MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **62** (2009) 251-259

ISSN 0340 - 6253

An Algorithm for Construction of All Perfect Phylogeny Matrices

Hanieh Mirzaei[a], Sarah Ahmadian[a], Sepideh Mahabadi[a], Mehdi Sadeghi[b], Changiz Eslahchi[c,d,1], Hamid Pezeshk[e]

a: Department of Computer Engineering, Sharif University of Technology, Tehran, Iran.

b: National Institute of Genetic Engineering and Biotechnology, Tehran, Iran.

c: Department of Mathematical Sciences, Shahid Beheshti University, Tehran ,Iran.

d: Institute for Studies in Theoretical Physics and Mathematics, Tehran, Iran.

e: School of Mathematics, Statistics and Computer Science and Center of Excellence in Biomathematics, University of Tehran, Tehran, Iran.

(Received October 14, 2008)

Abstract

We developed a simulation tool, ASILA, for generating all the different matrices containing k different haplotypes on n SNPs, $k, n \in N$ such that these matrices satisfy the perfect phylogeny property or (coalescent property) $k \leq n+1$. We show that the number of these matrices is an exponential function of k or n. The program is available at {http://bioinf.cs.ipm.ac.ir/softwares/ASILA}.

1 Introduction

With the widespread availability of molecular data, computational methods for gene mapping are being developed. It is often the case that the statistical properties and behavior of these methods need to be assessed and tested by simulation. By increasing the rate of the number of computational methods for gene mapping, there is an increasing need for tools that can simulate data appropriate for long genomic regions. A number of simulation programs have

[1] Corresponding author. E-mail address: ch-eslahchi@sbu.ac.ir

been developed and currently been used [1-4]. One of the most popular model is based on the perfect phylogeny property or coalescent. By using a set of haplotypes which satisfy coalescent property we can simulate a long genomic regions which the simulated data does not depend on an existing data set and also is an approximation to the evolutionary processes that produced the real data. The coalescent have been proved to be a powerful simulation tool in this context [5]. Also in a case in which several widely separated regions were being considered, one might simulate these regions independently and unlinked. On the other hand in a situation in which recombination hot spots were present, one might try to independently simulate regions between hot spots. Hudson and Griffiths et al [6,7] introduce some methods for recombination into coalescent model. Therefore a method to construct the set of k different haplotypes on n SNP which satisfies the coalescent property will be very useful for simulating data. When studying evolution, the divergence patterns leading form a single ancestor spices to its contemporary descendants are usually modeled by a tree structure, called phylogenetic tree or phylogeny. Extant spices correspond to the tree leaves while their common progenitor corresponds to the root. Internal nodes correspond to haplotype ancestral spices, which putatively split up and evolved into distinct spices. The character based approach of tree describes contemporary spices by characters. Each character takes on one of the several possible states. The tree is represented by a matrix $A = [A(i, j)]$ where $A(i, j)$ is the state of character j in species i and the i -th row is the character vector of species i. In this paper we concentrate just on binary matrices and assume that the major and minor alleles are represented by 0 and 1, respectively, and the minor allele frequency (MAF) is at most 49%. Therefore in every two columns of A the gamete (0, 0) is seen.

We state that a matrix A satisfies perfect phylogeny property if and only if there are not three different rows r_1, r_2, r_3 and two different columns c_1, c_2 of A such that:

$$\{(A(r_1, c_1), A(r_1, c_2)), (A(r_2, c_1), A(r_2, c_2)), (A(r_3, c_1), A(r_3, c_2))\} = \{(0,1), (1,0), (1,1)\}$$

Pe'er et al [8] gave a graph formulation for perfect phylogeny matrices. Extending this formulation, in section 2 we show that the number of perfect phylogeny matrices (PPM) with k different rows and n columns is greater than the number of different partitions of k into positive integers. In Section 3, we introduce an algorithm for generating all these matrices.

