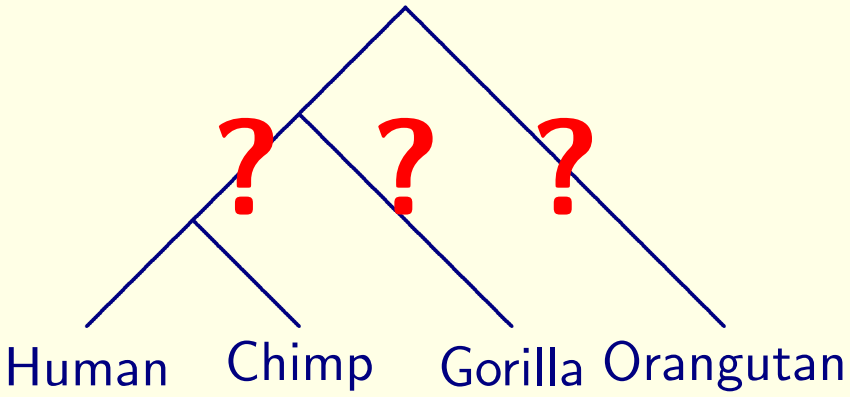
Fast Neighbor Joining

Isaac Elias **Jens Lagergren**

Royal Institute of Technology

Sweden

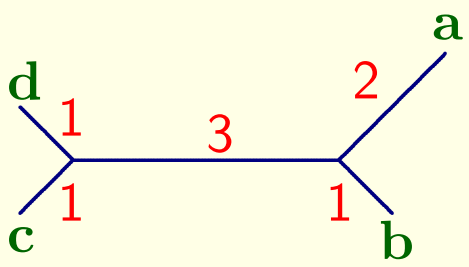
Evolutionary History



- Distance methods
- Parsimony methods
- ML methods

Tree Reconstruction Problem

Tree T



Additive Metric

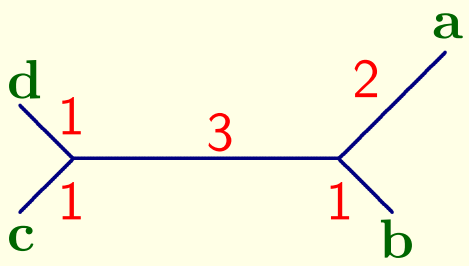
$D_T(x,y) = \sum_{e \in \text{path}(x,y)} l(e)$

$D_T =$

	a	b	c	d
a	0	3	6	6
b		0	5	5
c			0	2
d				0

Tree Reconstruction Problem

Tree T



Additive Metric

$D_T(x,y) = \sum_{e \in \text{path}(x,y)} l(e)$

$D_T =$

	a	b	c	d
a	0	3	6	6
b		0	5	5
c			0	2
d				0

Input A non-additive metric D.

Output Tree S, without edge lengths, that is closest to D,

$$\min_{D_S} |D_S - D|_\infty.$$

$D =$

	a	b	c	d
a	0	3	5	6
b		0	4	5
c			0	1
d				0

The Mighty Error Correcting Code

1. G*d is sending us the message T.

2. He has written down D_T.

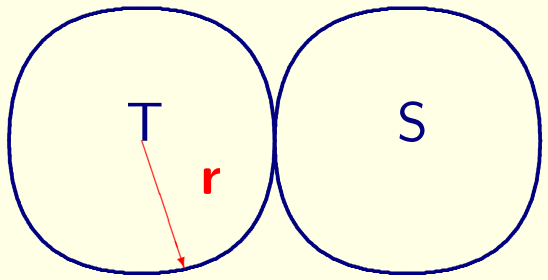
3. D_T changes atmost r.

$$D_T \rightsquigarrow D \implies |D_T - D|_\infty < r$$

4. Find the closest tree S.

$$D_S = \text{argmin}_{D_S} |D_S - D|_\infty$$

How big can r be such that $T = S$?



Optimal Reconstruction Radius [Atteson]

$\mu(T) =$ shortest edge length in T.

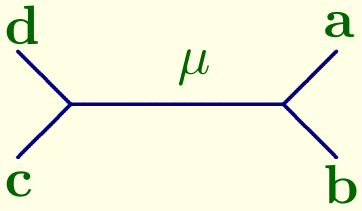
1. If $r < \frac{\mu(T)}{2}$ then $S = T$ (D is nearly additive).

2. If $r \geq \frac{\mu(T)}{2}$ then it can be that $S \neq T$.

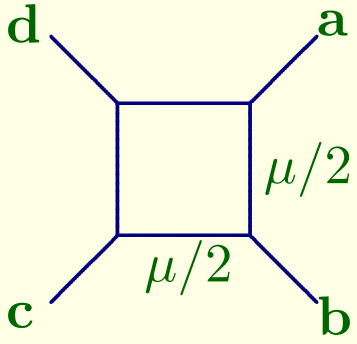
No algorithm can have reconstruction radius $> \frac{\mu(T)}{2}$.

