Bioinformatics Algorithms

(Fundamental Algorithms, module 2)

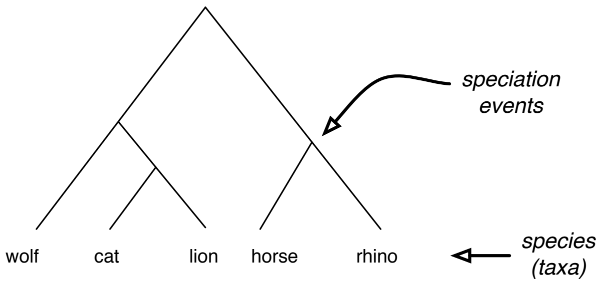
Zsuzsanna Lipták

Masters in Medical Bioinformatics
academic year 2018/19, II. semester

Phylogenetics I[1]

[1] These slides are partially based on the Lecture Notes from Bielefeld University "Algorithms for Phylogenetic Reconstruction" (2016/17), by J. Stoye, R. Wittler, et al.

What is a phylogenetic tree?



Phylogenetic trees display the evolutionary relationships among a set of objects (species). Contemporary species are represented by the leaves. Internal nodes of the tree represent speciation events ($\approx$ common ancestors, usually extinct).

Different types of phylogenetic trees

- rooted vs. unrooted (root on top/bottom vs. root in the middle)
- binary (fully resolved) vs. multifurcating (polytomies)
- are edge lengths significant?
- is there a time scale on the side?

Phylogenetic reconstruction

Goal

Given n objects and data on these objects, find a phylogenetic tree with these objects at the leaves which best reflects the input data.

Phylogenetic reconstruction

Note:

We need to define more precisely

- what kind of input data we have,
- what kind of tree we want (e.g. rooted or unrooted), and
- what we mean by "reflect the data."

Phylogenetic reconstruction

There are two main issues:

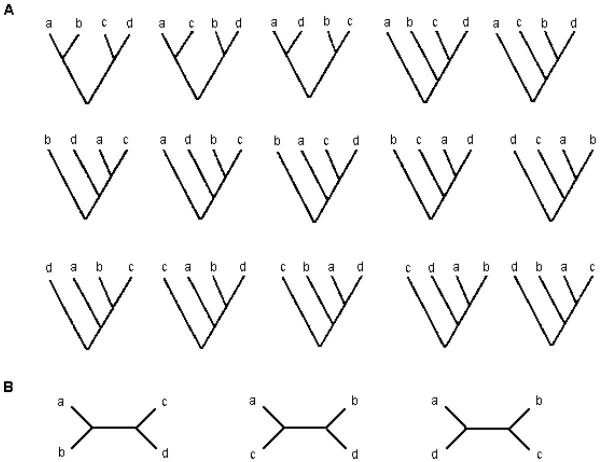
1. How well does a tree reflect my data?
2. How do we find such a tree?

Number of phylogenetic trees

Say we have answered these questions, then: Could we just list all possible trees and then choose the/a best one?

# taxa n	# unrooted trees $(2n-5)!!$	# rooted trees $(2n-3)!!$
1	1	1
2	1	1
3	1	3
4	3	15

Number of phylogenetic trees



All phylogenetic trees (rooted and unrooted) on 4 taxa.

Number of phylogenetic trees

Theorem

There are $U_n = (2n-5)!! = \prod_{i=3}^{n}(2i-5)$ unrooted binary phylogenetic trees on n objects, and $R_n = (2n-3)!! = \prod_{i=2}^{n}(2i-3)$ rooted binary phylogenetic trees on n objects.

Proof

By induction on n, using that (1) we can get every unrooted tree on $n+1$ objects in a unique way by adding the $(n+1)$st leaf to an unrooted tree on the first n objects; (2) an unrooted binary tree with n leaves has $2n-3$ edges, (3) every unrooted tree on n objects can be rooted in (number of edges) ways, yielding a rooted tree on n objects.

