CS 466 – Introduction to Bioinformatics Lecture 17

Mohammed El-Kebir

October 28, 2020

Document history:

- 10/31/2018: Initial version
- 10/23/2019: Minor changes
- 10/23/2020: Fixed proof of Lemma 1, updated Observation 2.
- 10/28/2020: Minor corrections.

Contents

1 Two-state Perfect Phylogeny Problem

These notes are based on Ref. [1].

We are given a binary matrix $B \in \{0,1\}^{n \times m}$ with n taxa (think of them as species) and m characters. We say that taxon $f \in [n]$ *possesses* character $c \in [m]$ if $b_{f,c} = 1$. We consider the large maximum parsimony phylogeny problem. That is, given matrix B, infer a phylogenetic tree T where each taxon in B uniquely corresponds to a leaf of T and the internal vertices of T are labeled by n binary characters with minimum parsimony score (i.e. the number of state changes is minimum). In general, this problem is NP-hard but we consider a version of this problem with two additional constraints. First, the root vertex of T must have state 0 for each character. Second, each character changes state from 0 to 1 only once on the tree and never reverts back from 1 to 0. The latter constraint is known as the infinite sites assumption in populations genetics, and also known as the two-state perfect phylogeny model. Let's formalize this.

Definition 1. *A rooted tree T with n leaves is a* two-state perfect phylogeny *for a given binary matrix $B \in \{0,1\}^{n \times m}$ provided:*

(i) each taxon (row of B) labels only one leaf,

(ii) each character labels only one edge, and

(iii) only the characters possessed by a taxon (leaf of T) are present on the unique path to the root.

Importantly, there need not exist a two-state perfect phylogeny for a given matrix B. This gives rise to the following problem.

Problem 1 (Two-state Perfect Phylogeny)**.** *Does a given binary matrix $B \in \{0,1\}^{n \times m}$ have a two-state perfect phylogeny T? If so, construct T.*

The way that this problem is posed should remind you of the large additive distance phylogeny problem, where solutions where edge-labeled trees that generated a given distance matrix D. The question that we asked there was to decide if D is additive. In our first attempt, we had only a constructive definition of solutions (relying on the existence of a tree T that generates D). Only later did we identify the four-point condition as a *complete characterization* of the solution set.

Similarly, here, we only have a constructive definition of binary matrices that are two-state perfect phylogeny matrices. This is not good enough! We want to identify a condition Φ that is both necessary and sufficient for a binary matrix B to be generated by a two-state perfect phylogeny tree T. That is, should T be a two-state perfect phylogeny for B then B must satisfy Φ (necessary); and should B satisfy Φ then there must exist a two-state perfect phylogeny T for B (sufficient). If Φ is necessary and sufficient then we say that Φ is a *complete characterization* of the set of two-state perfect phylogeny matrices. Can we find such a condition?

The answer is yes. There exists a condition Φ that can be computed in $O(nm)$ time. It will be helpful to sort the columns of B by the number of ones they contain, in descending order (largest first). Ties are broken arbitrarily. Let $\bar{B}$ denote the sorted binary matrix. We make the following observation, which follows directly from Definition 1.

Observation 1. *Let $\bar{B} \in \{0,1\}^{n \times m}$ be obtained from $B \in \{0,1\}^{n \times m}$ by sorting columns of B in descending order by the number of ones they contain. Matrix B has a two-state perfect phylogeny if and only if matrix $\bar{B}$ has a two-state perfect phylogeny.*

We have the following definition.

Definition 2. *Binary matrix $B \in \{0,1\}^{n \times m}$ is* conflict free *if no pair of columns c and d contain the three binary pairs $(0,1)$, $(1,0)$ and $(1,1)$.*

Clearly, using a naive algorithm we can check in $O(n^3m^2)$ if a matrix B is conflict free. We have the following lemma.

Lemma 1 (Shared-prefix property)**.** *Let d be the rightmost column in $\bar{B}$ possessed by two taxa f and g. If $\bar{B}$ is conflict free then f and g must be identical from column 1 to column d.*

Proof. Let $\bar{B}$ be a conflict-free matrix whose columns occur in sorted order. We proof the lemma by contradiction. That is, there exist taxa f and g that do no have a shared prefix and differ at character $c < d$. In other words, one of the two taxa will have a 0 at column c and the other taxon will have a 1 at the same column. Say, without loss of generality, $\bar{b}_{f,c} = 1$ and $\bar{b}_{g,c} = 0$. Column c must have at least as many 1s as column d. Thus, to offset the $(0, 1)$ pair at taxon g, we have that c and d must also contain the binary pair $(1, 0)$, say in taxon h. We thus have the following situation:

taxon	c	d
f	1	1
g	0	1
h	1	0

This means that $\bar{B}$ is not conflict free, a contradiction. Hence, the lemma follows and taxa f and g must be identical from columns 1 to d. □

We have the following theorem.

