Recent Progress on the Multi-State Perfect Phylogeny Problem

Dan Gusfield
UC Davis

June 20, 2011 Newton Institute, Cambridge Univ.

As we last saw,

In September 2007 …

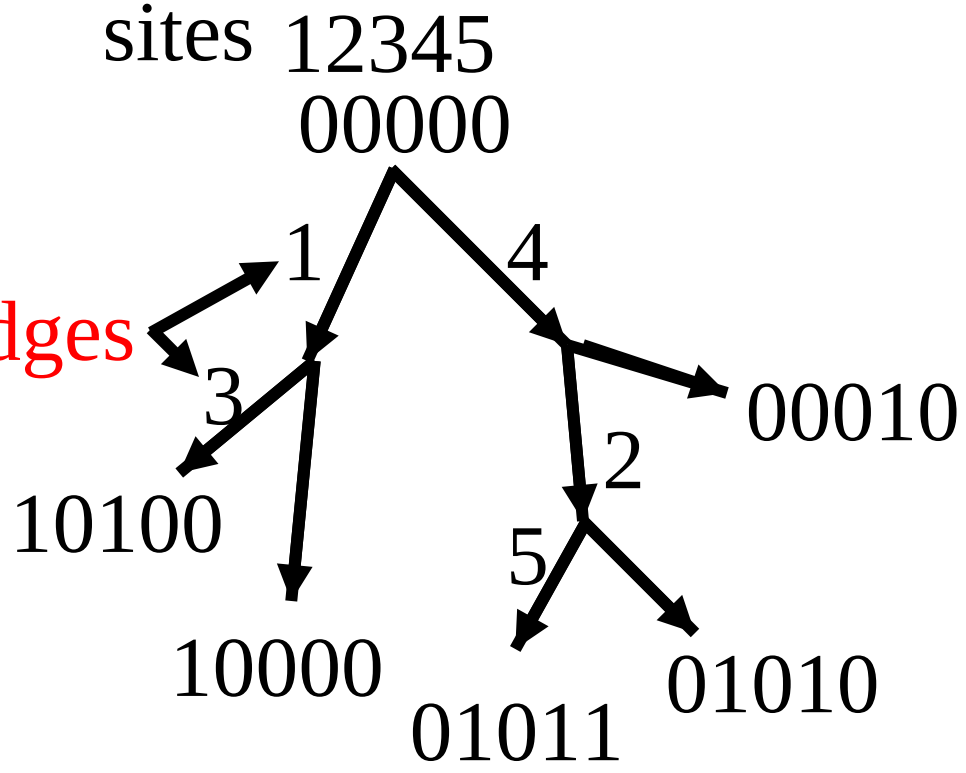
The Perfect Phylogeny Model for binary sequences

Only one mutation per site allowed (infinite sites)

Site mutations on edges

The tree derives the set M:

10100
10000
01011
01010
00010



Extant sequences at the leaves

When can binary data be derived on a perfect phylogeny?

The Four Gametes Condition (compatibility, splits equivalence)

Four Classic Gametes Theorem:

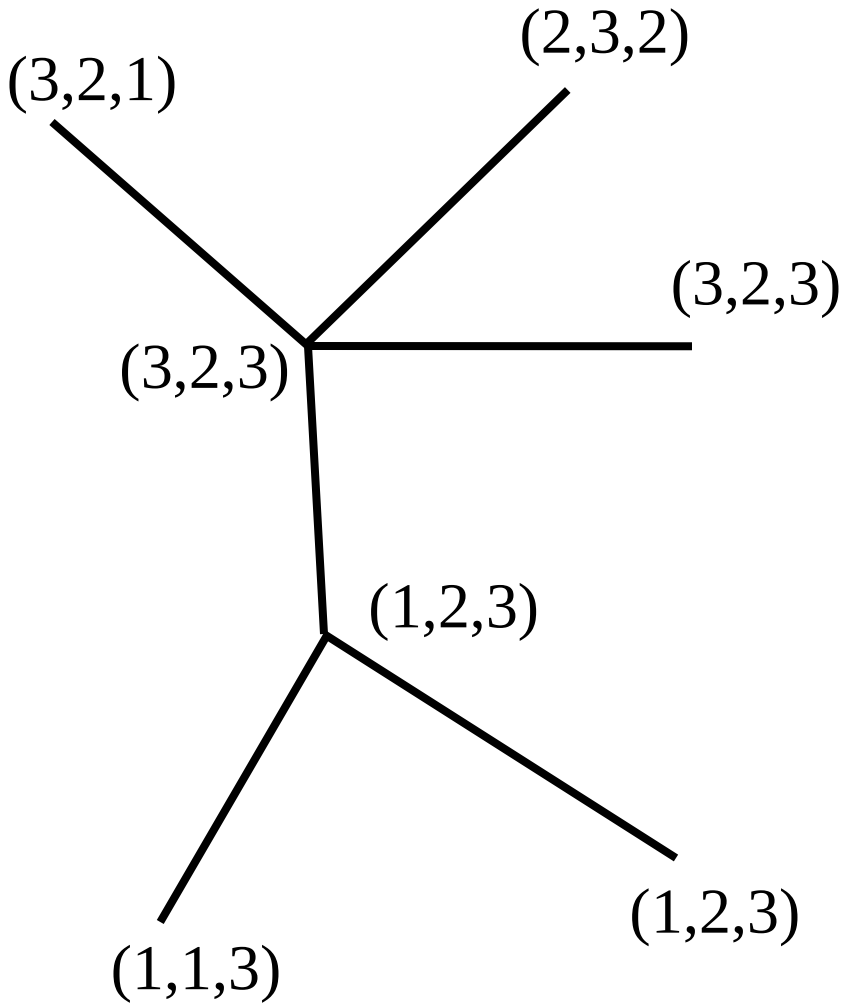
For binary data, there is a perfect phylogeny if and only if no pair of sites contains all four gametes, i.e., all four binary combinations 0,0; 0,1; 1,0; 1,0

Beyond Binary; beyond SNPs

The binary perfect phylogeny model has been widely used in population genetics (four-gametes), phylogenetics (compatibility); and many problems and methods build on the model (haplotyping, networks with recombination).

But, non-binary, non-SNP data is becoming more important in population genomics: CNVs, full DNA sequences, micro-sats; quantitative phenotyping; other applications in phylogenetics.