Upper Bound on Reconstruction Radius [Atteson]

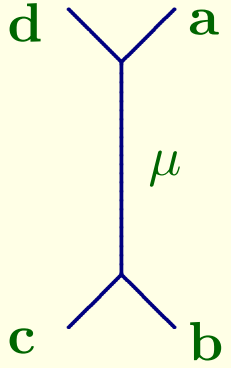
Tree T



Input distances D



Tree S



$$|D_T - D|_\infty = \mu/2 \qquad |D_S - D|_\infty = \mu/2$$

NJ and FNJ has Optimal Reconstruction Radius

$\mu(T) =$ shortest edge length in T.

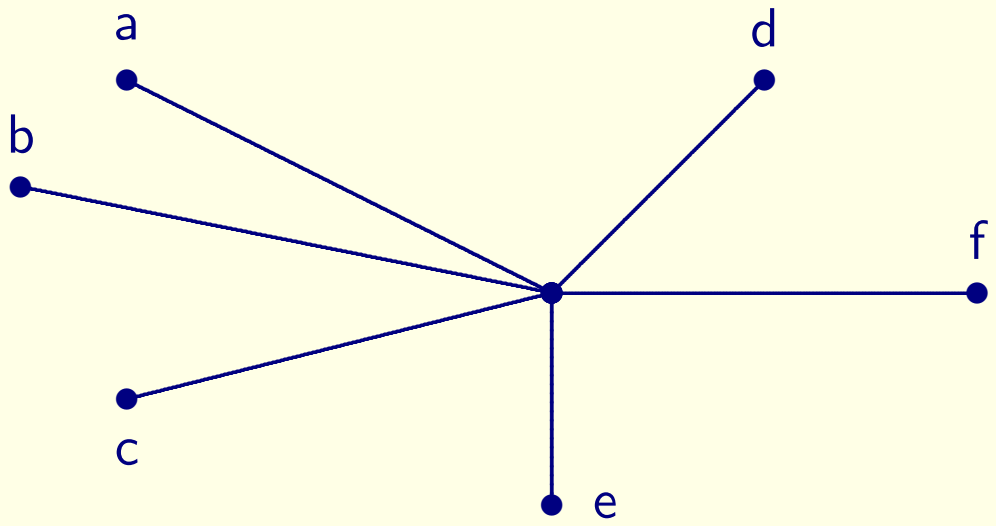
1. If $r < \frac{\mu(T)}{2}$ then $S = T$ (D is nearly additive).
2. If $r \geq \frac{\mu(T)}{2}$ then it can be that $S \neq T$.

No algorithm can have reconstruction radius $> \frac{\mu(T)}{2}$.

	Time	Radius	Our contribution
NJ	$O(n^3)$	$\frac{\mu(T)}{2}$	simplify the proof
FNJ	$O(n^2)$	$\frac{\mu(T)}{2}$	new fast algorithm