2 A Lower Bound for the Number of PPM's

In this section we show that the number of different PPM with k different rows and n columns is exponentially increased when k or n increase. According to the tree structure of perfect phylogeny trees, every PPM with n columns has at most $n+1$ different rows. Therefore the condition $k \leq n+1$ always holds. First we will give some definitions. For notation and concepts of graph theory [9].

Definition 1: Let $G = (S,T)$ be a bipartite graph with $|S| \leq |T|+1$. G is Good if and only if for every two vertices u and v in S, $N(u) \neq N(v)$.

Definition 2: The bipartite graph $G = (S,T)$ is called $\sum$ - free if there is not any induced P_5 subgraph starting from S in G.

Pe'er et al [8] assigned a bipartite graph to every PPM as follows. Let $A_{k\times n}$ be a matrix with k different rows and n columns. They associated a bipartite graph $G_A = (S,T)$ with $S = \{u_1, u_2, ..., u_k\}$ and $T = \{v_1, v_2, ..., v_n\}$. The $e = u_i v_j$ is an edge of G_A if and only if $A(i, j) = 1$. In Theorem 3 of their paper they showed that G is $\sum$ - free if and only if A is PPM. Let A be a PPM with k different rows (so $k \leq n+1$). Therefore for every two vertices v_i and v_j we have $N(v_i) \neq N(v_j)$ and so G_A is a Good graph. To show that the number of PPM matrices with k different rows and n SNPs exponentially increases with respect to k or n, it is enough to show that the number of non-isomorphic graph $G_A = (S,T)$ with $|S| \leq |T|+1$ increases with respect to k or n. From now a $\sum$ - free and Good graph $G_A = (S,T)$ with $|S| \leq |T|+1$ is called a PPG. First we need a definition.

Definition 3: The set $\{(x_1, y_1), (x_2, y_2), ..., (x_p, y_p)\}$ is called a {*positive-pair-partition*} of (k, n) if

1- $x_i, y_i \in N$, $\sum_{1\leq i\leq p} x_i = k$, and $\sum_{1\leq i\leq p} y_i = n$.

2- $x_i \leq y_i$ for all i.

Definition 4: The set $\{(x_1, y_1),(x_2, y_2),...,(x_p, y_p)\}$ is called a *{pair-partition}* of (k, n) if

1- This set is positive-pair-partition for (k, n) .

or

2- If there exists $1 \le i \le p$ such that $(x_i, y_i) = (1,0)$ and $\{(x_1, y_1),(x_2, y_2),...,(x_p, y_p)\} - \{(x_i, y_i)\}$ is a positive-pair-partition for $(k-1, n)$.

Theorem 1: Let $B = \{(x_1, y_1),(x_2, y_2),...,(x_p, y_p)\}$ be a pair-partition of (k, n) . Then there exists a bipartite graph $G_B = (S,T)$ with the following conditions:

1- $|S| = k$ and $|T| = n$.

2- G_B is -$\sum$- free and Good.

3- The connected components of $G_B = (S,T)$ are $G_1 = (S_1,T_1), G_2 = (S_2,T_2),...,G_p = (S_p,T_p)$ such that $|S_i| = x_i$ and $|T_i| = y_i$, $1 \le i \le p$

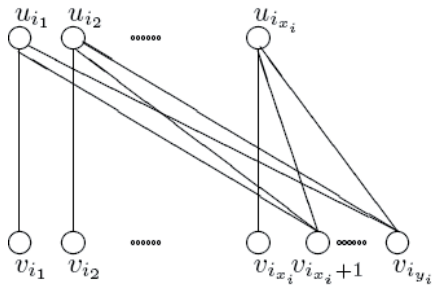
Proof: We construct $G_i = (S_i, T_i)$ by taking the following steps:

- If $(x_i, y_i) \neq (1,0)$ let

$S_i = \{u_{\{i_1\}}, u_{\{i_2\}}, ..., u_{\{i_{x_i}\}}\}$

$T_i = \{v_{\{i_1\}}, v_{\{i_2\}}, ..., v_{\{i_{y_i}\}}\}$

$E(G_i) = \{u_{\{i_k\}} v_{\{i_j\}} \mid 1 \le k \le x_i, x_{i+1} \le j \le y_i\} \bigcup \{u_{\{i_k\}} v_{\{i_k\}}\} \mid 1 \le k \le x_i\}$



- If there exists $1 \le i \le p$ such that $(x_i, y_i) = (1,0)$ let $S_i = \{u_{t_1}\}$, $T_i = \{\}$, $E(G_i) = \{\}$.

