Zinc finger gene clusters and tandem gene duplication

Mengxiang Tang* Michael Waterman† Shibu Yooseph‡

Department of Mathematics
University of Southern California
Los Angeles, CA 90089

ABSTRACT

Zinc finger genes in mammalian genomes are frequently found to occur in clusters with cluster members appearing in a tandem array on the chromosome. It has been suggested that *in situ* gene duplication events are primarily responsible for the evolution of such clusters. The problem of inferring the series of duplication events responsible for producing clustered families is different from the standard phylogeny problem. In this paper we study this inference problem using a graph called *Duplication Model* that captures the series of duplication events while taking into account the observed order of the genes on the chromosome. We provide algorithms to reconstruct a duplication model for a given data set. We use our method to hypothesise the series of duplication events that may have produced the *ZNF45* family that appears on human chromosome 19.

1. Introduction

Zinc finger (ZNF) genes code for proteins that contain one or more zinc finger motifs. The zinc finger motif is one of the most common motifs involved in nucleic acid-protein interaction. This motif was first discovered in the transcription factor TFIIIA in *Xenopus* [10]. Since then, the database of ZNF genes has been increasing rapidly. Of the various zinc finger motif types, the most commonly occuring type is the C2H2 type. In fact, it is estimated that mammalian genomes contain between 600 and 1000 genes that code for C2H2 type zinc finger proteins [6, 16]. Experimental studies on functions of these ZNF genes suggest that many of them code for transcription factors, and some of them are known to take part in cellular growth and development [5]. However, the biological functions of most of these ZNF genes are currently unknown.

The various sequencing projects reveal an interesting picture regarding the organization of these ZNF genes in mammalian genomes. In mammals, ZNF genes tend to occur as clusters (families) scattered throughout the various chromosomes. For instance, studies [2, 15] on human chromosome 19 (H19) indicate that there are at least 100 ZNF genes on this chromosome; in addition, these genes are found to occur in clusters in five major sites. One of the most striking features of the organization of these families is that the family members appear in a tandem array on the chromosome.

Most studies on zinc fingers have focussed on the three dimensional structure of fingers and on the biochemistry of DNA recognition by zinc fingers. However, less attention has been paid to the organization and evolution of ZNF gene families. In this paper we undertake a study of the evolution of ZNF gene families whose members occur in a tandem array on the same chromosome. Evolutionary studies can shed light on the mechanisms of evolution of such gene families. Also, evolutionary studies in conjunction with other analyses (for instance, analysis of upstream sequences) can yield valuable insights into the regulatory mechanisms controlling these genes and thus provide clues to their functions.

2. Representing evolutionary history using a Duplication Model

We introduce the evolutionary inference problem by using the *ZNF45* gene family as an example; the organization of the members of this family is typical of several known zinc finger gene families. The *ZNF45* gene family is found in the q13.2 gene cluster on H19 chromosome [16]. This family is estimated to have between 15 and 20 members. Each member in this family is a KRAB-ZNF gene; that is, each member has a Kruppel-associated box (KRAB) at the head and zinc fingers at the tail. The head is separated from the tail by a spacer element. The KRAB is a non-ZNF sequence that is known to take part in transcriptional repression [9] and it is highly conserved in each member of the family. Every zinc finger is C2H2 type and has 28 amino acids. A typical C2H2 type finger sequence is of the form $CysX_2CysX_3PheX_5LeuX_2HisX_3HisX_7$ (with X being

*Current e-mail: metang@sbh.com
†Current e-mail: msw@hto.usc.edu
‡Current e-mail: shibu.yooseph@celera.com

RECOMB 2001, Montreal, Canada

any amino acid) [16, 5]. The members of the *ZNF45* family have differing number of zinc fingers. The members are approximately evenly spaced on the chromosome and are arranged in a tandem array.

The organization and features of the members of the *ZNF45* gene family suggest that the members in the family may have been produced by a series of *in situ* tandem gene duplication events [12, 16]. The duplication process operates on a unit of one or more genes. It copies the unit and places the copy adjacent to the original. A mechanism for such a process could be unequal crossing-over during meiosis or mitosis that occurs in the germ line [8]. In our model of evolution we assume that transposition events do not occur; that is, a gene cannot be excised and reinserted in another region in the chromosome.

Recent work [3] notes that inferring evolutionary relationships of sequences evolving through tandem duplication is different from the traditional phylogeny inference problem as there is the additional constraint imposed by the order of the sequences in the genome. We argue similarly and introduce the notion of a *Duplication Model* - a graph that can be used to capture the series of duplication events and the evolutionary relationships while taking into account the observed left-to-right order of the genes on the chromosome.

A duplication model is a directed graph that has *nodes*, *edges*, and *blocks*. A node represents a gene, an edge between two nodes represents parent-child relationship, and a block represents a gene duplication event. Certain edges in a duplication model are allowed to cross each other; this is described using an edge-crossing rule. Figure 1 shows an example of a duplication model.