Iterative Clustering

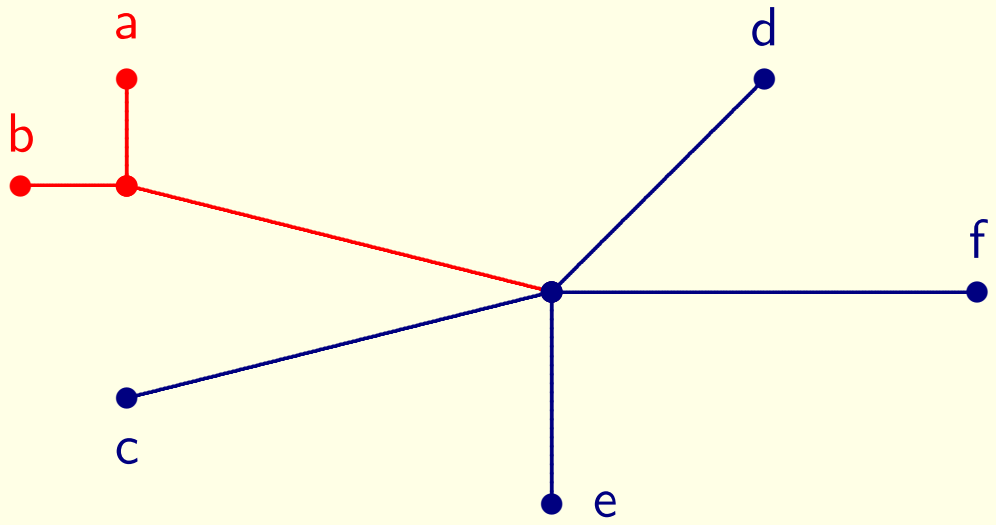
Unresolved



$\mathbf{n = 6}$

Iterative Clustering

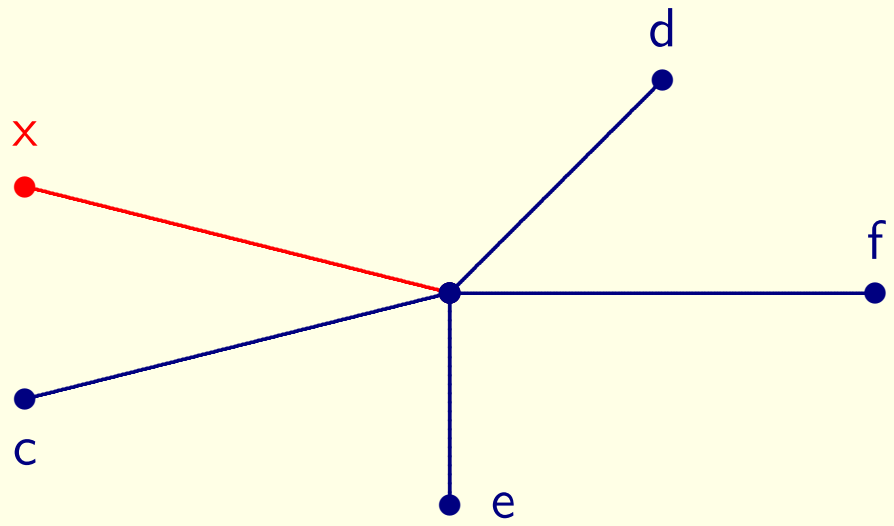
Cluster - find two siblings



$\mathbf{n = 6}$

Iterative Clustering

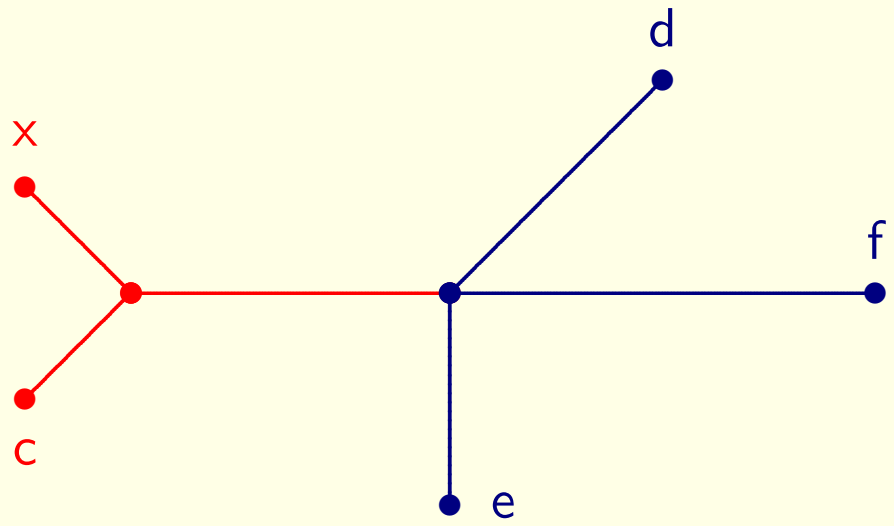
Reduce - replace by parent



$\mathbf{n = 5}$

Iterative Clustering

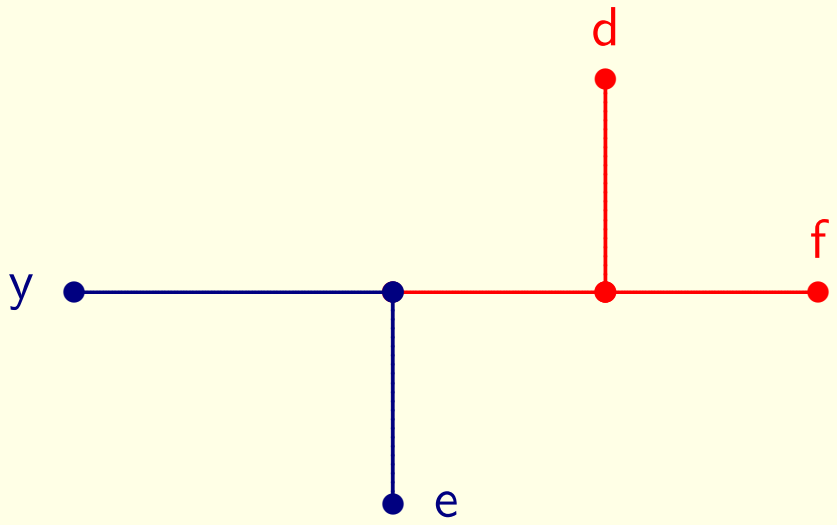
Cluster and Reduce



$\mathbf{n = 5}$

Iterative Clustering

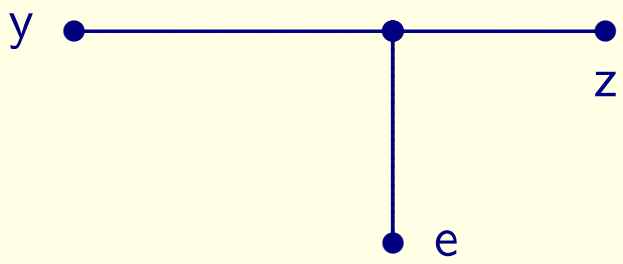
Cluster and Reduce



$\mathbf{n = 4}$

Iterative Clustering

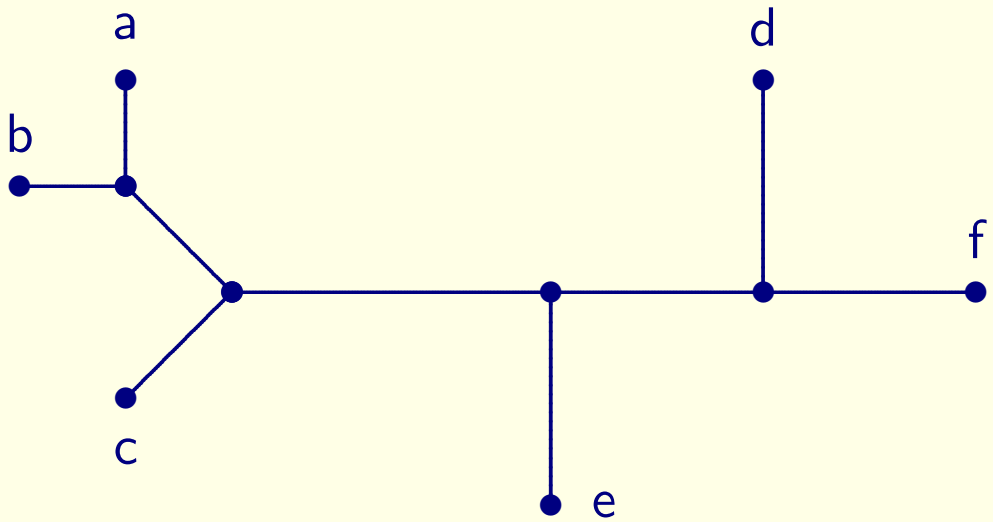
Three leafs



$\mathbf{n = 3}$

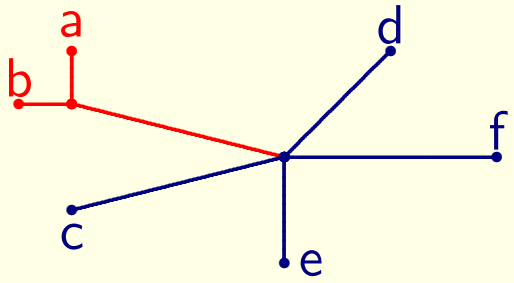