Number of phylogenetic trees

#taxa n	#unrooted trees $(2n-5)!!$	#rooted trees $(2n-3)!!$
1	1	1
2	1	1
3	1	3
4	3	15
5	15	105
6	105	945
7	945	10,395
8	10,395	135,135
9	135,135	2,027,025
10	2,027,025	34,459,425

Number of phylogenetic trees

So there are super-exponentially many trees:
We cannot check all of them!

Types of input data

We can have two kinds of input data:

- distance data: $n \times n$ matrix of pairwise distances between the taxa, or
- character data: $n \times m$ matrix giving the states of m characters for the n taxa

Distance data

Distance data is given as an $(n \times n)$ matrix M with the pairwise distances between the taxa.

Ex.

	a	b	c
a	0	5	2
b	5	0	4
c	2	4	0

E.g., $M_{a,b} = 5$ means that the distance between a and b is 5. Often, this is the edit distance (between two genomic sequences, or between homologous proteins, ...).

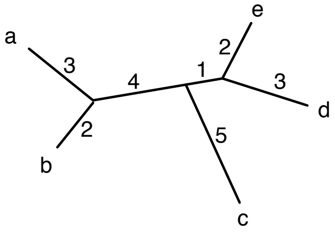
We want to find a tree with a, b, c at the leaves s.t. the distance in the tree (the path metric) between a and b is 5, between a and c is 2, etc.

Distance data

Path metric of a tree

Given a tree T, the path-metric of T is d_T, defined as: $d_T(u, v) =$ sum of edge weights on the (unique) path between u and v.

Example



$$d_T(a, b) = 5,$$
$$d_T(a, d) = 11,$$
$$d_T(c, d) = 9, \ldots$$

Note

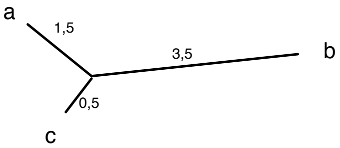
$d_T(u, v)$ is also defined for inner nodes u, v, but we only need it for leaves.

Example

For our earlier example, we can find such a tree:

Ex. 1 (from before)

	a	b	c
a	0	5	2
b	5	0	4
c	2	4	0

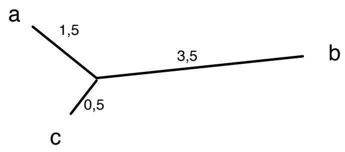


Example

For our earlier example, we can find such a tree:

Ex. 1 (from before)

	a	b	c
a	0	5	2
b	5	0	4
c	2	4	0



Question

Is it always possible to find a tree s.t. its path-metric equals the input distances? I.e. does such a tree exist for any input matrix M?

Distance data

First of all, the input matrix M has to define a metric (= a distance function), i.e. for all x, y, z,

Distance data

First of all, the input matrix M has to define a metric (= a distance function), i.e. for all x, y, z,

- $M(x, y) \geq 0$ and ($M(x, y) = 0$ iff $x = y$) (positive definite)
- $M(x, y) = M(y, x)$ (symmetry)
- $M(x, y) + M(y, z) \geq M(x, z)$ (triangle inequality)

For example, the edit distance is a metric (on strings), the Hamming distance (on strings of the same length), the Euclidean distance (on $\mathbb{R}^2$).

Conditions on distance matrix

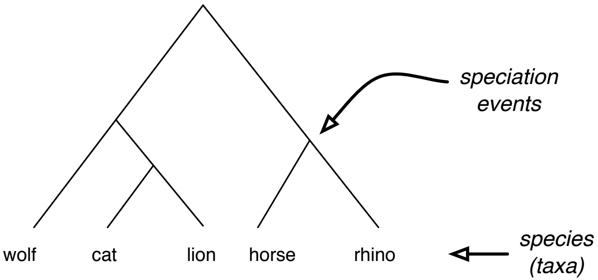
Question:

When does a tree exist whose path metric agrees with a distance matrix M?

Answer:

- if we want a rooted tree: M needs to be ultrametric
- if we want an unrooted tree: M needs to be additive

Rooted trees and the molecular clock



In a rooted phylogenetic tree, the molecular clock assumption holds: that the speed of evolution is the same along all branches, i.e. the path distance from each leaf to the root is the same. Such a tree is also called an ultrametric tree.