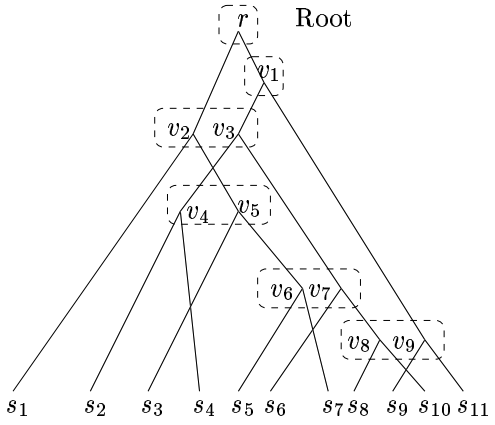

Figure 1: Example of a duplication model on a leaf set $\mathcal{L} = \{s_1, s_2, .., s_{11}\}$. The six blocks are represented by boxes; they are $B_1 = \langle r \rangle, B_2 = \langle v_1 \rangle, B_3 = \langle v_2, v_3 \rangle, B_4 = \langle v_4, v_5 \rangle, B_5 = \langle v_6, v_7 \rangle, B_6 = \langle v_8, v_9 \rangle$.

There is a special node designated the *root* node r that has only outgoing edges; the root represents the ancestral gene of the genes in our data set. An edge $e = (u, v)$ from node u to node v indicates that u is the parent of v. A node that has no outgoing edges is called a *leaf* node. The set of leaves is denoted by $\mathcal{L}$. The leaves of a duplication model represent our data set (that is, the genes that we observe on the chromosome). The left-to-right order of the leaf nodes (genes) defined by the duplication model is the same order in which these genes appear on the chromosome; for instance, in Figure 1, the left-to-right order is $s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}$, and s_{11}.

Every non-root node in a duplication model has exactly one incoming edge and every non-leaf node has exactly two outgoing edges. A node u is an ancestor of node v if u is on the path from r to v; in this case, v is also said to be a descendent of u. The direction of the edges in Figure 1 has been omitted with the understanding that the orientation is from top to bottom, i.e. a node appears below its ancestor in the graph.

A block consists of one or more non-leaf nodes. A block with one node represents a single gene duplication event and a block with at least two nodes represents a multigene duplication event. Every non-leaf node is part of exactly one block. No node in a block is the ancestor of another node in the same block. The following rule applies to the children of nodes in a block: let $B = \langle v_{i_1}, v_{i_2}, ..., v_{i_b} \rangle$ denote a block containing genes $v_{i_1}, v_{i_2}, ..., v_{i_b}$ in left-to-right order. Then, for each $1 \leq k < l \leq b$, the duplication event places v_k's left child to the left of v_l's left child and v_k's right child to the left of v_l's right child; this is an alternate way of stating that a duplication event places the copy of a block adjacent to the original.

Certain edges in a duplication model cross each other as described by the following edge-crossing rule. Let block $B = \langle v_{i_1}, v_{i_2}, ..., v_{i_b} \rangle$ and let $lc(v)$ and $rc(v)$ denote the left child and right child respectively, of a node v. Then, for every k, l, where $1 \leq k < l \leq b$, the edges $(v_k, rc(v_k))$ and $(v_l, lc(v_l))$ cross each other. No other edges in a duplication model cross each other.

If we represent only the parent-child relations defined by a duplication model DM, then the resulting structure T_{DM} is planar and can be drawn without the edges having to cross each other. This is apparent from the definition of nodes and edges in a duplication model. The graph T_{DM} is the rooted binary tree for the given set of genes and is said to be the *associated phylogeny* for DM. Clearly, the associated phylogeny of DM is unique. Figure 2 shows an example of an associated phylogeny of a duplication model. While the nodes in DM and T_{DM} are the same, the key difference between T_{DM} and DM is that T_{DM} does not contain blocks and thus cannot directly explain the duplication events that produced the genes we observe.

In summary, a duplication model captures three aspects of the evolutionary history of a gene family: the ancestral relationship, the duplication history, and the ordering of the members on the chromosome. For the remainder of the paper we assume that the set $\mathcal{L} = \{s_1, s_2, ..., s_n\}$

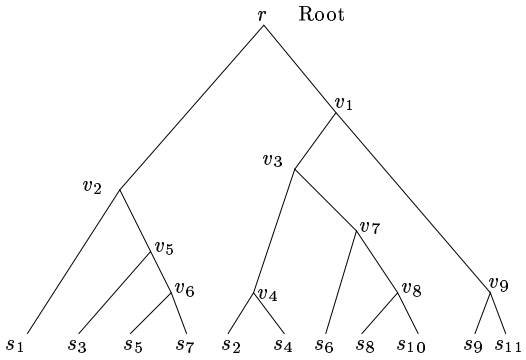

Figure 2: The associated phylogeny of the duplication model in Figure 1.

of genes under consideration appear as $s_1, s_2, .., s_n$ on the chromosome; we say that a duplication model is *consistent* with $\mathcal{L}$ if the left-to-right ordering of its leaves is $s_1, s_2, ..., s_n$.

The remainder of the paper is organized as follows. In Section 3 we define the cost of a duplication model and formulate the problem of inferring a duplication model of minimum cost. We provide algorithms to solve a restricted version of the inference problem. In Section 4 we provide a general method to construct a duplication model and discuss the performance of this method using simulation. In Section 5 we address the following question: given a phylogenetic tree T, does there exist a duplication model DM such that its associated phylogeny T_{DM} is equal to T? In Section 6 we apply the method described in Section 4 to the *ZNF45* family.

3. An inference problem

Let $\mathcal{L} = \{s_1, s_2, .., s_n\}$ be a set of genes where each gene s_i is a sequence of length k, with each position (or site) in the gene having a label from an alphabet Σ of size r. Thus, each $s_i \in \Sigma^k$. Let $Dist$ be a pairwise distance measure on sequences in Σ^k that satisfies the triangle inequality.