A 3-state perfect phylogeny



	A	B	C
1	3	2	1
2	2	3	2
3	3	2	3
4	1	1	3
5	1	2	3

M

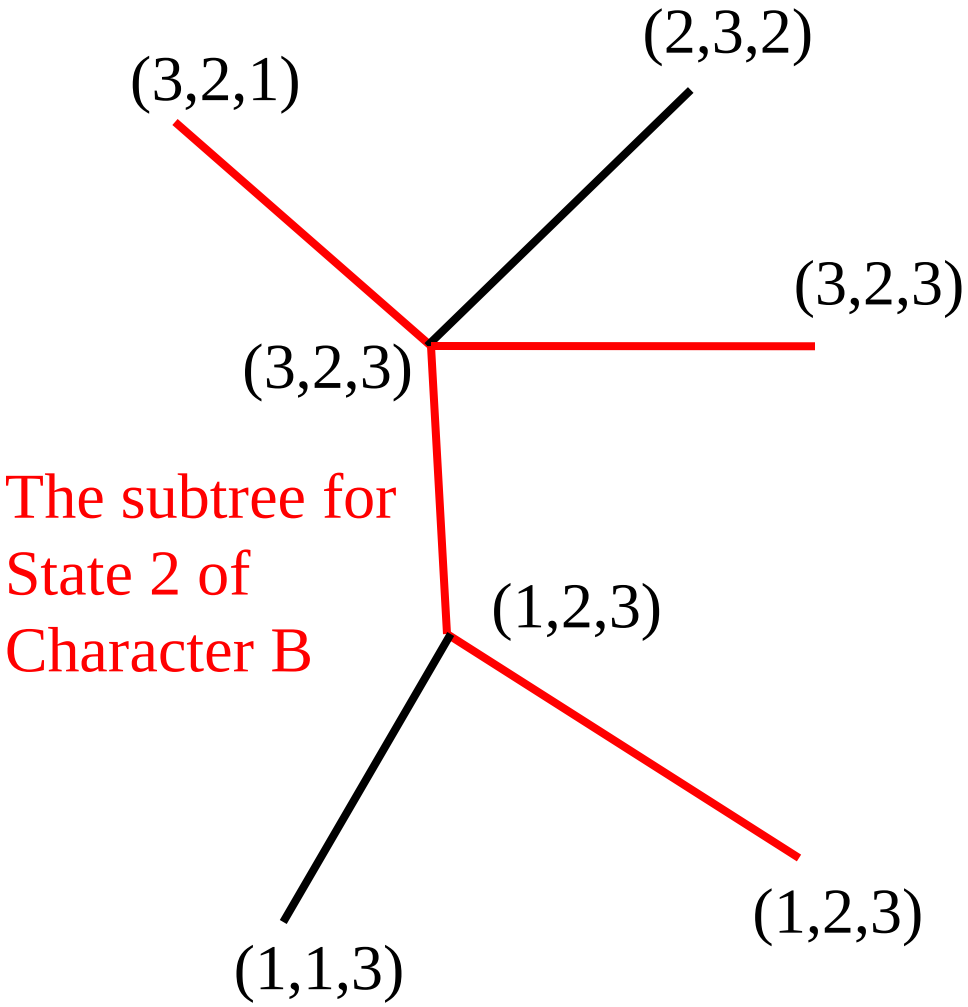
n = 5 number of taxa
m = 3 number of sites
k = 3 number of states

A formal definition of a k-state unrooted perfect phylogeny

- Input consists of n sequences M with m sites (characters) each, where each site can take one of k > 2 states (values).
- T has n leaves, one for each sequence X in M, labeled by X.
- Each node of T is labeled with an m-length sequence (not necessarily from M) where each site has a value from 1 to k.
- For each character-state pair (C,s), the nodes of T that are labeled with state s for character C form a connected subtree of T. This is the convexity requirement.

This more reflects the infinite alleles model rather than the infinite sites model in binary perfect phylogeny. It also models Dollo parsimony.

Convexity



	A	B	C
1	3	2	1
2	2	3	2
3	3	2	3
4	1	1	3
5	1	2	3

M

n = 5 number of taxa
m = 3 number of sites
k = 3 number of states

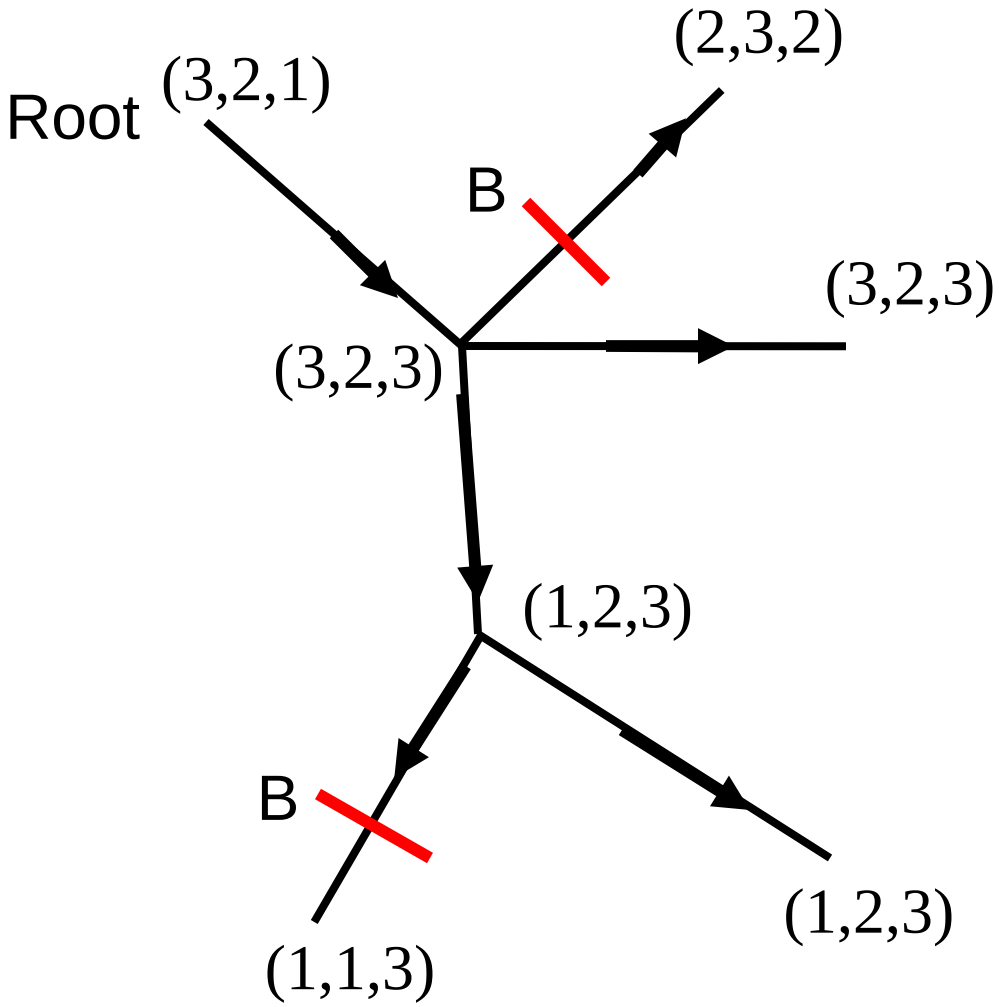
Alternative view of the convexity requirement for T