Iterative Clustering

Resolved



Neighbor Joining [Saitou,Nei]

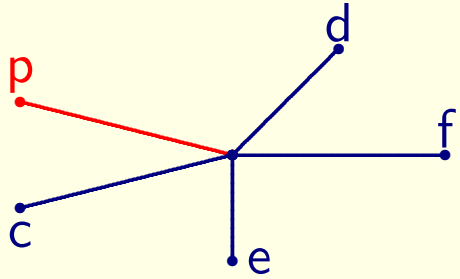
Clustering - $O(n^2)$



(a, b) is the pair minimizing

$S_D(x, y) \triangleq (n-2)D(x, y) - \sum_z (D(z, x) + D(z, y))$

Reduction - $O(n)$



Replace (a, b) by p

$D(p, x) \triangleq \frac{D(a,x)+D(b,x)}{2}$

Total time (n-3) iterations - $\mathbf{O(n^3)}$

Fast Neighbor Joining

NJ

$(a, b) \leftarrow \text{argmin}_{(x,y)} S_D(x, y)$

$D(p, x) = \frac{D(a,x)+D(b,x)}{2}$

FNJ

$(a, b) \leftarrow \text{argmin}_{(x,y)\in \mathbf{V}} S_D(x, y)$

$D(p, x) = \frac{D(a,x)+D(b,x)}{2}$

The minimal pair is selected from the visible set V of size $O(n)$.