Let DM be a duplication model consistent with $\mathcal{L}$. Further, let $s(v) \in \Sigma^k$ denote the sequence at internal node v in DM. The *size* of a block in DM is the number of nodes in that block. We use $C(DM)$ to denote the cost of DM and define it as

$$C(DM) = \sum_{block\ b} C_d(b_i) + \sum_{edge\ e} C(e),$$

where $C_d(b_i)$ denotes the duplication cost for block b of size b_i and $C(e) = Dist(s(u), s(v))$ for edge $e = (u, v)$. For instance, $Dist$ can be the Hamming distance between the sequences.

Our discussions motivate the following inference problem.

Gene duplication problem -
Input : Set $\mathcal{L} = \{s_1, s_2, .., s_n\}$ of genes and integer b.
Output: A duplication model DM consistent with $\mathcal{L}$ having maximum block size b, and of minimum cost.

In the remainder of this section we consider the simplest version of the gene duplication problem - the *single* gene duplication problem. In this problem, the maximum block size $b = 1$; alternatively, $C_d(1) = 0$ and $C_d(b_i) = \infty, \forall b_i > 1$.

3.1 Single gene duplication. The single gene duplication problem is essentially a phylogeny inference problem but with the additional constraint that we seek a phylogeny whose planar representation (i.e. edges not allowed to cross) induces the same ordering on the genes as their ordering on the chromosome. We refer to such a tree as an *ordered* tree.

An *ordered tree* on a set $\mathcal{L} = \{s_1, s_2, ..., s_n\}$ is a rooted tree with n leaves in which each leaf has a unique sequence from $\mathcal{L}$ and such that the left-to-right ordering of the leaves is $s_1, s_2, ..., s_n$.

Single gene duplication problem - (Maximum Parsimony)
Input : Set $\mathcal{L} = \{s_1, s_2, .., s_n\}$ of genes, where each gene has length k.
Output: An ordered tree T (on $\mathcal{L}$) in which each internal node is labelled with a sequence of length k and such that the cost of T is minimum.

We conjecture the single gene duplication problem to be NP-complete. The single gene duplication problem has been addressed in [3] and an approximation algorithm with ratio 2 has been given. In this paper we provide a dynamic programming framework that provides algorithms whose quality (of solution) and running time complexity can be controlled by choice of different sequence labels at the internal nodes. Using this framework we provide an exact algorithm and then an approximation algorithm with ratio 2 (this algorithm is different from the one in [3]). We then modify this approach to provide a polynomial time approximation scheme for the problem.

Algorithms: We describe a dynamic programming framework for solving the single gene duplication problem.

For $1 \leq i \leq j \leq n$, we use $[i, j]$ to denote the gene set $\{s_i, s_{i+1}, ..., s_j\}$, whose elements appear in consecutive order on the chromosome. We say that an ordered tree T *spans* $[i, j]$ if T has leaf set $\{s_i, s_{i+1}, .., s_j\}$ and the left-to-right order of the leaves in T is $s_i, s_{i+1}, .., s_j$.

Consider an ordered tree T that spans $[i, j]$. This tree is obtained by combining ordered subtrees T_1 and

T_2 that span $[i,m]$ and $[m+1,j]$ respectively, for some $i \leq m < j$. Let $C([i,j],s,m)$ denote the minimum cost among all ordered trees T that span $[i,j]$ and have root labelled with $s \in \Sigma^k$, and such that T's left subtree spans $[i,m]$ and T's right subtree spans $[m+1,j]$. Let $C([i,j],s)$ denote the minimum cost among all ordered trees that span $[i,j]$ and have root labelled with s. Let $C([i,j])$ denote the minimum cost among all ordered trees that span $[i,j]$. Then, we have the following recurrence relations:

$$C([i,j],s,m) = \min_{s^* \in \mathcal{S}_1}[C([i,m],s^*) + Dist(s,s^*)] + \min_{s^{**} \in \mathcal{S}_2}[C([m+1,j],s^{**}) + Dist(s,s^{**})] \quad (3.1)$$

$$C([i,j],s) = \min_{i \leq m < j} C([i,j],s,m) \quad (3.2)$$

$$C([i,j]) = \min_{s \in \mathcal{S}} C([i,j],s) \quad (3.3)$$

where $Dist$ is the pairwise distance measure on sequences in Σ^k, and $\mathcal{S}, \mathcal{S}_1$, and $\mathcal{S}_2$ are subsets of Σ^k. The dynamic programming algorithm works by considering each subproblem $[i,j]$ in the order of increasing cardinality (note that the cardinality of a subproblem $[i,j]$ is $j-i+1$). The base cases are $C([i,i],s_i) = 0, \forall 1 \leq i \leq n$, and $C([i,i],s_j) = \infty, s_j \neq s_i$. The minimum cost ordered tree can be recovered by keeping track of its subtrees.

The framework we discussed defines algorithms for which the quality (of solution) and running time can be controlled by the choice of $\mathcal{S}, \mathcal{S}_1$, and $\mathcal{S}_2$. If we set $\mathcal{S} = \mathcal{S}_1 = \mathcal{S}_2 = \Sigma^k$, then we are allowing all possible choices of sequence labels. It is clear that the solution returned by the dynamic programming algorithm is optimal. The running time of the algorithm is computed as follows: all pairwise distances of the sequences in Σ^k can be computed in $O(r^{2k}k)$, where $r = |\Sigma|$; there are $O(n^2)$ subproblems of the form $[i,j]$ and each subproblem $[i,j]$ can be solved in $O(r^{2k}n)$. Thus the total running time is $O(r^{2k}(k+n^3))$.