Arbitrarily choose a root of T and direct all the edges of T away from the root.

Then, any character can mutate into a given state at most once, but never mutate into its root state.

This view makes a k-state perfect phylogeny a natural generalization of a binary perfect phylogeny.

The requirement that there is at most one mutation into any state of a character reflects the infinite alleles model in population genetics, and the Dollo parsimony model of evolutionary biology.



	A	B	C
1	3	2	1
2	2	3	2
3	3	2	3
4	1	1	3
5	1	2	3

M

n = 5
m = 3
k = 3

Generalizing the Four Gametes Condition

Fumei Lam, Gusfield, Sridhar

(WABI 2009, SIAM Discrete Math 2011)

General K-state Perfect Phylogeny Problems (Gusfield, JCB 2010)

Existence Problem:
Given M and k, is there a k-state Perfect Phylogeny for M?

Missing Data (MD) Problem:
For a given k, if there are cells in M without values, can values less than or equal to k be imputed so that the resulting matrix M' has a k-state perfect phylogeny?

Handling missing data extends the utility of the perfect-phylogeny model.

Status of the Existence Problem

Poly-time algorithm for 3 states, Dress-Steel (1993)

Poly-time algorithm for 3 or 4 states, Kannan-Warnow (1994)

Poly-time algorithm for any fixed number of states - polynomial in n and m, but exponential in k, Agarwalla and Fernandez-Baca (1994)

Speed up of the AFB method by Kannan-Warnow (1997)

When k is not fixed, the existence problem is NP-hard

The missing data challenge

The general AFB,KW algorithms that solve the existence problem are not easily adapted to handle the missing data problem. They seem to extend only by brute-force enumeration of imputed values.

So, we need another approach to the missing data problem.

Prior work on the Missing Data problem

NP-complete even for k = 2; effective, practical approaches for k = 2. (GFB in cocoon 2007; Satya, Mukherjee, TCBB 2008)

Polynomial-time methods for a `directed' variant of k = 2.

specialized ILP methods for k = 3,4,5 (Newton 2007)

The general k-state MD problem (JCB 2010)

New approach to existence and missing data problems

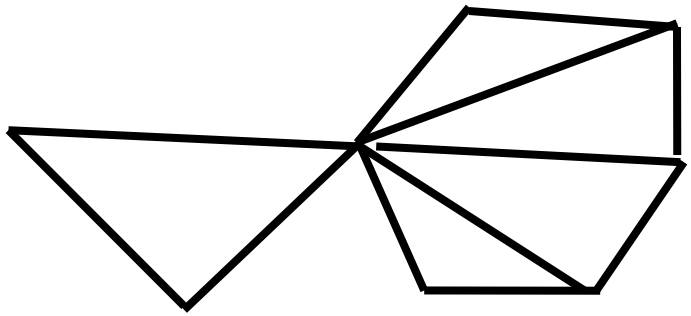
Based on an old theorem and newer techniques.

Old theorem: Buneman's Theorem relating Perfect-Phylogeny to chordal graphs. (thirty-five years old)

Newer techniques and theorems: Minimal triangulations of a non-chordal graph to make it chordal. The literature on minimal triangulations is current and ongoing.

Definition: Chordal Graphs

A graph G is called Chordal if every cycle of length four or more contains a chord. Chordal graphs are also called triangulated graphs.

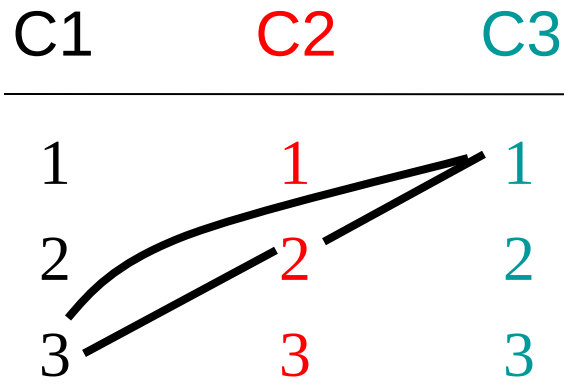


G

Buneman's Approach to Perfect Phylogeny (1974)

C1	C2	C3
3	2	1
2	3	2
3	2	3
1	1	3
1	2	3

Input M, n by m



G(M)

Partition-Intersection Graph G(M) has one node for each character-state pair in M, and an edge between two nodes if and only if there is a row in M with both those character-state pairs.

Each row of table M induces a clique in G(M).

G(M) is the superposition of m

Definitions

If M has m characters, then G(M) is an m-partite graph.
The nodes associated with a single
character (class in the partition) are given a distinct color.

An edge (u,v) not in G(M) is called legal
if u and v do not have the same color.

Two nodes with the same color are called a
mono-chromatic pair.

Buneman's Theorem

Theorem (Buneman 1974)

There is a perfect phylogeny for M if and only if legal edges can be added to graph G(M) to make it chordal.

If there is such a chordal graph, denote it G'(M).

G'(M) is called a legal triangulation of G(M).

From a Chordal Graph to a Perfect Phylogeny

Fact: Given a legal triangulation G'(M), a Perfect Phylogeny for M can be constructed in linear time.