	Time	**Radius**
NJ	$O(n^3)$	$\frac{\mu(T)}{2}$
FNJ	$O(n^2)$	$\frac{\mu(T)}{2}$

FNJ - Detailed

FNJ(D)

1. For each node a add $(a, b) \leftarrow {}_{\text{argmin}(a,y)}\ S_D(a, y)$ to V

2. For each $i \leftarrow 1$ to $n - 3$ do

   (a) $(a, b) \leftarrow \text{argmin}_{(x,y)\in V} S_D(x, y)$

   (b) Reduce $(a, b) \rightarrow p$ using $D(p, x) = (D(a, x) + D(b, x))/2$

   (c) Add $(p, b) \leftarrow \text{argmin}_{(p,y)}\ S_D(p, y)$ to V

The Proof

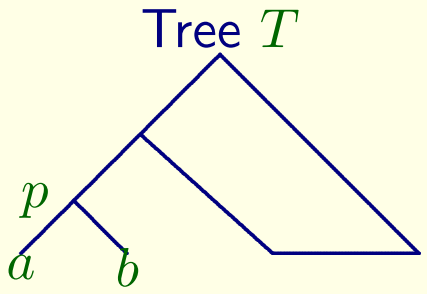
$$|D_T - D|_\infty < \frac{\mu(T)}{2} \quad \Longrightarrow \quad \mathrm{FNJ}(D) = T$$

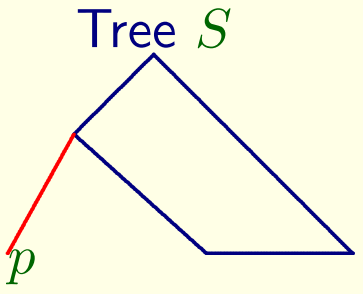
We prove by induction

1. $(a, b) \leftarrow \mathrm{argmin}_{(x,y)\in V}\, S_D(x, y) \implies (a, b)$ are siblings in T

2. After reducing a sibling pair $(a, b) \to p$ the matrix D is nearly additive to a tree $S = T \setminus \{a, b\}$.

Reduction Correct





$$|D - D_T|_\infty < \mu(T)/2$$

Reduction step

$$D(p,x) \triangleq \frac{D(a,x) + D(b,x)}{2}$$

Show that

$$|D - D_S|_\infty < \mu(S)/2$$

Proof Sketch

1. For each node a add $(a, b) \leftarrow \text{argmin}_{(a,y)}\ S_D(a, y)$ to V **Part 1**
2. For each $i \leftarrow 1$ to $n - 3$ do
   (a) $(a, b) \leftarrow \text{argmin}_{(x,y)\in V} S_D(x, y)$ **Part 2**
   (b) Reduce $(a, b) \rightarrow p$ using $D(p, x) = (D(a, x)+D(b, x))/2$
   (c) Add $(p, b) \leftarrow \text{argmin}_{(p,y)}\ S_D(p, y)$ to V **Part 1**

Part 1 If a has sibling b then $(a, b) \leftarrow \text{argmin}_{(a,x)\in V}\ S_D(a, x)$.

$$\Longrightarrow V \text{ contains all sibling pairs}$$

Part 2 If (c, d) is not a sibling pair $\Longrightarrow \exists (a, b)$ s.t. $S_D(a, b) < S_D(c, d)$.