LEMMA 3.1. *The single gene duplication problem can be solved exactly in $O(r^{2k}(k+n^3))$.*

We obtain a 2-approximation algorithm by combining the dynamic programming described above with the technique of *lifting* [20]. Consider a rooted tree T in which node v is labelled with sequence $s(v) \in \Sigma^k$. *Lifting* is a process that replaces the sequence at a node v with a sequence of one of the leaves in the subtree rooted at v. Lifting operates in a bottom-up fashion, i.e. a node is lifted only after all its children have been lifted. Suppose we are at a node v (labelled with sequence $s(v)$) and we have already lifted its children v_1 and v_2, whose *new* sequences are denoted by s_{i_1} and s_{i_2} respectively (where $s_{i_1} \in \mathcal{L}$ is in the subtree rooted at v_1 and $s_{i_2} \in \mathcal{L}$ is in the subtree rooted at v_2). Then, to lift v, we replace $s(v)$ with s_{i_1} if $Dist(s(v), s_{i_1}) \leq Dist(s(v), s_{i_2})$; otherwise, we replace $s(v)$ with s_{i_2}.

LEMMA 3.2. *Let $Dist$ be a pairwise distance measure that satisfies the triangle inequality. For a tree $T = (V,E)$, let cost of T be given by*

$$C(T) = \sum_{(u,v) \in E} Dist(s(u), s(v))$$

Let T^* be a most parsimonious ordered tree for the single gene duplication problem and let T^L be the tree obtained from T^* by the lifting process described above. Then $C(T^L) \leq 2C(T^*)$.

The proof of the above lemma uses the results from [20, 19]. The above lemma shows the existence of an ordered tree T^L that is lifted and whose cost is at most twice the optimal cost.

Recall our dynamic programming framework. If we set $\mathcal{S} = [i,j]$, $\mathcal{S}_1 = [i,m]$, and $\mathcal{S}_2 = [m+1,j]$, then the algorithm returns an ordered tree T that is lifted and has minimum cost among all lifted ordered trees. From the above lemma, we have $C(T) \leq C(T^L) \leq 2C(T^*)$. Since lifting considers only labels from $\mathcal{L}$, the total running time is $O(n^2(k+n^3))$.

LEMMA 3.3. *The single gene duplication problem has a 2-approximation algorithm with running time $O(n^2(k+n^3))$.*

The lifting technique can be combined with local optimization to provide a Polynomial Time Approximation Scheme (PTAS) for the single gene duplication problem. A PTAS is an algorithm which, for every $\epsilon > 0$, returns a solution whose cost is at most $(1+\epsilon)$ times the cost of the optimal solution, and runs in time bounded by a polynomial (depending on ϵ) in the input size. The key lemma for designing the PTAS is based on the results on the problem of tree alignment with a given phylogeny [20, 19]. We leave the details of the PTAS to the full version of the paper.

LEMMA 3.4. *The single gene duplication problem has a PTAS.*

4. Window method

In this section we describe a general distance based method for reconstructing a duplication model for a given set of genes. This method is analogous to sequential clustering methods like UPGMA [11, 17] and Neighbor-Joining [14]. Our method, called the *Window Method*, detects both single gene and multigene duplication events.

The window method works with a *list* $\mathcal{L}$ of genes. Initially, list $\mathcal{L} = (s_1, s_2, ..., s_n)$, that is, the list of all the genes in the order they appear on the chromosome. The window method operates on sublists of $\mathcal{L}$ called *windows*. It identifies a pair of adjacent non-overlapping windows (of same length) that are *closest* to each other. List $\mathcal{L}$ is then modified by merging these two windows. The process of identifying closest adjacent windows and

merging them is repeated till the list $\mathcal{L}$ has a single gene. The duplication model is reconstructed by recording the sequence of window merges. We now discuss the details.

Let $\mathcal{L} = (a_1, a_2, ..., a_p)$ be the list of genes at the start of the current iteration. A *window* W of length l is a sublist $(a_i, a_{i+1}, ..., a_{i+l-1})$ of l consecutive genes in $\mathcal{L}$. Let $\mathcal{D}(W_1, W_2)$ denote the distance between two adjacent non-overlapping windows W_1 and W_2 of same length l, where $W_1 = (a_i, a_{i+1}, ..., a_{i+l-1})$ and $W_2 = (a_{i+l}, a_{i+l+1}, ..., a_{i+2l-1})$. Among all adjacent non-overlapping windows of the same length, the window pair (W_1, W_2) is said to be a *closest pair* if $\mathcal{D}(W_1, W_2)$ is minimum. The merging step in the window method involves identifying a closest pair (W_1, W_2) and merging them to give a new window $(a'_i, a'_{i+1}, ..., a'_{i+l-1})$. The list $\mathcal{L}$ is then modified resulting in $\mathcal{L} = (a_1, a_2, ..., a_{i-1}, a'_i, a'_{i+1}, ..., a'_{i+l-1}, a_{i+2l}, ..., a_p)$ at the end of the iteration.