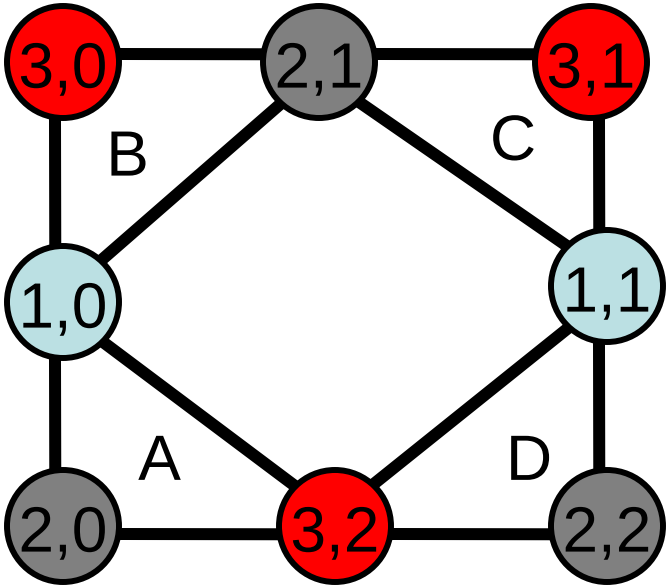
The algorithms are based on `perfect elimination orders' and `clique trees', classic objects in the chordal graph literature.

Example

M

	1	2	3
A:	0	0	2
B:	0	1	0
C:	1	1	1
D:	1	2	2

Each node represents a Character-State pair

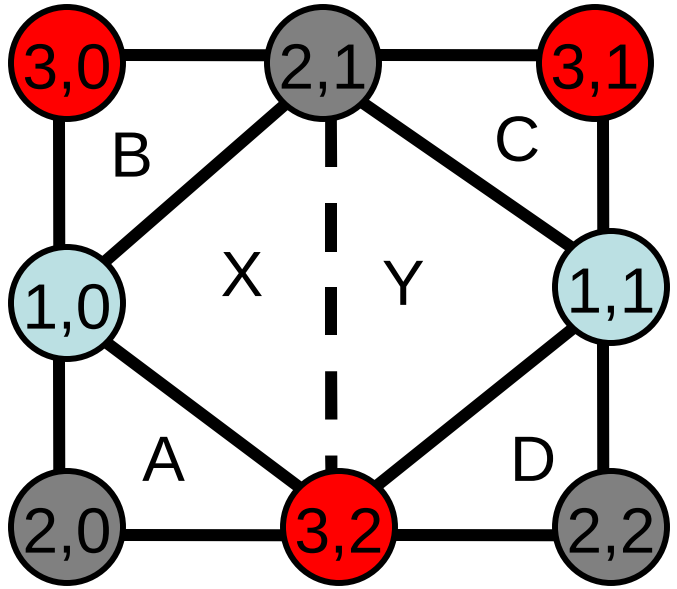


G(M)

A legal triangulation

M

	1	2	3
A:	0	0	2
B:	0	1	0
C:	1	1	1
D:	1	2	2



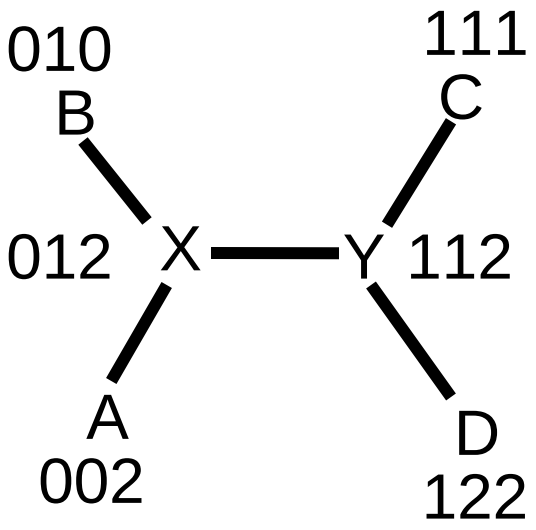
G’(M)

Yields a Perfect Phylogeny

M

	1	2	3
A:	0	0	2
B:	0	1	0
C:	1	1	1
D:	1	2	2

(Fact: every clique-tree of the Chordal graph G'(M) is a perfect Phylogeny for M)



One node in T for each maximal clique in G'(M)

What about Missing Data?

If M is missing data, build the partition intersection graph G(M) using the known data in M. Buneman's theorem still holds:

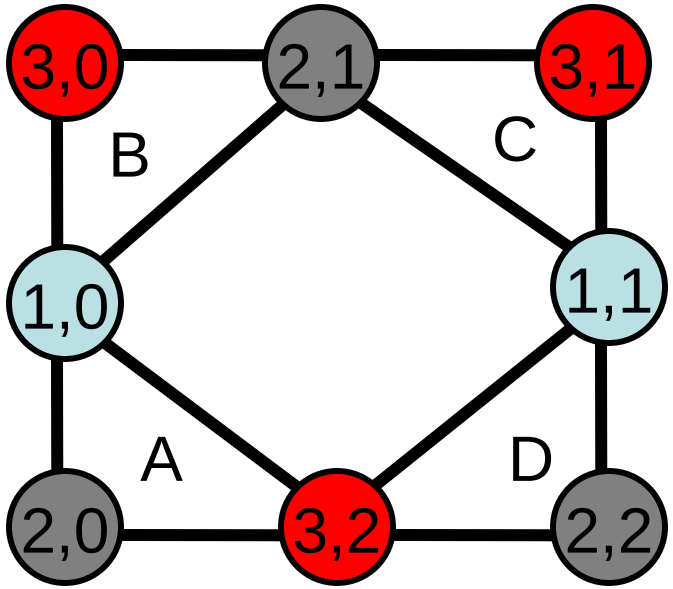
Theorem: There is a perfect phylogeny for some imputation of missing data in M, if and only if there is a legal triangulation of G(M).

The legal triangulation gives a perfect phylogeny T for M with some imputed data, and then the imputed values for M' can be obtained from T.