$$\Longrightarrow \text{the minium over } V \text{ is a sibling pair}$$

The Additive Case

I will only show the additive case,

$$\mathrm{FNJ}(\mathbf{D_T}) = \mathbf{T}$$

The Additive Case

$$D_T(x,y) = \sum_{e \in \mathrm{path}(x,y)} l(e)$$

$$S_D(x,y) \triangleq (n-2)D(x,y) - \textstyle\sum_z (D(z,x) + D(z,y))$$

$$S_{D_T}(x,y) = \sum_{e \in E(T)} \mathbf{w_e(x,y)}\, l(e), \text{ where}$$

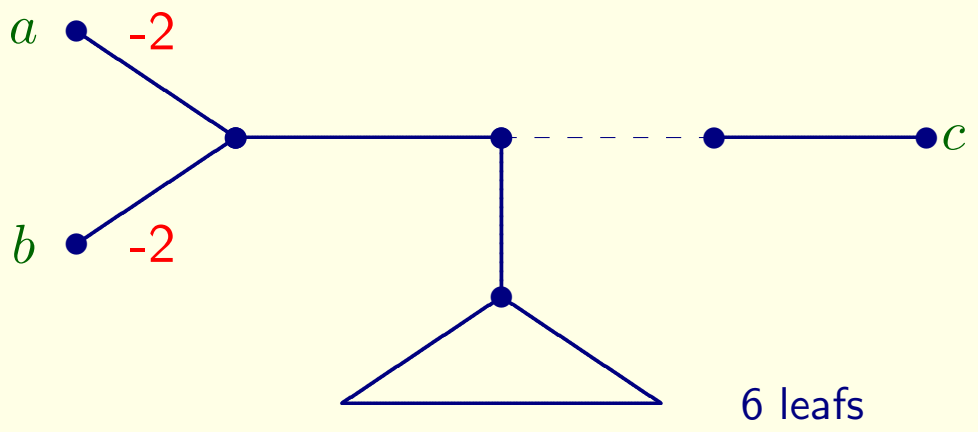
$$w_e(x,y) = \begin{cases} -2 & \text{if } e \in \mathrm{path}(x,y) \\ -2|L(T) \setminus \mathcal{L}_T(x,e)| & \text{otherwise.} \end{cases}$$

Part 1. The Additive Case (cont.)

$$S_{D_T}(x,y) = \sum_{e \in E(T)} \mathbf{w_e(x,y)}\, l(e), \text{ where}$$

$$w_e(x,y) = \begin{cases} -2 & \text{if } e \in \text{path}(x,y) \\ -2|L(T) \setminus \mathcal{L}_T(x,e)| & \text{otherwise.} \end{cases}$$

$S_{D_T}(a,b)$

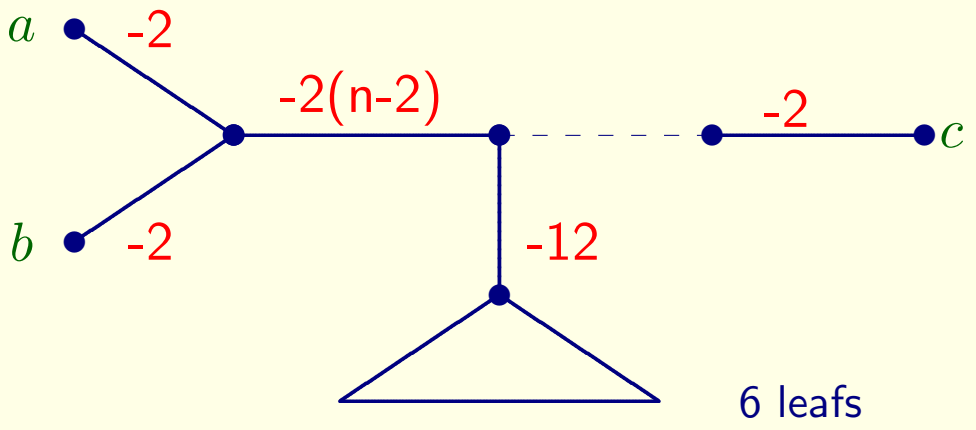


Part 1. The Additive Case (cont.)

$$S_{D_T}(x,y) = \sum_{e \in E(T)} \mathbf{w_e(x,y)}\, l(e), \text{ where}$$

$$w_e(x,y) = \begin{cases} -2 & \text{if } e \in \text{path}(x,y) \\ -2|L(T) \setminus \mathcal{L}_T(x,e)| & \text{otherwise.} \end{cases}$$

$S_{D_T}(a,b)$



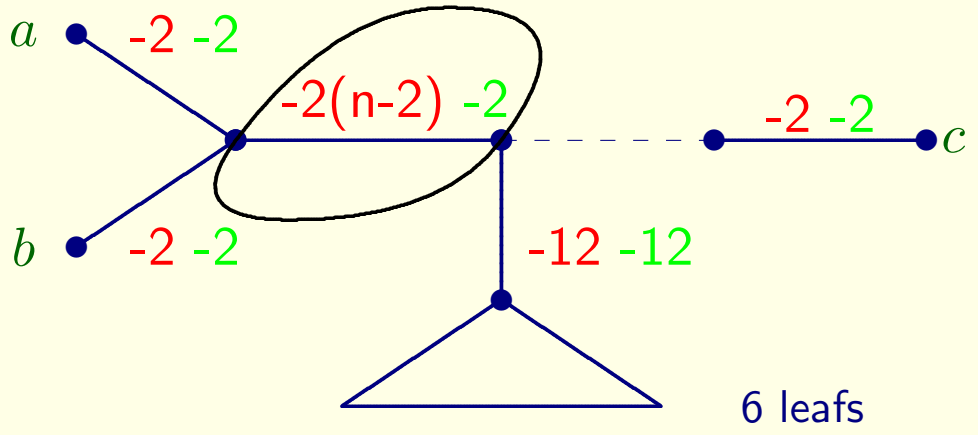
Part 1. The Additive Case (cont.)

