Integer Programming in Computational Biology

D. Gusfield

University of California, Davis

Presented December 12, 2016.

There are many important phylogeny problems that depart from simple tree models:

- Missing entries
- Data generated by complex biology, such as recombination or recurrent mutation
- Genotype (conflated) sequences, rather than simpler haplotype sequences

Most of these problems are NP-hard, although some elegant poly-time solutions exist (and are well-known) for simpler data.

Question

Can Integer Programming efficiently solve these problems in practice on ranges of complex data of current interest in biology?

We have recently developed ILPs for many such problems and intensively studied their performance (speed, size and biological utility).

In this talk I will first concentrate on
ILP problems relating to networks caused by back mutation and recombination. Then, if time permits, I will talk about RNA folding.

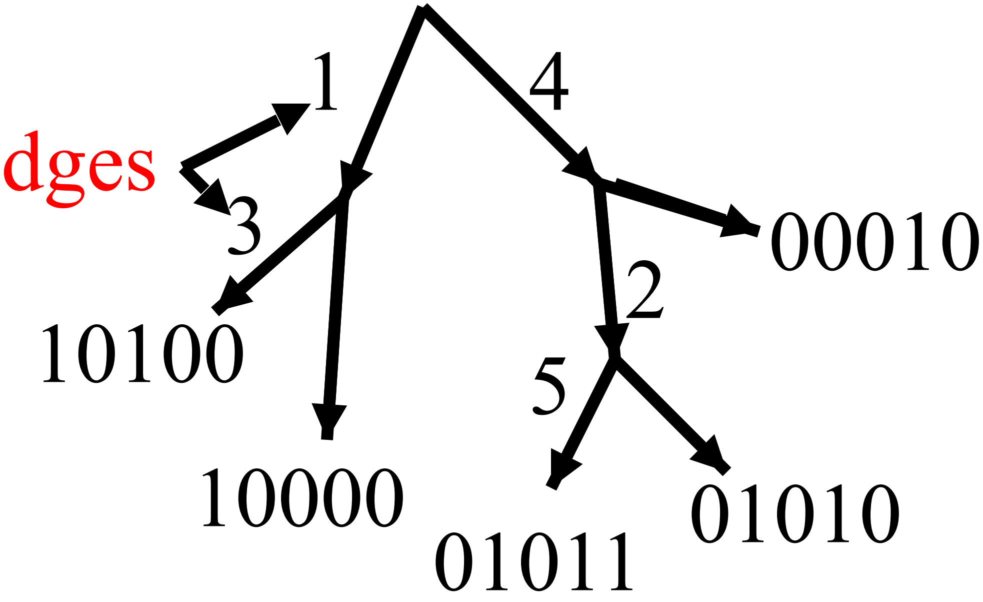
We start with the Perfect-Phylogeny Model, which is the case when neither back mutation or recombination are allowed.

Starting Model: Perfect Phylogeny (infinite sites) model for binary sequences

Only one mutation per site allowed.

sites 12345

Ancestral sequence 00000



Site mutations on edges

The tree derives the set M:
10100
10000
01011
01010
00010

Extant sequences at the leaves

When can a set of sequences be derived on a perfect phylogeny?

Classic NASC: Arrange the sequences in a matrix. Then (with no duplicate columns), the sequences can be generated on a unique perfect phylogeny if and only if no two columns (sites) contain all three binary pairs:

0,1 and 1,0 and 1,1

This is the 3-Gamete Test.

Each binary pair is called a gamete.

A pair of sites that has all three gametes is called incompatible.

Problem MD: Missing Data

Given ternary sequences (0s, 1s, ?s), change the ?s to 0s and 1s in order to minimize the resulting number of incompatible pairs of sites. NP-hard.

Simple ILP for the Missing Data problem

Create a binary variable Y(i,p) for a ? in cell (i,p), indicating whether the cell will be set to 0 or to 1.

For each pair of sites p, q that could be made incompatible, let D(p,q) be the set of missing or deficient gametes in site pair p,q.

For each gamete a,b in D(p,q), create the binary variable B(p,q,a,b),

and create inequalities to set it to 1 if the Y variables for cells for sites p,q are set so that gamete a,b is created in some row for sites p,q.

Example

p	q
0	0
?	1
1	0
?	?
?	0
0	?

D(p,q) = {1,1; 0,1}

To set the B variables, the ILP will have inequalities for each a,b in D(p,q), one for each row where a,b could be created at site p,q.

For example, for a,b = 1,1 the ILP has:
Y(2,p) <= B(p,q,1,1)   for row 2
Y(4,p) + Y(4,q) -- B(p,q,1,1) <= 1   for row 4

Example continued

p	q
0	0
?	1
1	0
?	?
?	0
0	?

D(p,q) = {1,1; 0,1}

For a,b = 0,1 the ILP has:

Y(2,p) + B(p,q,0,1) => 1   for row 2
Y(4,q) -- Y(4,p) -- B(p,q,0,1) <= 0   for row 4
Y(6,q) -- B(p,q,0,1) <= 0   for row 6

The ILP also has a variable C(p,q) which is set to 1 if every gamete in D(p,q) is created at site-pair p,q.

In the example:

$$B(p, q, 1, 1) + B(p, q, 0, 1) -- C(p,q) <= 1$$

So, C(p,q) is set to 1 if (but not only if) the Y variables for sites p, q (missing entries in columns p, q) are
set so that sites p and q become incompatible.