Example

M

	1	2	3
A:	0	0	2
B:	0	1	0
C:	1	1	1
D:	1	2	2

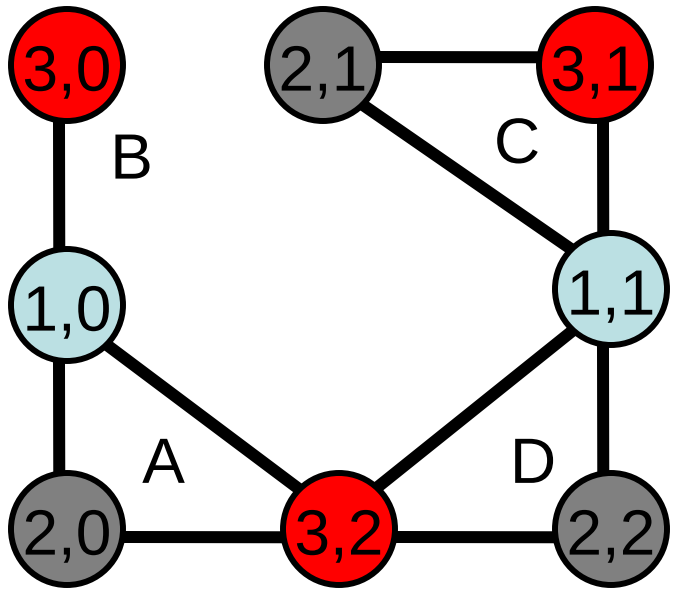


G(M)

Example

M

	1	2	3
A:	0	0	2
B:	0	?	0
C:	1	1	1
D:	1	2	2



G(M)

The Key Problem

The PI graph is conceptually perfect for modeling missing data.

So the key problem, in both the Existence and the Missing Data problems, is how to find a legal triangulation, if there is one.

Some triangulation problems are NP-hard (Tree-width, Minimizing the number of added edges).

But, there is a robust and still expanding literature on efficient algorithms to find a minimal triangulation of a non-chordal graph.

Minimal triangulation

A triangulation of a non-chordal graph G is minimal if no subset of added edges is a triangulation of G.

Clearly, if there is a legal triangulation G'(M) of G(M), then there is one that is a minimal triangulation. A minimal triangulation is good enough for us.

So we can take advantage of the minimal triangulation technology, and the contemporary literature. The minimal vertex separators are the key objects.

Minimal vertex separators

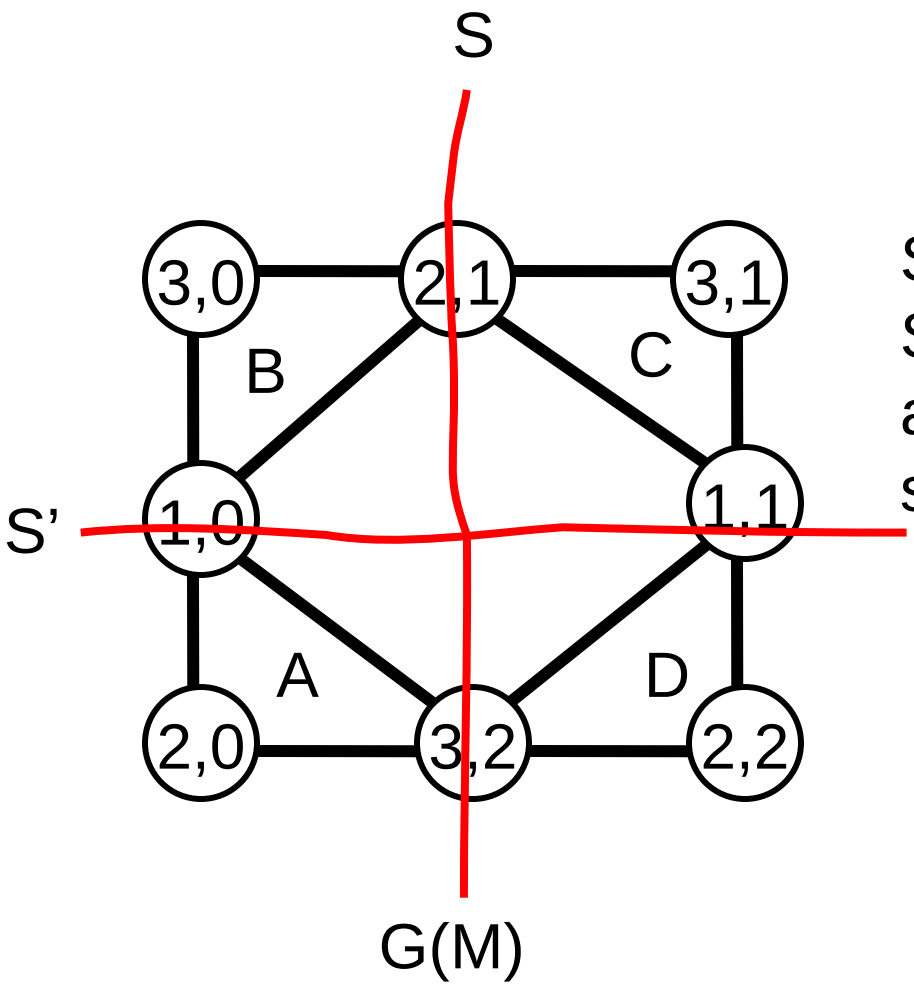
A set of nodes S whose removal separates vertices u and v is called a u,v separator. S is a minimal u,v separator if no subset of S is a u,v separator.

S is a `minimal separator' if it is a minimal u,v separator for some vertex pair u,v.

Minimal separator S crosses minimal separator S', if S separates some pair of nodes in S'.