$$S_{D_T}(x,y) = \sum_{e \in E(T)} \mathbf{w_e(x,y)}\, l(e), \text{ where}$$

$$w_e(x,y) = \begin{cases} -2 & \text{if } e \in \text{path}(x,y) \\ -2|L(T) \setminus \mathcal{L}_T(x,e)| & \text{otherwise.} \end{cases}$$

$S_{D_T}(a,b)$

$S_{D_T}(a,c)$



$$\mathbf{S_{D_T}(a,c) - S_{D_T}(a,b) > 2(n-3)\mu(T)}$$

Proof Sketch (cont.)

1. For each node a add $(a, b) \leftarrow \text{argmin}_{(a,y)}\ S_D(a, y)$ to V **Part 1**
2. For each $i \leftarrow 1$ to $n - 3$ do
   (a) $(a, b) \leftarrow \text{argmin}_{(x,y)\in V} S_D(x, y)$ **Part 2**
   (b) Reduce $(a, b) \rightarrow p$ using $D(p, x) = (D(a, x)+D(b, x))/2$
   (c) Add $(p, b) \leftarrow \text{argmin}_{(p,y)}\ S_D(p, y)$ to V **Part 1**

Part 1 If a has sibling b then $(a, b) \leftarrow \text{argmin}_{(a,x)\in V}\ S_D(a, x)$.

$\Longrightarrow V$ contains all sibling pairs

Part 2 If (x, y) is not a sibling pair $\Longrightarrow \exists (a, b)$ s.t. $S_D(a, b) < S_D(x, y)$.

$\Longrightarrow$ the minium over V is a sibling pair

Proof Sketch (cont.)

1. For each node a add $(a, b) \leftarrow \text{argmin}_{(a,y)}\ S_D(a, y)$ to V **Part 1**
2. For each $i \leftarrow 1$ to $n - 3$ do
   (a) $(a, b) \leftarrow \text{argmin}_{(x,y)\in V} S_D(x, y)$ **Part 2**
   (b) Reduce $(a, b) \rightarrow p$ using $D(p, x) = (D(a, x)+D(b, x))/2$
   (c) Add $(p, b) \leftarrow \text{argmin}_{(p,y)}\ S_D(p, y)$ to V **Part 1**

Part 1 If a has sibling b then $(a, b) \leftarrow \text{argmin}_{(x,y)\in V}\ S_D(x, y)$.

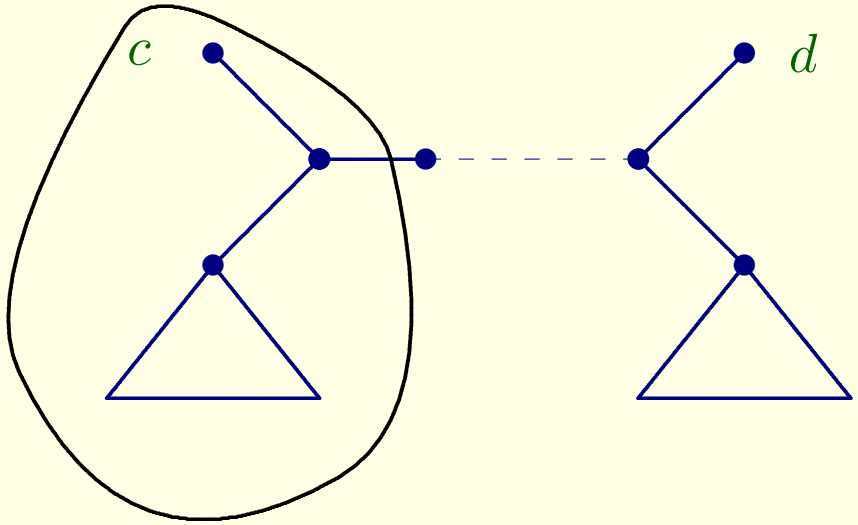
$\Longrightarrow V$ contains all sibling pairs

Part 2 If (c, d) is not a sibling pair $\Longrightarrow \exists (a, b)$ s.t. $S_D(a, b) < S_D(c, d)$.

$\Longrightarrow$ the minium over V is a sibling pair

Part 2. The Additive Case

If (c, d) is not a sibling pair $\Longrightarrow \exists (a, b)$ s.t. $S_{D_T}(a, b) < S_{D_T}(c, d)$.