Ultrametrics and the three-point condition

Three point condition

Let d be a metric on a set of objects O, then d is an ultrametric if $\forall\ x, y, z \in O$:

$$d(x, y) \leq \max\{d(x, z), d(z, y)\}$$

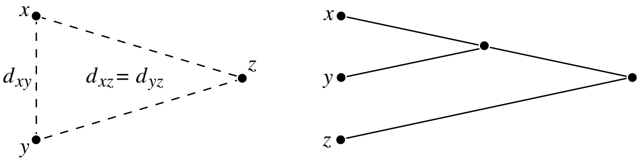


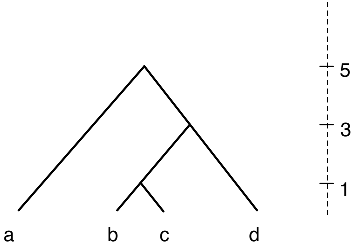
Figure: Three point condition. It implies that the path metric of a rooted tree is an ultrametric.

In other words, among the three distances, there is no unique maximum.

Example

Ex. 2

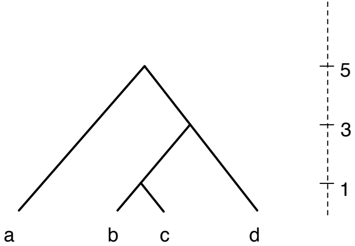
	a	b	c	d
a	0	10	10	10
b	10	0	2	6
c	10	2	0	6
d	10	6	6	0



Example

Ex. 2

	a	b	c	d
a	0	10	10	10
b	10	0	2	6
c	10	2	0	6
d	10	6	6	0



Checking the ultrametric condition, we see that:

- for a, b, c we get $2, 10, 10$ — okay
- for a, b, d we get $6, 10, 10$ — okay
- for a, c, d we get $6, 10, 10$ — okay
- for b, c, d we get $2, 6, 6$ — okay

Example

Compare this to our earlier example. There the matrix M does not define an ultrametric!

Ex. 1 (from before)

	a	b	c
a	0	5	2
b	5	0	4
c	2	4	0

For the triple a, b, c (the only triple), we get: $2, 4, 5$, and there is a unique maximum: 5.

Example

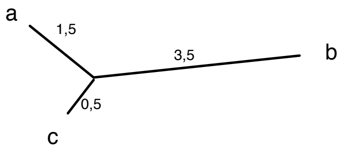
Compare this to our earlier example. There the matrix M does not define an ultrametric!

Ex. 1 (from before)

	a	b	c
a	0	5	2
b	5	0	4
c	2	4	0

For the triple a, b, c (the only triple), we get: $2, 4, 5$, and there is a unique maximum: 5.

Indeed, the only tree we found was not rooted:



Ultrametrics and the three-point condition

Theorem

Given an $(n \times n)$ distance matrix M. There is a rooted tree whose path metric agrees with M if and only if M defines an ultrametric (i.e. if and only if it is a metric and the 3-point-condition holds). This tree is unique[2].

[2] i.e. there is only one such tree

Ultrametrics and the three-point condition

Theorem

Given an $(n \times n)$ distance matrix M. There is a rooted tree whose path metric agrees with M if and only if M defines an ultrametric (i.e. if and only if it is a metric and the 3-point-condition holds). This tree is unique[2].

Algorithm

The algorithm UPGMA (*unweighted pair group mtheod using arithmetic averages*, Michener & Sokal 1957), a hierarchical clustering algorithm, constructs this tree, given an input matrix which is ultrametric. Its running time is $O(n^2)$.

[2] i.e. there is only one such tree

Additive metrics and the four-point condition

So what is the condition on the matrix M for unrooted trees?

Four point condition.

Let d be a metric on a set of objects O, then d is an additive metric if $\forall\ x, y, u, v \in O$:

$$d(x,y) + d(u,v) \leq \max\{d(x,u) + d(y,v), d(x,v) + d(y,u)\}$$

In other words, among the three sums of two distances, there is no unique maximum.