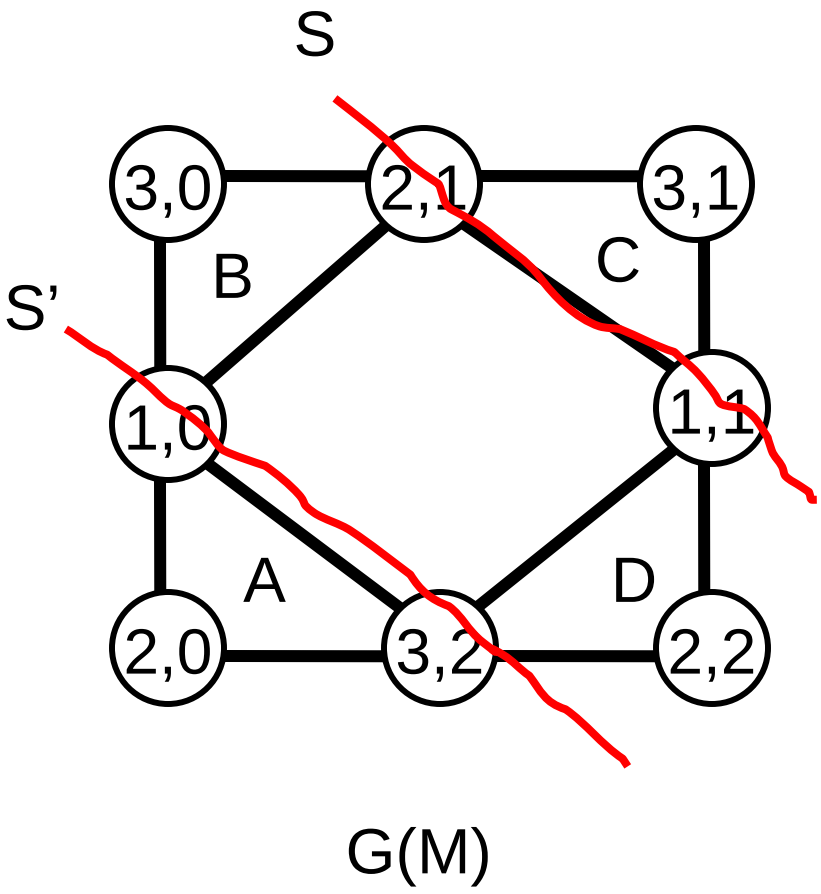
Crossing is a symmetric relation for minimal separators.

Example



S = {(2,1), (3,2)} and
S' = {(1,0), (1,1)}
are crossing minimal separators.

Example



G(M)

S = {(2,1), (1,1)} and
S' = {(1,0), (3,2)} are non-crossing minimal separators.

A lucky break:
A complete characterization of the minimal triangulations of G was derived in 1997

Definition: Completing a minimal separator S means adding all the missing edges between pairs of nodes in S to make S a clique.

Capstone Theorem on Minimal Triangulations

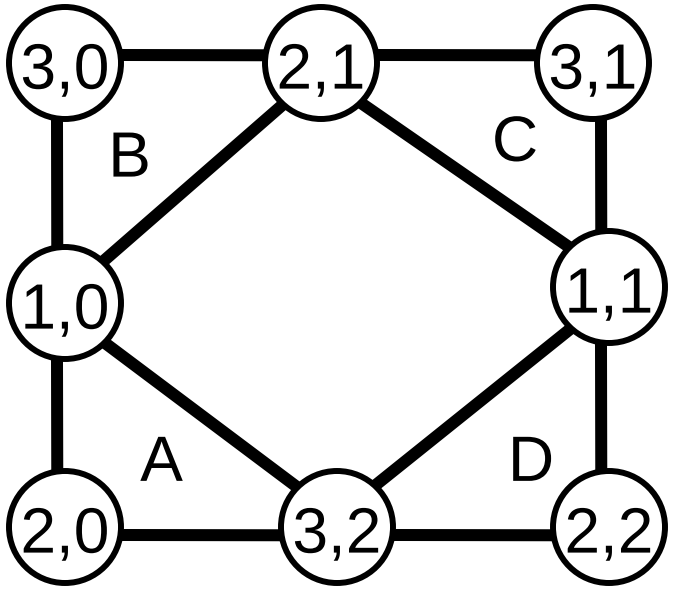
Parra, Scheffler (1997): Every minimal triangulation of G is obtained by completing each minimal separator in a
maximal set of pairwise non-crossing minimal separators of G.

Conversely, completing every minimal separator in a maximal set of pairwise non-crossing minimal separators yields a minimal triangulation of G.

Example:

M

	1	2	3
A:	0	0	2
B:	0	1	0
C:	1	1	1
D:	1	2	2



G(M)

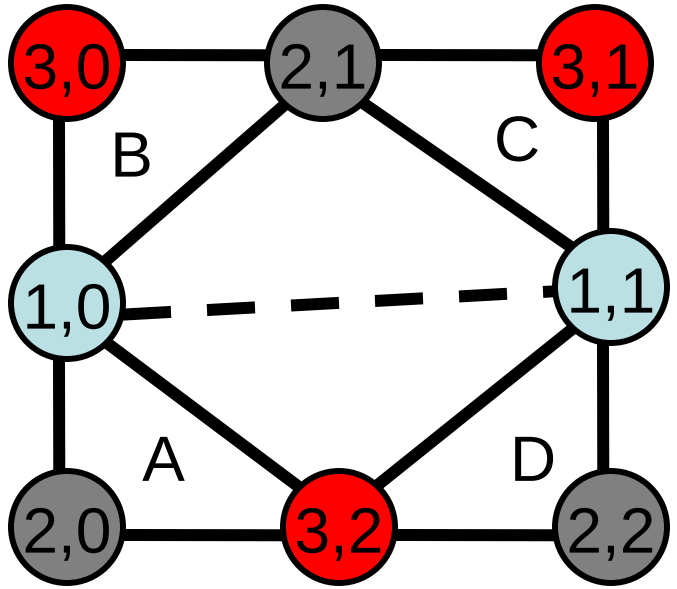
There are 6 minimal separators.

There are two maximal sets of 5 pairwise non-crossing minimal separators.

A minimal (illegal) triangulation obeying the P,S Theorem

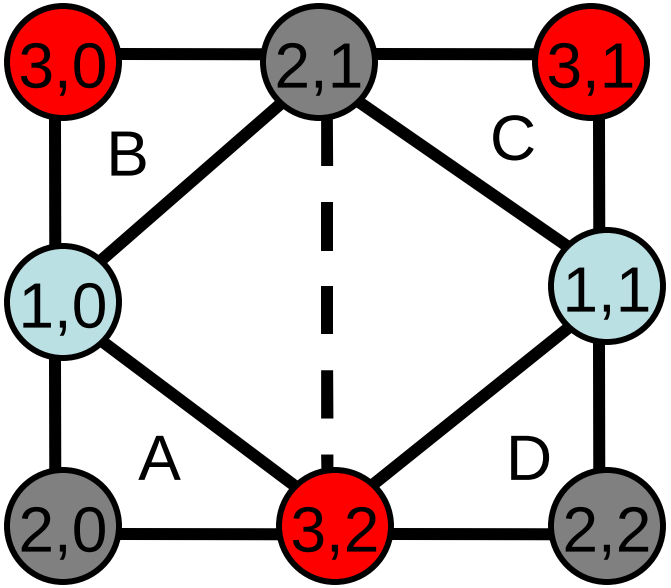
M

	1	2	3
A:	0	0	2
B:	0	1	0
C:	1	1	1
D:	1	2	2



G’(M)

