq b$ is the following:

1. Replace every a in M by 0, every b by 1, and every other entry by 2; let M_{ab} be the matrix obtained in this way.

2. Set $L = \lceil (\log_3 n)/2 \rceil$. Define a *compressed matrix* M^c_{ab} of size $\lceil n/L \rceil \times m$ by
$$M^c_{ab}[i,j] = \sum_{\ell=0}^{L-1} 3^\ell M_{ab}[iL + \ell, j]\,.$$

3. Define a matrix R of size $3^L \times 3^L$ as follows. Let p and q be any two integers between 0 and $3^L - 1$. Let also I_p and I_q be the 3-ary representations of p and q, respectively, consisting of L digits from $\{0,1,2\}$. Then, do the following:

   $b_0 := 0$;
   $b_1 := 0$;
   if *there is any digit such that* $I_p[i] = 0$ *and* $I_q[i] = 1$ **then**
   $b_0 := 1$;
   if *there is any digit such that* $I_p[i] = 1$ *and* $I_q[i] = 0$ **then**
   $b_1 := 1$;
   $R[p,q] := b_0 + 2b_1$;

4. Do the following:

   $P_{ab} := \emptyset$;
   for $k := 1$ **to** $\lceil n/L \rceil$ **do**
   for $i := 1$ **to** $m - 1$ **do**
   for $j := i + 1$ **to** m **do**
   $p := M^c_{ab}[k,i]$;
   $q := M^c_{ab}[k,j]$;
   if $R[p,q] = 1$ **then**
   $P_{ab} := P_{ab} \cup \{(i,j)\}$
   if $R[p,q] = 2$ **then**
   $P_{ab} := P_{ab} \cup \{(j,i)\}$
   if $R[p,q] = 3$ **then**
   $P_{ab} := P_{ab} \cup \{(i,j),(j,i)\}$

The algorithm for the case $a = b$ is similar, even simpler, and we omit it.

Theorem 1. *The algorithm given above runs in $O(nm^2/\log n)$ time.*

4. The Reduced Template Number Conjecture

Another method for computing template matrices was sketched in [1, Lem. 11]. Given an $n \times m$ matrix M with no duplicate rows or columns over an alphabet Σ, and for each column $1 \leqslant j \leqslant m$ and for each symbol $a \in \Sigma$, let $M_{j,a}$ be the ordered list of maximal intervals $[\ell, r]$ such that $M[i, j] = a$ for all $\ell \leqslant i \leqslant r$. It is easy to see that these lists of intervals can be constructed in $O(nm)$ time, by traversing M in column order.

Once the lists of intervals $(M_{j,a})_{1\leqslant j\leqslant m}$, $(M_{j,b})_{1\leqslant j\leqslant m}$ are available, a template matrix $\mathcal{M}_{ab}$ can be built in $O(nm)$ time as follows. The idea is to enumerate the set P_{ab} of pairs (j, k) for which there exists a row i such that $M[i, j] = a$ and $M[i, k] = b$. Then, as in the previous section, $\mathcal{M}_{ab}[j, k] = 1$ if $(j, k) \in P_{ab}$ and $\mathcal{M}_{ab}[j, k] = 0$ otherwise.

The set P_{ab} can be computed by performing, for each pair j, k of different columns, a simultaneous traversal of $M_{j,a}$ and $M_{k,b}$ during which, upon intervals $[\ell, r]$ of $M_{j,a}$ and $[\ell', r']$ of $M_{k,b}$, we advance along $M_{j,a}$ if $r < \ell'$, we advance along $M_{k,b}$ if $r' < \ell$, and if $\ell \leqslant \ell' \leqslant r$ or $\ell' \leqslant \ell \leqslant r'$, there exists a row i such that $M[i, j] = a$ and $M[i, k] = b$. The total exact cost of this procedure is given by the value $R_{ab}(M)$ defined next.

Definition 3. Given an $n \times m$ matrix M over an alphabet Σ, for every $1 \leqslant j \leqslant m$ and for every $a \in \Sigma$, let $M_{j,a}$ be the ordered set of maximal row intervals $[i_1, i_2]$ such that $M[i_1, j] = \cdots = M[i_2, j] = a$. For every $[i_1, i_2] \in M_{j,a}$, let $\rho^j_a(i_1, i_2)$ be the rank of $[i_1, i_2]$ in it. For every $1 \leqslant j, k \leqslant m$ with $j \neq k$ and for every $a, b \in \Sigma$, let

$$R^{(j,k)}_{ab}(M) = \begin{cases} \min\Big\{\rho^j_a(i_1, i_2) + \rho^k_b(i_3, i_4) \mid\ [i_1, i_2] \in M_{j,a}, [i_3, i_4] \in M_{k,b}, \\ \qquad [i_1, i_2] \cap [i_3, i_4] \neq \emptyset\Big\} & \text{if it exists} \\ |M_{j,a}| + |M_{k,b}| & \text{otherwise} \end{cases}$$

The *reduced template number* for M and ab is, then,

$$R_{ab}(M) = \sum_{\substack{1\leqslant j,k\leqslant m \\ j\neq k}} R^{(j,k)}_{ab}(M).$$

Thus, the reduced template number for a matrix M over an alphabet Σ and $a, b \in \Sigma$ describes the time needed to compute the template matrix $\mathcal{M}_{ab}$ derived from M for the template ab using the method described above.