Several definitions for $\mathcal{D}$, the distance measure on windows, are possible. Recall that $Dist$ denotes the pairwise distance measure between genes. We implemented the window method with the following definitions.

$$\mathcal{D}(W_1, W_2) = \frac{\sum_{j=i}^{j=i+l-1} Dist(a_j, a_{j+l})}{l},$$

where $W_1 = (a_i, a_{i+1}, ..., a_{i+l-1})$ and $W_2 = (a_{i+l}, a_{i+l+1}, ..., a_{i+2l-1})$. The pairwise distances between the genes in $\mathcal{L}$ are updated using the following equations:

$$Dist(a'_r, a_q) = \frac{Dist(a_r, a_q) + Dist(a_{r+l}, a_q)}{2},$$
where $i \le r \le i + l - 1$, and
either $1 \le q \le i - 1$ or $i + 2l \le q \le p$

$$Dist(a'_r, a'_q) = \frac{Dist(a_r, a_q) + Dist(a_r, a_{q+l})}{4} + \frac{Dist(a_{r+l}, a_q) + Dist(a_{r+1}, a_{q+l})}{4},$$
where $i \le r < q \le i + l - 1$

The running time of the window method is $O(n^4)$.

4.1 Simulation Results. We assess the performance of the window method using simulations. In each run of the simulation, a duplication model and the gene sequences are generated for the corresponding values of the parameters. These sequences are used to compute a distance matrix which is given as input to the window method. The performance of the window method on the run is measured by the *exact* recovery of the original duplication model. The parameters are k - the length of each gene sequence, p - the mutation rate, and f - the number of generations.

The duplication model and the gene sequences are generated as follows. A list gen_i is used to keep track of all genes at generation i as they appear on the chromosome. List gen_1, corresponding to the first generation, has a single gene of length k where each position in the gene is one of the four nucleotides A, C, T, G. Suppose gen_i has r_1 genes. Then gen_{i+1} is obtained from gen_i by selecting a random sublist (say, of length r_2), duplicating this sublist and placing the copy next to it in the list; gen_{i+1} thus has $r_1 + r_2$ genes, each of length k. Finally, for each gene, the nucleotide at each site is allowed to mutate to a different nucleotide, independently and identically, with probability p. The gene sequences are allowed to evolve in this manner for f generations. The duplication model is obtained by keeping track of the gen_i's.

The above procedure was used to generate duplication models and sequence data for p from 0.01 to 0.5 (increments of 0.05), k from 100 to 800 (increments of 100), and $f = 6, 10$. For each combination of the parameter values, 200 data sets were generated. We note that $f = 6$ resulted in an average of 25 genes in a data set and $f = 10$ resulted in an average of 55 genes in a data set. From each data set, the aligned gene sequences were used to create a distance matrix (using hamming distances) and the window method was used to infer a duplication model from this distance matrix. The duplication model was compared with the original. The performance is measured by the fraction of runs for which the inferred duplication model is equal to the original duplication model. Figure 3 reports two graphs, one with fixed sequence length of 600 base pairs (with varying mutation rates and number of generations) and the other with fixed mutation rate of 0.05 (with varying sequence length and number of generations).

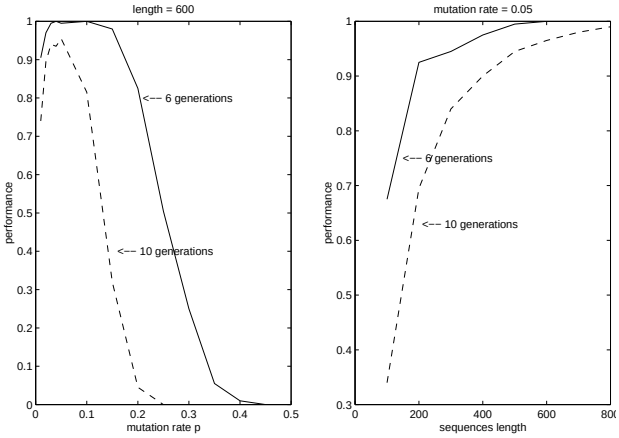

Figure 3: Performance of the window method.

The graphs in Figure 3 show the effects of mutation rate, sequence length and number of generations on the performance of the window method. For a fixed sequence length, the performance of the window method is good at low and moderate ends of the mutation rate scale; however, the performance drops off as the mutation rate increases. Also, for a fixed mutation rate, the performance of the window method improves with an increase in the sequence length. The two graphs also show that the performance of the window method degrades with increasing f. Similar observations have been made in the context of phylogeny

inference from simulated data [7, 18].

The simulation study reported here is a preliminary one. Several modifications can be made to improve the performance of the window method. These include using an alternative distance update formula (from the one discussed in Section 4) and using appropriate distance correction [18] to account for the assumptions of the evolutionary model; we used uncorrected distances in our study. Finally, we note that in our study, performance is measured by the *exact* recovery of the original duplication model. It is also possible to measure the *partial* recovery of the original duplication model (in terms of number of edges and blocks that are recovered); this may provide a better insight into the performance of the window method at high mutation rates.

5. Reconstructing a duplication model from a phylogeny

From our discussions in Section 2 recall that every duplication model DM has a unique associated phylogeny T_{DM}. In this section we consider the inverse question, namely, given a phylogeny T, does there exist a duplication model DM such that $T = T_{DM}$?

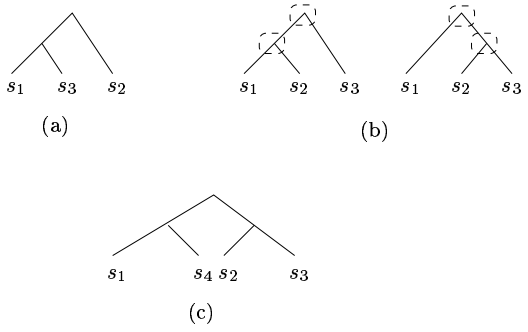

Figure 4: Counter example.