A legal minimal triangulation



G’(M)

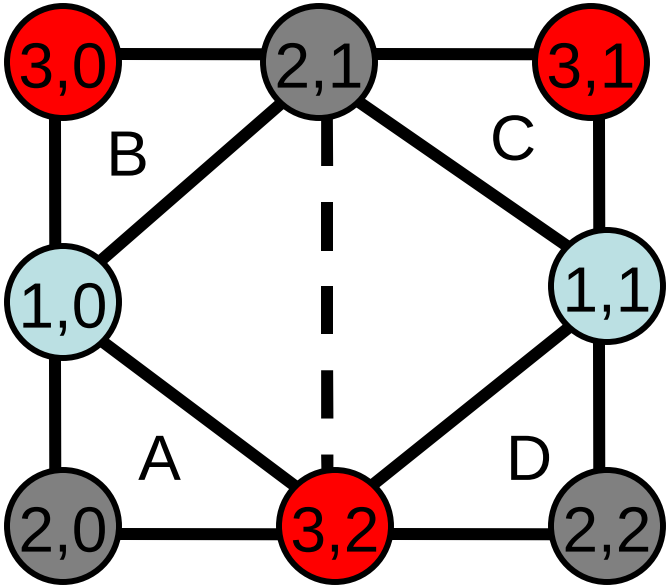
Back to Perfect Phylogeny

A minimal separator S in the partition intersection graph G(M)
Is called legal if it does not use an edge between two nodes of the same color, and is called illegal if it does.

P,S Theorem can be used to prove the Main New Results

Theorem 1:
There is a perfect phylogeny for M, even if M has missing data,
If and only if there is a set Q of pairwise non-crossing, legal, minimal separators in G(M) that separate every mono-chromatic pair of nodes in G(M).

The legal minimal triangulation, obeying Theorem 1



G’(M)

From G’(M), we get a perfect phylogeny for M.

Hint of the Proof of Theorem 1

First a different Theorem

Theorem 2: There is a perfect phylogeny
for M if and only if there is a set of pairwise non-crossing
legal minimal separators that cross all of the illegal minimal separators. This is true even when M has missing data.

Corollaries to Theorem 1

Cor 1: If there is a mono-chromatic pair of nodes in G(M) that is not separated by any legal minimal separator, then M has no perfect phylogeny.

Cor 2: If G(M) has no illegal minimal separators, then M has a perfect phylogeny.

Cor 3: If every mono-chromatic pair of nodes is separated
by some legal minimal separator, and no legal minimal separators cross, then M has a perfect phylogeny.

Recipe to solve the missing data problem with Theorem 1

Given M, find all legal minimal separators in G(M);
for each legal minimal separator, determine which mono-chromatic pairs of nodes it separates, and which legal minimal separators it crosses.

Determine if any of the Corollaries hold. If so, either there is no perfect phylogeny (Cor. 1) or a set Q needed in Theorem 1 can be found greedily.

If no Cor. holds, set up and solve a (straightforward) integer linear program to find a set Q of pairwise non-crossing legal minimal separators that separate every mono-chromatic pair of nodes in G(M).

If the ILP is feasible, greedily extend Q to be a maximal set of pairwise non-crossing legal minimal separators, and use Q to get a legal triangulation G'(M) of G(M).

From G'(M), construct a perfect phylogeny T, and from T impute values for the missing entries.

Conceptually nice, but

Does it work in practice?

It works surprisingly (shockingly) well

Simulations with data from program ms, characteristic of many current applications in phylogenetics and population genetics - but not genomic scale or tree-of-life scale.

Surprising empirical results

The minimal separators are found quickly by existing algorithms from 1999: cubic-time per minimal separator, but we have methods (not in the paper) to speed this up.

When there is no missing data, all the legal minimal separators can be found in O(nm^2) worst-case time, for any fixed k.

The observed number of minimal separators is small. There are few crossing pairs of legal minimal separators.

Until a large percentage of missing data, most problems are solved by the Corollaries, without the need for an ILP.

When an ILP is needed, it has been tiny. For the existence problem, the size of the ILP is polynomialy bounded.

The ILPs solve quickly in practice - all have solved in 0.00 CPLEX-reported seconds (CPLEX 11 on 2.5 Ghz machine).

Most solve in the CPLEX pre-processor.

n, m	k	% miss	# v, e	# $L, Iseps$	sep time	% $d, c1, c2, c3$	% ilps	# var con	# inf	% pre
20, 20	4	0	70.9, 990	16, 0.47	0.07 s	0, 0, 72, 27	1	3, 2	0	100
20, 20	4	35	62.3, 696	20.6, 5.4	0.066 s	0, 0, 10, 15	75	8.4, 10.3	0	88
40, 40	10	20	272, 7141	57, 27	1.03 s	0, 0, 0, 11	89	19, 36	0	91
40, 40	10	35	255, 5874	69, 11	2.3 s	0, 0, 0, 0	100	36, 136	0	40
60, 60	15	10	581, 24904	89, 56	7.4 s	0, 0, 0, 9	91	22, 67	0	89
60, 60	15	20	559, 22553	94, 107	9.9 s	0, 0, 0, 1	99	37, 118	0	85
80, 80	10	10	597, 35841	102, 37	14 s	0, 0, 0, 21	79	18, 48	0	95
80, 80	10	20	584, 33863	109, 70	18 s	0, 0, 0, 7	93	32, 83	0	87
80, 80	20	0	1026, 62540	118, 85	32 s	0, 0, 0, 12	88	21, 106	0	83
80, 80	20	1	1023, 62081	118, 91	33 s	0, 0, 0, 12	88	22, 1113	0	84
100, 100	10	0	791, 62733	122, 33	33 s	0, 0, 0, 36	64	13, 51	0	94
100, 100	10	5	783, 60924	123, 42	34 s	0, 0, 0, 36	64	17, 60	0	94
20, 20	10	10	119, 1493	23, 6	0.096 s	3, 27, 27, 23	20	6,8	1	95
20, 20	10	35	103, 1015	29, 16	0.095 s	1, 12, 3, 3	81	12, 21	1	75
40, 40	10	20	267, 7217	55, 97	2.4 s	15, 46, 1, 4	34	20,56	1	85
40, 40	10	35	250, 5899	67, 334	7.3 s	3, 55, 0, 0	42	36,145	1	41
80, 80	10	5	610, 38359	97, 41	15 s	50, 28, 1, 3	18	13, 31	0	94
80, 80	10	20	584, 34353	105, 117	29 s	35, 42, 1, 2	20	36,86	0	90
80, 80	20	5	1006, 60449	114, 159	55 s	2, 49, 0, 3	46	27, 128	0	83
80, 80	20	20	954, 52563	107, 321	82 s	0, 54, 0, 0	46	60, 264	0	65

So

Although the chordal graph approach may at first seem impractical, it works on a large range of data of sizes that are typical of current phylogenetic problems, and degree of missing data.