Example 1. Let Id_m be the $m \times m$ diagonal matrix and Tr_m be the upper triangular $m \times m$ matrix defined at the end of Section 2. Recall from *loc. cit.* that $T_{11}(\mathrm{Id}_m), T_{10}(\mathrm{Tr}_m) \in O(m^3)$. Now, it is straightforward to check that

$$R_{11}(\mathrm{Id}_m) = R_{10}(\mathrm{Tr}_m) = 2m(m-1) \in O(m^2)\,.$$

Definition 4. An $n \times m$ matrix M over an alphabet Σ is said to be *canonical* if it has no duplicate rows or columns and its rows are sorted in lexicographical order.

Notice that any matrix M over an alphabet Σ can be resolved into a canonical matrix M', by removing duplicate rows and columns and sorting the rows in lexicographical order, in $O(nm)$ time by radix sorting techniques [9, App. A.5]. Notice also that, if the original matrix M does not have any duplicate columns, $\mathcal{M}_{ab} = \mathcal{M}'_{ab}$ for all $a, b \in \Sigma$. Therefore, it is enough to consider reduced template numbers of canonical matrices.

Now, we make the following conjecture.

Conjecture 1. For every $n \times m$ canonical matrix M over an alphabet Σ and for every $a, b \in \Sigma$, $R_{ab}(M) \in O(nm + m^2)$.

Notice that if this conjecture is true, then we can compute template matrices in time $O(nm + m^2)$, which provides algorithms for the PPH and the SCPP problems running in this time as described in the Introduction.

As the reader will guess, we have not been able to prove this conjecture. We have obtained explicit formulas for the reduced template numbers for several families of matrices M, in the spirit of Example 1, and they always grow in $O(nm + m^2)$. Furthermore, we have computed the reduced template numbers for 216 150 random binary matrices (50 instances for each pair (n, m), with $2 \leqslant m \leqslant 24$ and $2m \leqslant n \leqslant \min\{m^2, 2^m\}$) under the uniform distribution, which correspond to samples drawn from a population evolving according to a Wright-Fisher neutral model of genetic variation [8], sorting them in lexicographical order by rows before computing the reduced template numbers. Their reduced template numbers turn out to be bounded by $O(nm + m^2)$, with a multiplicative constant in the asymptotic upper bound that grows very slowly relative to the input size.

Finally, let us mention that argument similar to that used in [3, Lemma 5] proves the following result, which shows that Conjecture 1 implies the fast multiplication of boolean matrices.

Lemma 1. *Let $T(n, m)$ be the time needed to construct a template matrix $\mathcal{M}_{ab}$ for any $a, b \in \Sigma$, given an $n \times m$ matrix M over Σ. Two boolean matrices with dimensions $m_1 \times n$ and $n \times m_2$, respectively, can be multiplied in $O(T(n, m_1 + m_2))$ time.*

References

[1] ASANO, T., EVANS, P., UEHARA, R., VALIENTE, G. Site consistency in phylogenetic networks with recombination. In *Algorithms in Bioinformatics*, chapter 2, King's College London Publications (2006), pp. 15–26

[2] ASANO, T., ROSSELLO, F., VALIENTE, G. Template Matrices for Perfect Phylogeny Haplotyping and Site Consistency. In *Proc. 10th Annual Int. Conf. Research in Computational Molecular Biology (Posters)*, to appear.

[3] BAFNA, B., GUSFIELD, D., HANNENHALLI, S., YOSSEPH, S. A note on efficient computation of haplotypes via perfect phylogeny. *J. Comput. Biol.*, 11 (2004), pp. 858–866.

[4] DAY, W.H.E., SANKOFF, D. Computational complexity of inferring phylogenies by compatibility. *Syst. Zool.*, 35 (1986), pp. 224–229.

[5] DING, Z., FILKOV, V., GUSFIELD, D. A linear-time algorithm for the perfect phylogeny haplotyping (PPH) problem. In *Proc. 9th Annual Int. Conf. Research in Computational Molecular Biology*, Lect. Notes Comput. Sci. 3500 (2005), pp. 585–600.

[6] GUSFIELD, D. Haplotyping as perfect phylogeny: Conceptual framework and efficient solutions. In *Proc. 6th Annual Int. Conf. Research in Computational Molecular Biology*, ACM Press (2002), pp. 166–175.

[7] GUSFIELD, D., EDDHU, S., LANGLEY, C. Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. *J. Bioinformatics Comput. Biol.*, 2 (2004), pp. 173–213.

[8] HUDSON, R. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, 18 (2002), pp. 337–338.

[9] VALIENTE, G. *Algorithms on Trees and Graphs.* Springer-Verlag (2002).