Figure 4(a) shows a phylogeny T on leaf set $\mathcal{L}_1 = \{s_1, s_2, s_3\}$ that is not the associated phylogeny of any duplication model. This is easily seen by considering all possible duplication models (they are shown in Figure 4(b)) that are consistent with $\mathcal{L}_1$; none of the duplication models have an associated tree equal to T. Figure 4(c) shows a phylogeny on leaf set $\mathcal{L}_2 = \{s_1, s_2, s_3, s_4\}$ that is not the associated phylogeny of any duplication model. These examples imply that given a phylogeny T, a duplication model which has T as its associated phylogeny need not exist.

In the remainder of this section we show that if a duplication model DM exists for a given phylogeny T such that $T = T_{DM}$, then DM is unique. This is done by using an encoding scheme that uniquely defines a duplication model. We conclude this section by providing an algorithm to construct the duplication model (when it exists) from a given phylogeny.

Definitions: Let DM be a duplication model consistent with $\mathcal{L} = \{s_1, s_2, ..., s_n\}$. As before we use $lc(v)$ to denote the left child of v and $rc(v)$ to denote the right child of v. A leaf in DM that is labelled with s_i is said to have *index* i. The height of a node v in DM is defined to be the number of edges in the path from v to root r. The height of a block B is defined inductively as follows: the height of the block containing root r is 0; if $H(parent(v))$ denotes the height of the block containing the parent of node v, then height of block $B = \max_v\{H(parent(v))|v \text{ is in block } B\} + 1$. We use MAX_HEIGHT to denote the maximum height of a block in DM. For instance, the heights of the blocks in Figure 2 are as follows:
Height of $B_1 = 0$, height of $B_2 = 1$, height of $B_3 = 2$, height of $B_4 = 3$, height of $B_5 = 4$, and height of $B_6 = 5$.

Encoding of nodes in DM: We define an encoding of DM, denoted by $Enc(DM)$, in which each node v is represented by a pair (L_v, R_v) of leaf indices; this pair is called the LR pair of v. The LR pair is defined as follows:1. The LR pair of a leaf with index i is (i, i); 2. The LR pair of a node v is $(L_v, R_v) = (L_{lc(v)}, R_{rc(v)})$, where $(L_{lc(v)}, R_{lc(v)})$ and $(L_{rc(v)}, R_{rc(v)})$ are the LR pairs of v's left and right children respectively. We define $Enc(DM) = \{(L_v, R_v)|v \in DM\}$.

We describe a property of $Enc(DM)$ that is useful for our purposes. To describe this property, we introduce the concept of a generation list gen_h - a list containing nodes of a certain height in DM. A gen_h list is defined inductively, for $0 \leq h < MAX\_HEIGHT + 1$, as follows: $gen_0 = (r)$; gen_{h+1} is obtained from gen_h by the following operation - every sublist $\langle v_{p_1}, v_{p_2}, .., v_{p_b}\rangle$ in gen_h that corresponds to a block of height i is replaced with the sublist $\langle lc(v_{p_1}), lc(v_{p_2})..,lc(v_{p_b}), rc(v_{p_1}), rc(v_{p_2})..,rc(v_{p_b})\rangle$.

For instance, the generation lists for the duplication model in Figure 1 are:
$gen_0 = (r), gen_1 = (v_2, v_1), gen_2 = (v_2, v_3, v_9)$,
$gen_3 = (s_1, v_4, v_5, v_7, v_9)$,
$gen_4 = (s_1, s_2, s_3, s_4, v_6, v_7, v_9)$,
$gen_5 = (s_1, s_2, s_3, s_4, s_5, s_7, v_8, v_9)$,
$gen_6 = (s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11})$.

The proofs of the lemmas in this section will be given in the full version of the paper.

LEMMA 5.1. *Let DM be a duplication model consistent with* $\mathcal{L} = \{s_1, s_2, .., s_n\}$. *Let* $Enc(DM)$ *denote the set of LR pairs of all the nodes in DM. For any* $0 \leq h \leq (MAX\_HEIGHT+1)$, *let* $gen_h = (v_{i_1}, v_{i_2}, ..., v_{i_k})$. *Then* $L_{v_{i_1}} < L_{v_{i_2}} < .. < L_{v_{i_k}}$ and $R_{v_{i_1}} < R_{v_{i_2}} < .. < R_{v_{i_k}}$

COROLLARY 5.1. *Let DM be a duplication model consistent with* $\mathcal{L} = \{s_1, s_2, .., s_n\}$. *Then*
1. No two nodes in DM have the same LR pair.
2. Let v be a non-leaf node in DM. Then $L_v = L_{lc(v)} < L_{rc(v)}$ and $R_{lc(v)} < R_v = R_{rc(v)}$

3. For a non-leaf node v with LR pair (L_v, R_v), $L_v < R_v$.
4. For any node v with LR pair (L_v, R_v), the following is true: L_v is the smallest index of a leaf that is a descendent of v and R_v is the largest index of a leaf that is a descendent of v.
5. Let $B = \langle v_{p_1}, v_{p_2}, .., v_{p_b} \rangle$ be a block in DM. Then, $L_{v_{p_1}} < L_{v_{p_2}} < .. < L_{v_{p_b}} < R_{v_{p_1}} < R_{v_{p_2}} < .. < R_{v_{p_b}}$
6. Let the LR pairs for nodes v and u be (L_v, R_v) and (L_u, R_u) respectively.
(a) Let $L_u = L_v$. Then $R_u < R_v$ iff u is a descendent of v.
(b) Let $R_u = R_v$. Then $L_u < L_v$ iff u is a descendent of v.