If M is an n by m matrix, then we have at most nm Y variables; $2m^2$ B variables; $m^2/2$ C variables; and $O(nm^2)$ inequalities in worst-case.

Finally, we have the objective function:

$$\text{Minimize} \sum_{(p,q) \text{ in } P} C(p, q)$$

Where P is the set of site-pairs that could be made to be incompatible.

Or, we could require that the sum of the C(p,q) variables be zero, and then there is a way to set the missing values to form a Perfect Phylogeny, if and only if the ILP is feasible.

Empirically these ILPs solve very quickly, in fractions of seconds or seconds for n and m up to hundreds of rows and columns.

The software for to create the ILP formulations was written in 2006, but is paying dividends now.

Persistent and Dollo: Deviations from Perfect Phylogy

- Extends the Perfect Phylogeny Model by allowing each site to revert from state 1 to state 0.
- Persistent Phylogeny: Each site mutates back to 0, at most once in the tree. So this is like the infinite sites model in for both forward and backward mutations.
- Dollo Model: Forward mutation once, but backwards any number of times.

A range of possibilities

- So given binary data either it can be generated on a Perfect Phylogeny, or a Galled-Tree, or a Persistent Phylogeny, or a Dollo Phylogeny, or none of the above - i.e., a more general network is needed.
- Given binary data, how do we determine what case we have?

The Dollo model was introduced more than 100 years ago, but the persistent phylogeny model was only introduced recently, by T. Przytycka and D. Durand, has been studied intensively by P. Bonizzoni and co-authors.

The Persistent Phylogeny Problem: Given M, determine if M can be derived on a Persistent Phylogeny.

The question of whether the Persistent Phylogeny Problem is NP-hard is open. So, we take an ILP approach.

The Persistent Phylogeny Problem

- The key to the ILP for it, is the following formalism developed by P. Bonizzoni et al. in 2013.

Definition: Given a binary matrix M, the extended matrix Me contains two columns, j1 and j2, for each column j in M.

Column j1 of Me is derived from column j in M by replacing every occurrence of `0' in column j of M with `?' in column j1 of Me.

Column j2 of Me is derived from column j1 by replacing every occurrence of `1' in j1 with `0’.

So a 0 in j becomes ??, and a 1 becomes 10.

Completing Me

A completion M’e of Me changes each ?’ to either 0 or 1, with the requirement that for every pair of sites (j1, j2) in Me that originated from an entry of value 0 in cell (i,j) in M, cells (i, j1) and (i, j2) in M’e must get the same value, i.e., they either get 0,0 or 1,1.

Extension Me and Completion M’e of M

M	Me	M’e
1110	101010??	10101000
0111	??101010	11101010
0000	????????	00000000
1010	10??10??	10001000
1100	1010????	10101100
1111	10101010	10101010

For character j in M, character j1 in Me is ``a mutation of character j has occurred”, and character j2 is ``a back mutation of character j has occurred”.

Theorem of Bonizzoni et al.

M can be represented by a Persistent Phylogeny if and only if there is a completion M’e of Me that is a Perfect Phylogeny. And if so, the perfect phylogeny for M’e is a Persistent Phylogeny for M.

This theorem shows the way to formulate the ILP for the Persistent Phylogeny problem.

The ILP

Given M, we form Me and treat that as input to problem MD, but for every pair of sites (j1, j2) in Me that originated from an entry of value 0 in cell (i, j) in M, we add the constraint: Y(i,j1) = Y(i,j2).

Then the ILP has optimal value zero if and only if M has a persistent phylogeny.

Problems related to M1

- Site-Removal Problem for complete data: Remove the minimum number of sites from the data, so that no incompatibilities remain. This is a common approach to incompatible data in phylogenetics. NP-hard.
- Site-Removal Problem with missing data (S1): Impute values for the missing entries to minimize the solution to the resulting Site-Removal Problem for complete data.

ILP for S1 - a simple extension to M1

- For each site i, let D(i) be a variable set to 1 if and only if site i is removed.
- For each site-pair p,q in P, add the inequality D(p) + D(q) -- C(p,q) => 0

  to the M1 formulation.

The objective function is now

Minimize Sum D(i)

Genotypes and Haplotypes

Each individual has two "copies" of each chromosome.

At each site, each chromosome has one of two alleles (states) denoted by 0 and 1 (motivated by SNPs)

0 1 1 1 0 0 1 1 0

1 1 0 1 0 0 1 0 0

Two haplotypes per individual

Merge the haplotypes

2 1 2 1 0 0 1 2 0

Genotype for the individual

Haplotyping (Phasing) Problem

- Biological Problem: For disease association studies, haplotype data is more valuable than genotype data, but haplotype data is hard to collect. Genotype data is easy to collect.

- Computational Problem: Given a set of n genotypes, determine the original set of n haplotype pairs that generated the n genotypes. This is hopeless without a genetic model or objective function that reflects the model. Many such models have been studied.

PPH model and objective

Given a set of genotypes, find (if possible) an explaining set of haplotypes (one pair for each genotype) that passes the ``four gamete test”.

The PPH problem can be solved in linear time by a very complex algorithm. But it is simple to formulate an ILP for the PPH problem.

A Natural Extension of the PPH model

MinIncompat Problem (HM1): Haplotype to minimize the resulting number of incompatible pairs of sites.

NP-hard problem, but solved efficiently in practice by an ILP which is a simple modification of the ILP for problem M1.

The MinIncompat ILP becomes an ILP for PPH with the addition of a constraint that requires the solution to have value 0. The resulting ILP is feasible if and only if there is a PPH solution.