Theorem 1. *Matrix B has a two-state perfect phylogeny tree if and only if B is conflict free.*

Proof. (⇒) We start with the forward direction. Let T be a two-state perfect phylogeny tree for B. Consider two characters c and d. Let e_c (e_d) be the edge where c (d) was introduced. By Definition 1, taxa that possess c (or d) must be present as leaves below the edge e_c (or e_d). We distinguish four cases.

1. $e_c = e_d$.

   There cannot be a taxon with state $(1, 0)$ or $(0, 1)$ for the considered characters (c, d), as taxa that possess either c or d are in the same subtree below $e_c = e_d$.

2. The edge e_c is on the unique path from the root to e_d.

   There cannot be a taxon with state $(c, d) = (0, 1)$, as character c was introduced prior to character to d in T.

3. The edge e_d is on the unique path from the root to e_c.

   There cannot be a taxon with state $(c, d) = (1, 0)$, as character d was introduced prior to character to c in T.

4. The unique path to the root from e_c does not contain e_d and vice versa.

   There cannot be a taxon with state $(c, d) = (1, 1)$, as T does not contain a path from the root containing both character c and d.

Hence columns c and d are conflict free. Since we chose c and d arbitrarily, matrix B itself is conflict free (recall that conflict-free definition considers all pairs of column).

(⇐) We use Observation 1 and consider without loss of generality a sorted matrix $\bar{B}$ obtained from B. Observe that in any two-state perfect phylogeny T for B it must hold that the characters labeling the edges of the unique path from the root to a taxon f are exactly the characters that taxon f possesses. Moreover, the characters that taxon f possesses will appear in the same order in which they occur in $\bar{B}$. To see why, suppose that f possesses characters $c < d$. Per the previous statement, characters c and d label edges on the unique path from the root to f. Now $\bar{B}$ contains more 1s for character c than for character d. Thus, the edge e_c must occur prior to e_d. Hence, the characters that a taxon f possesses will appear in the same order in which they occur in $\bar{B}$. This must hold for any two-state perfect phylogeny T for $\bar{B}$. Thus, all that remains to show is that the n paths for each taxon can assembled into a single tree if $\bar{B}$ is conflict free.

We show how to construct a two-state perfect phylogeny T for a conflict-free matrix $\bar{B}$. The algorithm will construct T one row at a time. Initially, we create a root vertex. Next, we consider taxon 1. We construct a path T_1 composed of labeled edges for each character possessed by this first taxon maintaining the order imposed by $\bar{B}$. We extend the path with an unlabeled edge leading to a new vertex that will correspond to the first taxon. Clearly, T_1 is a two-state perfect phylogeny tree for taxon 1.

Let T_f be the partial tree constructed from taxa 1 to f, and assume inductively that T_f is a two-state perfect phylogeny tree for the first f taxa in $\bar{B}$. We now describe how to construct T_{f+1}. We traverse the edges in T_f starting from the root walking down the tree as long as the traversed edges contain characters that are possessed by $f+1$ in the same order as in $\bar{B}$. Let v be the last vertex visited on this traversed path, and let d denote the last matched character. As T_f is a two-state perfect phylogeny, this path is unique. We then create a new path extending from v and containing all characters $e > d$ that are possessed by $f+1$ that have not been matched. This new path maintains the order of the columns in $\bar{B}$. Finally, we extend the new path by one edge, leading to a new leaf that corresponds to $f+1$. We claim that T_{f+1} is a two-state perfect phylogeny tree for the first $f+1$ taxa in $\bar{B}$.

First, observe that each path to a leaf $h \leq f+1$ in T_{f+1} contains exactly the characters that taxon h possesses. Moreover, no character on the path to v is anywhere else in T_{f+1}, as T_f is a two-state perfect phylogeny. Thus, we only need to show that none of the characters that are in the new path from v to $f+1$ are in T_f. Let d be the rightmost character in $\bar{B}$ that taxon $f+1$ possesses and that is also possessed by a taxon in T_f. Let e_d denote the edge in T_f labeled by d. By definition, any taxon (leaf) h that is below e_d possesses d. We can apply the shared-prefixed property as $\bar{B}$ is sorted and conflict-free. Thus, by the shared-prefix property, rows h and $f+1$ are identical from column 1 to d. As such, the walk from the root to v is also a walk from the root to h. Moreover, by the choice of d, taxa h and $f+1$ do not possess any other common character $e > d$. Thus, none of the characters that are in the new path from v to $f+1$ are in T_f. Hence, T_{f+1} is a two-state perfect phylogeny from the first $f+1$ taxa of $\bar{B}$.

When all taxa have been processed the resulting tree is thus a two-state perfect phylogeny for $\bar{B}$ and B as well. $\square$

References

[1] Dan Gusfield. *ReCombinatorics: The Algorithmics of Ancestral Recombination Graphs and Explicit Phylogenetic Networks.* The MIT Press, 2014.