CSCI 2950-C: Algorithms for Cancer Genomics **Spring 2015**

Lecture 14: April 21, 2015

Lecturer: Mohammed El-Kebir *Scribe:*

Note: *These notes may not accurately reflect what was said in class, and may have typos/omissions. If you discover any mistakes or inaccuracies, please bring them to the instructor's attention.*

1 Multi-state perfect phylogeny

Definition 1. *A* perfect phylogeny *for M is a tree T with n leaves such that:*

1. *Each taxon labels exactly one leaf;*
2. *Each node $v \in V(T)$ is labeled by $\{0, \ldots, k-1\}^m$;*
3. *Nodes labeled with state $i \in \{0, \ldots, k-1\}$ for character c form a connected subtree $T_c(i)$.*

In case $k = 2$ and assuming an all-zero root node, we have the following theorem.

Theorem 1 (Perfect phylogeny theorem)**.** *Matrix $M \in \{0,1\}^{n \times m}$ has a perfect phylogeny if and only if no pair of columns c, d conflicts, i.e. contains binary pairs $(0,1)$; $(1,0)$; and $(1,1)$.*

For general k we have the following hardness result.

Theorem 2 (Bodlaender 1992)**.** *The multi-state perfect phylogeny problem is NP-complete.*

1.1 Cladistic characters

A *cladistic* character c is defined by a tree S_c whose node set is given by $V(S_c) = \{s_0, \ldots, s_{k-1}\}$.

Definition 2. *The* reduced *tree R_c of perfect phylogeny T with respect to character c has vertex set $V(R_c)$ and edge set $E(R_c)$ where*

- *$V(R_c) = \{X_0, \ldots, X_{k-1}\}$ such that $X_i = V(T_c(i))$,*
- *$(X_i, X_j) \in E(R_c)$ iff $i \neq j$ and there exists $u \in X_i$ and $v \in X_j$ such that $(u, v) \in E(T)$.*

Definition 3. *A perfect phylogeny T is* consistent *with cladisitic character c provided that $(s_i, s_j) \in E(S_c)$ if and only if $(X_i, X_j) \in E(R_c)$.*

We say that a perfect phylogeny T is *consistent* if it is consistent with all its cladistic characters.

Definition 4. *The* cladistic expansion function *$h : \{1, \ldots, m\} \times \{0, \ldots, k-1\} \to \{0,1\}^k$ is defined as $h(c, p) = \mathbf{x}^T$ where*

$$x_l = \begin{cases} 1, & \text{if } l \text{ is a descendant of } p, \\ 0, & \text{otherwise.} \end{cases}$$

for all $0 \leq l < k$.

Definition 5. *Given a matrix* $M = [a_{ij}] \in \{0, \ldots, k-1\}^{n \times m}$*, its* cladistic expansion M' *is a* $n \times km$ *binary matrix defined as*

$$\begin{pmatrix} h(1, a_{1,1}) & \ldots & h(n, a_{1,m}) \\ \vdots & \ddots & \vdots \\ h(1, a_{n,1}) & \ldots & h(n, a_{n,m}) \end{pmatrix}.$$

Note that we can go from $M \leftrightarrow M'$. Also, by Theorem 1 we have $M' \leftrightarrow T'$. We now define $T \leftrightarrow T'$.

Lemma 1. *Let* $M \in \{0, \ldots, k-1\}^{n \times m}$*.* M *admits a consistent perfect phylogeny if and only if* M' *is conflict-free.*

Proof. ($\Leftarrow$) Let T' be the perfect phylogeny corresponding to M'. Obtain T from T'. We claim that T is a consistent perfect phylogeny for M.

1+2. By definition of T (and the transformation).

3. Consider cladistic character c and state p. Since T' is a perfect phylogeny, T' has exactly one edge labeled by (c, p). Therefore all descendants of this edge whose immediate ancestor for c is labeled by (c, p) form a subtree.

4. By definition of M'.

($\Rightarrow$) Let T be a consistent perfect phylogeny on M. Obtain T' from T. We claim that T' is a perfect phylogeny (on M').

1+2. By definition of T' (and the transformation).

3. Suppose for a contradiction, that binary character d does not induce a *connected* subtree $T'_d(1)$.[1] Let (c, p) be the corresponding cladistic character state pair.

   Let $u, v \in T'_d(1)$ be two distinct vertices whose (unique) outgoing arcs have target vertices that are not in $T'_d(1)$. Let s and t be the states of u and v, respectively. Since T is a perfect phylogeny, we have that $s \neq t$. Therefore we can assume w.l.o.g. that $s \neq p$. Hence, $p < s$.

   Let w be the unique parent of v and let q be its state for character c. Note that w has state 0 for binary character d. Thus, we have that $q < s$. Since $p < s$, w is the parent of v and S_c is a tree, we have that $p < q < s$. The transformation however would have then resulted in a 1 for binary character d (recall, it corresponds to state p for character c). This is a contradiction.

□

[1] Note that we can use state 1 without loss of generality, as $T'_d(0)$ is the complement of $T'_d(1)$.