When there is no missing data

All of the legal minimal separators can be found in O(nm^2) time for any fixed k.

Details - proper cluster; proper cluster induces a legal separator in the PI(M); can test in O(nm) time if a separator S is minimal - minimal if and only if there are two full connected components in G - S.

More structure

The empirical results suggest the existence of more combinatorial structure in the perfect-phylogeny problem. And more has been recently found.

(F. Lam) When k = 3, a NASC for the existence of a 3-state perfect-phylogeny is:

Every mono-chromatic pair of nodes in G(M) is separated by some legal minimal separator. (Compare to Theorem 1).

This does not extend to k = 4.

Using this for 3-states

- The NASC implies a simple algorithm to decide if a 3-state problem has a PP, in time equal to existing algorithms.
- But, constructing the PP in competitive time, using the chordal graph approach was a challenge.
- Now solved in upcoming WABI 2011 paper by Gysel, Lam, Gusfield.

Removable data

The CR Problem:
Given data that does not have a k-state perfect phylogeny, what is the minimum number of characters to remove so that the remaining data does have a k-state perfect phylogeny?

The MDCR Problem:
When there is missing data and there is no solution to the MD problem, what is the minimum number of characters to remove to that the remaining data does have a solution to the MD problem?

(Gysel,Gusfield – ISBRA 2010)

The CR problem can be formulated as an ILP using the chordal-graph view. It also solves the MDCR problem.

Preprocessing ideas can reduce the time needed to find all of the minimal separators. Time reduced to 1/3 to 1/2 of the original times.

Solving CR and MDCR by the chordal graph approach

In this solution we see the utility of using illegal minimal separators.

Main Theorem

The CR and MDCR problems are solved by finding a minimal triangulation, which may be illegal, maximizing the number of legal characters it contains.

It is easy to cast this as an ILP, again using the PS algorithm to ensure a minimal triangulation, and then adding in additional constraints to implement the theorem.

Empirical results for CR and MDCR

Good enough or I wouldn't be talking about it.

Details are in the paper.

Other results

- The 3-state perfect phylogeny existence problem reduces to 2-SAT, in competitive time. (Gusfield, Wu 2010)
- The k-state problems reduce to 2-state problem with missing data, in poly-time. Then existing 2-state approaches can then be used to solve the problems. (Stevens, Gusfield (WABI 2010)

One more

- MD and Removable data problems when the data satisfies the ``Rich Data Hypothesis" and generalizations of it.
- Poly-time algorithms are given (Stevens,

Dress-Steel solution for 3-state Perfect phylogeny(1991)

- Recode each site M(i) of M as three binary sites M'(i,1), M'(i,2), M'(i,3) each indicating the taxa that have state 1, 2, or 3, respectively.
- Theorem (DS):

  There is a 3-state perfect phylogeny for M, if and only if there is a subset S of pairwise compatible columns, such that S contains at least two of the columns

  M'(i,1), M'(i,2), M'(i,3), for each column i of M.

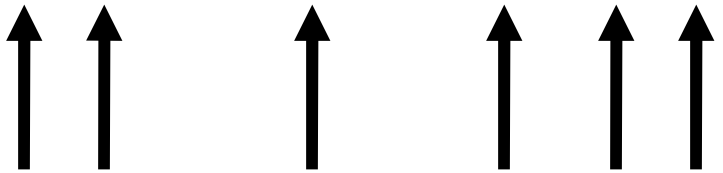
Example

M

	A	B	C
1	3	2	1
2	2	3	2
3	3	2	3
4	1	1	3
5	1	2	3

M’

	A,1	A,2	A,3	B,1	B,2	B,3	C,1	C,2	C,3
1	0	0	1	0	1	0	1	0	0
2	0	1	0	0	0	1	0	1	0
3	0	0	1	0	1	0	0	0	1
4	1	0	0	1	0	0	0	0	1
5	1	0	0	0	1	0	0	0	1



Compatible subset

Solving by 2-SAT

Dress and Steel gave a polytime algorithm to find S,
If it exists, but the problem of finding S can be naturally cast as an instance of 2-SAT.

Create a binary variable for each column of M':
Variables X(i,1), X(i,2), X(i,3) for the three columns in M' that originate from column i in M.

M

	A	B	C
1	3	2	1
2	2	3	2
3	3	2	3
4	1	1	3
5	1	2	3

M’

	A,1	A,2	A,3	B,1	B,2	B,3	C,1	C,2	C,3
1	0	0	1	0	1	0	1	0	0
2	0	1	0	0	0	1	0	1	0
3	0	0	1	0	1	0	0	0	1
4	1	0	0	1	0	0	0	0	1
5	1	0	0	0	1	0	0	0	1
	X(1,1)	X(1,2)	X(1,3)	X(2,1)	X(2,2)	X(2,3)	X(3,1)	X(3,2)	X(3,3)

Binary variables

2-SAT

For every pair of incompatible columns (i,j), (i',j') in M',

Create the clause (not X(i,j) v not X(i',j')) to assure that the columns in S are pairwise compatible.

For every column I in M, create the three clauses:
(X(i,1) v X(i,2))
(X(i,2) v X(i,3))
(X(i,1) v X(i,3))
to assure that S contains at least
two of the three columns in M' that originate from Column i in M.

Shameless Advertisement

Later in the week I will display a partial draft of the book in progress

RECOMBINATORICS: The Structure and Algorithmics of Phylogenetic Newtorks with Recombination

Comments appreciated