It is easy to see that for each i, G_i is a PPG. Hence $G_B = \bigcup_{i=1}^{p} G_i$ satisfies the conditions of the theorem.□

Theorem 2: The number of non isomorphic PPG $G = (S,T)$, $|S| = k$, $|T| = n$ is at least the number of partitions of k.

Proof: Let $B = \{t_i \mid 1 \le i \le m-1, t_i \le t_{i+1}\}$ be a partition for k $(\sum_{i=1}^{n} t_i = k)$. Now we consider the following pair-partition of (k, n) in the following two cases:

- If $k \le n$ let:

$$\{(t_1, t_1), (t_2, t_2), \ldots, (t_{m-1}, t_{m-1}), (t_m, t_{m+n-k})\}.$$

- If $k = n+1$:

$$\{(1,0), (t_1, t_1), \ldots, (t_m, t_m)\}.$$

By Theorem 1 there exists a bipartite graph G_B corresponding to the above partition of (k, n), which satisfies the conditions of theorem. Let $B' = \{t'_1, \ldots, t'_l\}$ be a different partition
of k. If $1 \ne m$ then it is obvious that G_B and $G_{B'}$ are non-isomorphic graphs. Now let $l=m$. in this case it is easy to see that there exists $1 \le i \le m-1$ such that (t_i, t_i) does not appear in pair-partition of B'. So the graph G_B and $G_{B'}$ are not isomorphic. Therefore the number of non isomorphic bipartite perfect phylogeny graph is greater than or equal to the number of different partitions of k. □

According to the above theorem the number of PPM with k different rows is greater than or equal to the number of partitions of k, $p(k)$. Since $p(k) \approx \frac{exp(p\sqrt{\frac{2k}{3}})}{4k\sqrt{3}}$ as $k \to \infty$, hence the number of PPM with k different rows and n columns is exponentially increased when k increases. Similarly one can show the same result for n. Therefore there is not any polynomial time algorithm to generate all different PPM with k different rows and n columns.

3 An Algorithm for Construction of PPM's

Since the number of different PPM's is exponentially increased when k or n increase, so there is not any polynomial time approach to generate all different PPM's with k different rows and n columns. In order to construct all PPM matrices with k different rows and n columns using Theorem 1, it is enough to construct all non isomorphic PPG. In this section we introduce a dynamic-backtrack algorithm, {**ASILA**}, for construction of all non-isomorphic PPGs such as $G = (S,T)$ with $|S| = k$ and $|T| = n$. The algorithm goes through the following steps:

1- a backtrack algorithm we construct all pair-partitions of (k, n) such as: $\{(x_1, y_1),(x_2, y_2),...,(x_p, y_p)\}$ where the pair (x_i, y_i) determines the size of the ith connected component of G, G_i.

2- By a dynamic approach we construct all non-isomorphic graphs G_i for each (x_i, y_i). Since G_i is PPG, by Proposition 6 of Pe'er et al [8] there exists a vertex v in T_i such that $N(v) = S_i$. All possible solutions for G_i-$\{v\} = (S_i, T_i$-$\{v\})$ are found in previous steps of dynamic algorithm. So it's enough to add v to these solutions and connect it to all the vertices in S_i for constructing G_i.

3- By selecting one of the non-isomorphic connected graphs with respect to (x_i, y_i) constructed in step 2, and combining these connected graphs, all non-isomorphic graphs G could be constructed.

Algorithm 3.1 and Algorithm 3.2 are the codes of step 1 of the algorithm. Algorithm 3.3 and 3.4 are the codes of the steps 2 and 3 of the algorithm.