Additive metrics and the four-point condition

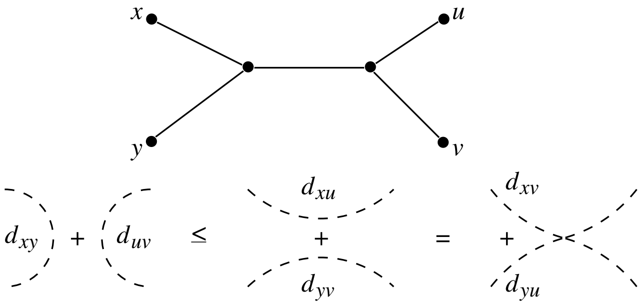
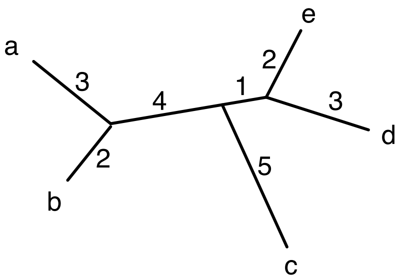


Figure: The four point condition. It implies that the path metric of a tree is an additive metric.

Example



For ex., choose these 4 points: a, b, c, e. Then we get the three sums: $d(a, b) + d(c, e) = 5 + 8 = 13$, $d(a, c) + d(b, e) = 12 + 9 = 21$, and $d(a, e) + d(b, c) = 10 + 11 = 21$. Among $13, 21, 21$, there is no unique maximum—okay. (Careful, this has to hold for all quadruples; how many are there?)

Additive metrics and the four-point condition

Theorem

Given an $(n \times n)$ distance matrix M. There is an unrooted tree whose path metric agrees with M if and only if M defines an additive metric (i.e. if and only if it is a metric and the 4-point-condition holds). This tree is unique.

Algorithm

The algorithm NJ (Neighbor Joining) constructs this tree, given an additive matrix M (Saitu & Nei, 1987). Its running time is $O(n^3)$.

Additive metrics and the four-point condition

Theorem

Given an $(n \times n)$ distance matrix M. There is an unrooted tree whose path metric agrees with M if and only if M defines an additive metric (i.e. if and only if it is a metric and the 4-point-condition holds). This tree is unique.

Algorithm

The algorithm NJ (Neighbor Joining) constructs this tree, given an additive matrix M (Saitu & Nei, 1987). Its running time is $O(n^3)$.

In fact, it is even possible to compute a "good" tree if the matrix is not additive but "almost" *(all this needs to be defined precisely, of course).*

Summary for distance data

- When the input is a distance matrix, then we are looking for a tree whose path metric agrees with M.

Summary for distance data

- When the input is a distance matrix, then we are looking for a tree whose path metric agrees with M.
- A rooted tree agreeing with M exists if and only if the distance matrix M defines an ultrametric.

Summary for distance data

- When the input is a distance matrix, then we are looking for a tree whose path metric agrees with M.
- A rooted tree agreeing with M exists if and only if the distance matrix M defines an ultrametric.
- This tree can then be computed efficiently (i.e. in polynomial time), with UPGMA.

Summary for distance data

- When the input is a distance matrix, then we are looking for a tree whose path metric agrees with M.
- A rooted tree agreeing with M exists if and only if the distance matrix M defines an ultrametric.
- This tree can then be computed efficiently (i.e. in polynomial time), with UPGMA.
- An unrooted tree agreeing with M exists if and only if the distance matrix M defines an additive metric.

Summary for distance data

- When the input is a distance matrix, then we are looking for a tree whose path metric agrees with M.
- A rooted tree agreeing with M exists if and only if the distance matrix M defines an ultrametric.
- This tree can then be computed efficiently (i.e. in polynomial time), with UPGMA.
- An unrooted tree agreeing with M exists if and only if the distance matrix M defines an additive metric.
- It can be computed efficiently (i.e. in polynomial time), with Neighbor Joining.