Encoding of nodes in T_{DM}: Consider the associated phylogeny T_{DM} of DM. Both DM and T_{DM} define the same evolutionary relationships on the elements in $\mathcal{L}$; that is, both of them induce the same topology on any subset of elements in $\mathcal{L}$. Therefore, we can define a similar encoding for nodes in T_{DM}. For node $v \in T_{DM}$, let S_v and G_v denote the smallest and largest leaf indices respectively, in the subtree rooted at v. We define $Enc(T_{DM}) = \{(S_v, G_v) | v \in T_{DM}\}$.

LEMMA 5.2. *Let DM be a duplication model consistent with $\mathcal{L} = \{s_1, s_2, .., s_n\}$ and let T_{DM} be the associated phylogeny of DM. Then $Enc(DM) = Enc(T_{DM})$.*

LEMMA 5.3. *Let DM_1 and DM_2 be duplication models consistent with $\mathcal{L} = \{s_1, s_2, ..., s_n\}$, and let T_{DM_1} and T_{DM_2} be their associated phylogenies respectively. If $Enc(DM_1) = Enc(DM_2)$ then $T_{DM_1} = T_{DM_2}$.*

We now show that the encoding scheme Enc defined above is unique; that is, different duplication models cannot have the same encoding.

LEMMA 5.4. *Let DM_1 and DM_2 be duplication models consistent with $\mathcal{L} = \{s_1, s_2, ..., s_n\}$. The following statement is true: $Enc(DM_1) = Enc(DM_2)$ iff $DM_1 = DM_2$.*

Next we show that if two associated phylogenies are equal then their corresponding duplication models are also equal.

LEMMA 5.5. *Let DM_1 and DM_2 be duplication models consistent with $\mathcal{L} = \{s_1, s_2, ..., s_n\}$. Let T_{DM_1} and T_{DM_2} be the associated phylogenies for DM_1 and DM_2 respectively. If $T_{DM_1} = T_{DM_2}$, then $DM_1 = DM_2$.*

We conclude this section by providing an algorithm for reconstructing DM from a given phylogeny T. If DM exists then, from Lemma 5.1, the statement about the generation list gen_h will be true for every $0 \leq h \leq (MAX\_HEIGHT+1)$. Thus, we reconstruct DM by doing a top-down traversal of the nodes in T. Two minor points have to be addressed here. We assume that $|Enc(T)|$ is equal to the number of nodes in T as otherwise, from Lemma 5.2 and Part (1) of Corollary 5.1, we conclude that the duplication model does not exist. Also, every node in T can be preprocessed to identify its left child and right child. If v_1 and v_2 are the children of v, then v_1 is v's left child if $S_v = S_{v_1} < S_{v_2}$ and $G_{v_1} < G_v = G_{v_2}$; if for any node, this test fails, then from Part (2) of Corollary 5.1 we conclude that the duplication model does not exist.

The algorithm starts at the root and at each iteration identifies blocks and replaces the nodes in the blocks with their children in the appropriate order. This is accomplished by using list gen. A sublist $\langle v_{i_j}, v_{i_{j+1}}, .., v_{i_{j+b}} \rangle$ of gen is identified as a block only if the condition in Part (5) of Corollary 5.1 holds for this sublist and Lemma 5.1 holds for the resulting modification of gen. If at any stage when $gen \neq (s_1, s_2, .., s_n)$ and it is not possible to identify a block, then the algorithm terminates and outputs that the duplication model does not exist. The correctness of the algorithm follows from Lemma 5.1. The running time of the algorithm is $O(n^2)$.

6. Analysis of ZNF45 family

The amino acid sequences for 16 members of the *ZNF45* family were aligned using the Multiple Sequence Alignment program available through the Sequence Analysis Server at Michigan Tech (*http://genome.cs.mtu.edu/map/map.html*). The parameters used in the alignment were the following - amino acid substitution matrix: blosum62, gap open penalty: 30, gap extension penalty: 2. The alignment was used to produce a matrix of pairwise distances between the members. This distance matrix was given as input to the window method described in Section 4. The resulting duplication model for the 16 members of the *ZNF45* family is shown in Figure 5. This duplication model hypothesises that the *ZNF45* family has evolved mainly through single gene duplication events.

The members of the *ZNF45* family have different number of zinc fingers. *In situ* tandem duplication events, involving zinc fingers of the *same* gene, are likely to have given rise to this feature. Studies [13, 4, 5] have shown that residues in certain key positions in a C2H2 type finger bind to DNA. We note that the zinc fingers of the *ZNF45* family are quite diverse in the residues in these key positions. This suggests a diversity in the DNA that the transcription factors encoded by these genes bind to. As noted in [16], the high degree of conservation of the KRAB domains coupled with the diversity in the zinc fingers of the *ZNF45* family members suggests that they may regulate different genes in a similar manner during development.