The way that this algorithm is implemented above show that the order of the algorithm is of the number of possible outputs for non-isomorphic PPGs. This program is available at http://bioinf.cs.ipm.ac.ir/softwares/ASILA.

Algorithm 3.1: PARTITION$(x, y, maxX, maxY)$

if x=0 and y=0
  then print *the PartitionSet*
if x=0 or y=0 or maxY=0 or x>y
  then comment: No solution could be found
    return
if $maxY \leq y$
  then for $i \leftarrow 1$ to MIN $(x, maxX, maxY)$
  do
    add pair (i,maxY) to the PartitionSet
    Partition(x-i , y-maxY , i , maxY)
    remove Pair (i , maxY) from the PartitionSet
Partition (x , y , x , maxY-1)

Algorithm 3.2: TOTALPARTITION(k, n)

comment: generate partitions without pair(1,0)
    Partition(k,n,k,n)
comment: generate partitions with pair(1,0)
    add pair(1,0) to Partition(k-1,n,k-1,n).

Algorithm 3.3: MAKEGRAPH($PartitionSet, pNum, index$)

comment: PartitionSet $=\{(x_1, y_1), \cdots, (x_k, y_k)\}$

if $pNum > |PartitionSet|$

then

print the graph and return

num$\leftarrow$ | possible graphs with size$(x_{pNum}, y_{pNum} - 1)$|

if $x_{pNum} = x_{pNum-1}$ and $y_{pNum} = y_{pNum-1}$

then

$start = index$

else $start = 1$

for $i \leftarrow start$ **to** num

- $G = (S, T) \leftarrow$the i^{th} generated graph with size $(x_{pNum}, y_{pNum} - 1)$
- add v to T and connect it to all vertices in S
- makeGraph (PartitionSet , pNum+1 , i)

Algorithm 3.4: TOTALPARTITION(k, n)

for $col \leftarrow 1$ **to** n

do for $row \leftarrow 1$ **to** $col + 1$

do

- TotalPartition(row,col)
- **for** each PartitionSet
  - **do** makeGraph(partitionSet, 1, 1)

Acknowledgement

Mehdi Sadeghi would like to thank the Center of Excellence in Biomathematics at University of Tehran. This research is in part supported by a grant form IPM (NO. CS 1385-1-02).

References

[1] R.R. Hudson, Generating samples under a Wright-Fisher neutral model. *Bioinformatics,* **18** (2002) 337-338.

[2] L. Excoffier, J. Novembre and S. Schneider, SIMCOAL: a general coalescent program for the simulation of molecular data in interconnected populations with arbitrary demography. *J. Heredity*, **91** (2000) 506-509.

[3] G. Montanna, HapSim: a simulation tool for generating haplotype data with pre specified allele frequencies and LD coefficients. *Bioinformatics*, **21** (2003) 4309-4311.

[4] P. Marjoram, and J. Wall, Fast coalescent simulation. *BMC Genet*., **7** (2006)16.

[5] S. F. Schaffner, C. Foo, S. Gabriel, D. Reich, M.J. Daly, and D. Altshuler, Calibrating a coalescent simulation of human genome sequence variation. *Genome Res.,* **15**, (2005)1576-1583.

[6] R.R Hudson, Properties of a neutral allele model with intragenic recombination. *Theor Popul Biol.,* **23**, (1983)183-201.

[7] R.C. Griffiths, and P. Marjoram, An ancestral recombination graph. In Progress in Population Genetics and Human Evolution, IMA Volumes in Mathematics and its Applications Volume 87. Edited by: Donnelly P, Tavare S**}.** *Springer Verlag*, (1997)100-117.

[8] I. Pe'er, T. Pupko, R. Shamir, and R. Sharan, Incomplete directed perfect phylogeny. *SIAM Journal of Computing,* **33**, (2004) 597-607.

[9] R. Diestel, Graph Theory, *Springer-Verlag, New York*, 2000.