It has been suggested that families like *ZNF45*, that have members in a tandem array with even spacing between the members, are gene batteries that have evolved in such a way that the family members are co-ordinately expressed at various stages of development [16, 1]. An

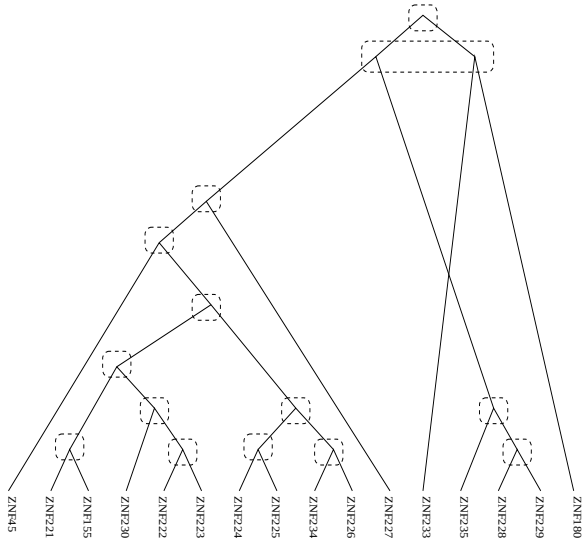

Figure 5: The duplication model for the *ZNF45* family. The left-to-right order of the above genes is the same as their ordering on the chromosome.

analysis of the upstream sequences for these genes could shed further light on this hypothesis.

7. Conclusion

In this paper we described a graph-theoretic structure called a duplication model to study the evolutionary relationships of members of gene families that have evolved through *in situ* gene duplication events. There are several extensions to this work. Statistical inference techniques (like maximum likelihood estimation) can be incorporated to construct duplication models. Another extension is to consider gene deletion events (which we do not presently consider) in the evolutionary model.

Acknowledgements We are very grateful to Elbert Branscomb and Lisa Stubbs at the DOE Joint Genome Institute for providing us with the sequences for the members of the *ZNF45* family. S.Y. acknowledges support by a Fellowship from the Program in Mathematics and Molecular Biology at the Florida State University, with funding from the National Science Foundation under Grant No. DMS-9406348.

References

[1] M. Arnone and E. Davidson. The hardwiring of development: organization and function of genomic regulatory systems. *Development*, 124:1851–1864, 1997.

[2] L. Ashworth, M. Batzer, B. Brandriff, E. Branscomb, P. deJong, E. Garcia, J. Garnes, L. Gordon, J. Lamerdin, G. Lennon, H. Mohrenweiser, A. Olsen, T. Slezak, and A. Cerrano. An integrated, metric physical map of human chromosome 19. *Nature Genetics*, 11:422–427, 1995.

[3] G. Benson and L. Dong. Reconstructing the Duplication History of a Tandem Repeat. In *Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 44–53, 1999.

[4] J. Desjarlais and J. Berg. Towards rules relating zinc finger protein sequences and DNA binding site preferences. *Proceedings of the National Academy of Sciences, USA*, 89:7345–7349, 1992.

[5] T. El-Baradi and T. Pieler. Zinc finger proteins: what we know and what we would like to know. *Mechanisms of Development*, 35:155–169, 1991.

[6] J. Hoovers, M. Mannens, R. John, J. Bliek, V. van Heyningen, D. Porteous, N. Leschot, A. Westerveld, and P. Little. High-resolution localization of 69 potential human zinc finger protein genes: A number are clustered. *Genomics*, 12:254–263, 1992.

[7] D. Huson, S. Nettles, K. Rice, T. Warnow, and S. Yooseph. Hybrid Tree Reconstruction Methods. *ACM Journal of Experimental Algorithmics*, 4, 1999.

[8] W. Li and D. Graur. *Fundamentals of molecular evolution.* Sinauer Associates, Inc., 1991.

[9] J. Margolin, J. Friedman, W. Meyer, H. Vissing, H-J. Thiesen, and F. Rauscher III. Kruppel-associated boxes are potent transcriptional repression domains. In *Proceedings of the National Academy of Sciences, USA*, volume 91, pages 4509–4513, 1994.

[10] J. Miller, A. McLachlan, and A. Klug. Repetitive zinc-binding domains in the protein transcription factor IIIA from *Xenopus* oocytes. *EMBO J*, 4:1609–1614, 1985.

[11] M. Nei. *Molecular Population Genetics and Evolution.* North-Holland, Amsterdam, 1975.

[12] S. Ohno. *Evolution by Gene Duplication.* Springer-Verlag, Berlin, New York, 1970.

[13] N. Pavletich and C. Pabo. Zinc Finger-DNA Recognition: Crystal Structure of a Zif2680-DNA Complex at 2.1 Å. *Science*, 252:809–817, 1991.

[14] N. Saitou and M. Nei. The neighbor-joining method. *Mol. Biol. Evol.*, 4:406–425, 1987.

[15] M. Shannon, L. Ashworth, M. Mucenski, J. Lamerdin, E. Branscomb, and L. Stubbs. Comparative analysis of a conserved zinc-finger gene cluster on human chromosome 19q and mouse chromosome 7. *Genomics*, 33:112–120, 1996.

[16] M. Shannon, J. Kim, L. Ashworth, E. Branscomb, and L. Stubbs. Tandem Zinc-Finger Gene Families in Mammals: Insight and Unanswered Questions. *DNA Sequence - The Journal of Sequencing and Mapping*, 8(5):303–315, 1998.

[17] R. Sokal and C. Michener. A statistical method for evaluating systematic relationships. *Univ. Kansas Sci. Bull.*, 28:1409–1438, 1958.

[18] D. Swofford, G. Olsen, P. Waddell, and D. Hillis. *Molecular Systematics (ed. D. Hillis and C. Moritz and B. Mable)*, chapter Phylogenetic Inference. Sinauer Associates, 1996.

[19] L. Wang and D. Gusfield. Improved Approximation Algorithms for Tree Alignment. *Journal of Algorithms*, 25(2):255–273, 1997.

[20] L. Wang, T. Jiang, and E. L. Lawler. Approximation Algorithms for Tree Alignment with a Given Phylogeny. *Algorithmica*, 16(3):302–315, 1996.