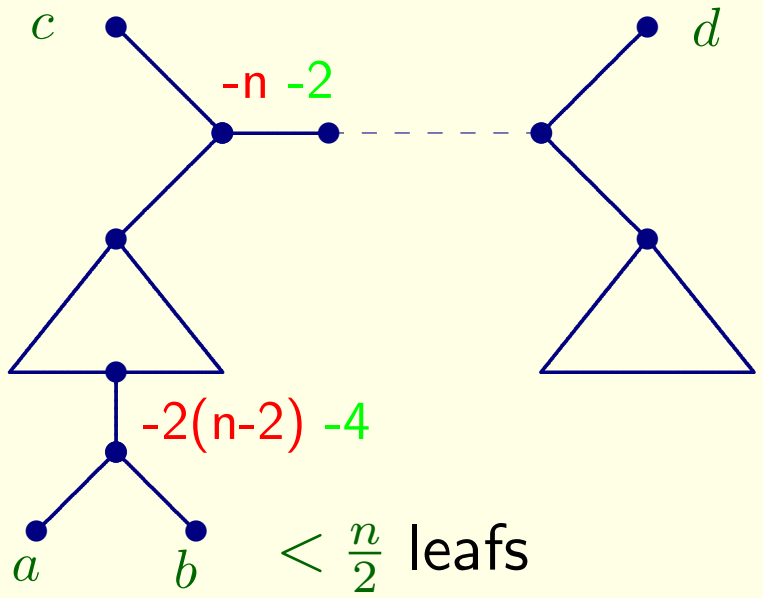
$< \frac{n}{2}$ leafs

Part 2. The Additive Case

If (c, d) is not a sibling pair $\Longrightarrow \exists (a, b)$ s.t. $S_{D_T}(a, b) < S_{D_T}(c, d)$.

$S_{D_T}(a, b)$

$S_{D_T}(c, d)$



$$\mathbf{S_{D_T}(c, d) - S_{D_T}(a, b) > 3(n - 4)\mu(T)}$$

Theory vs. Practice

Theory vs. Practice

Many algorithms have good theoretical properties.

Quartet methods are better than NJ like methods in theory but not in practice.

Reconstruction Radius - the whole tree is guaranteed to be correctly reconstructed.

Theory vs. Practice

Many algorithms have good theoretical properties.

Quartet methods are better than NJ like methods in theory but not in practice.

Reconstruction Radius - the whole tree is guaranteed to be correctly reconstructed.

Edge Radius - long edges that are guaranteed to be correctly reconstructed.

A method has edge radius α if it reconstructs all edges $|D - D_T| < \alpha \cdot l_e$.

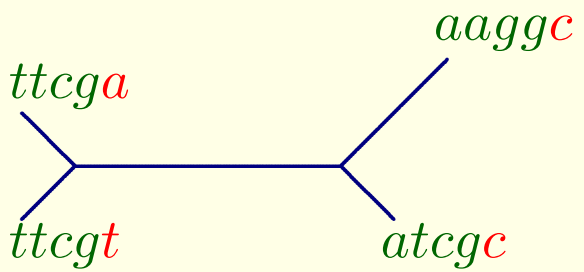
E.g. $\alpha = 1/2$ all edges which are longer than $2 \cdot |D - D_T|$ are correctly reconstructed.

Overview

	Time	Radius	Edge Radius
NJ	$O(n^3)$	$\frac{1}{2}$	$\frac{1}{4}$
BioNJ	$O(n^3)$	$\frac{1}{2}$	$\frac{1}{4}$
FNJ	$O(n^2)$	$\frac{1}{2}$	$\frac{1}{4}$
ADDTREE	$O(n^4)$	$\frac{1}{2}$	$\frac{1}{2}$
Buneman	$O(n^3)$	$\frac{1}{2}$	$\frac{1}{2}$
DLCA	$O(n^2)$	$\frac{1}{2}$	$\frac{1}{2}$

Biological Background

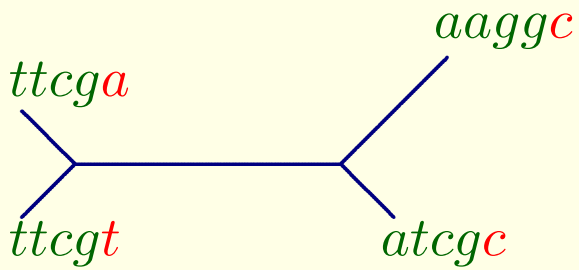
1. Genomic sequences from unkown tree.
2. Assume probabilistic model of evolution.
3. Estimate pairwise distances.
4. Use pairwise distances to build tree.



DNA sequences $\longrightarrow$ Estimated Distance Matrix $\longrightarrow$ Tree

Biological Background

1. Genomic sequences from unkown tree.
2. Assume probabilistic model of evolution.
3. Estimate pairwise distances.
4. Use pairwise distances to build tree.



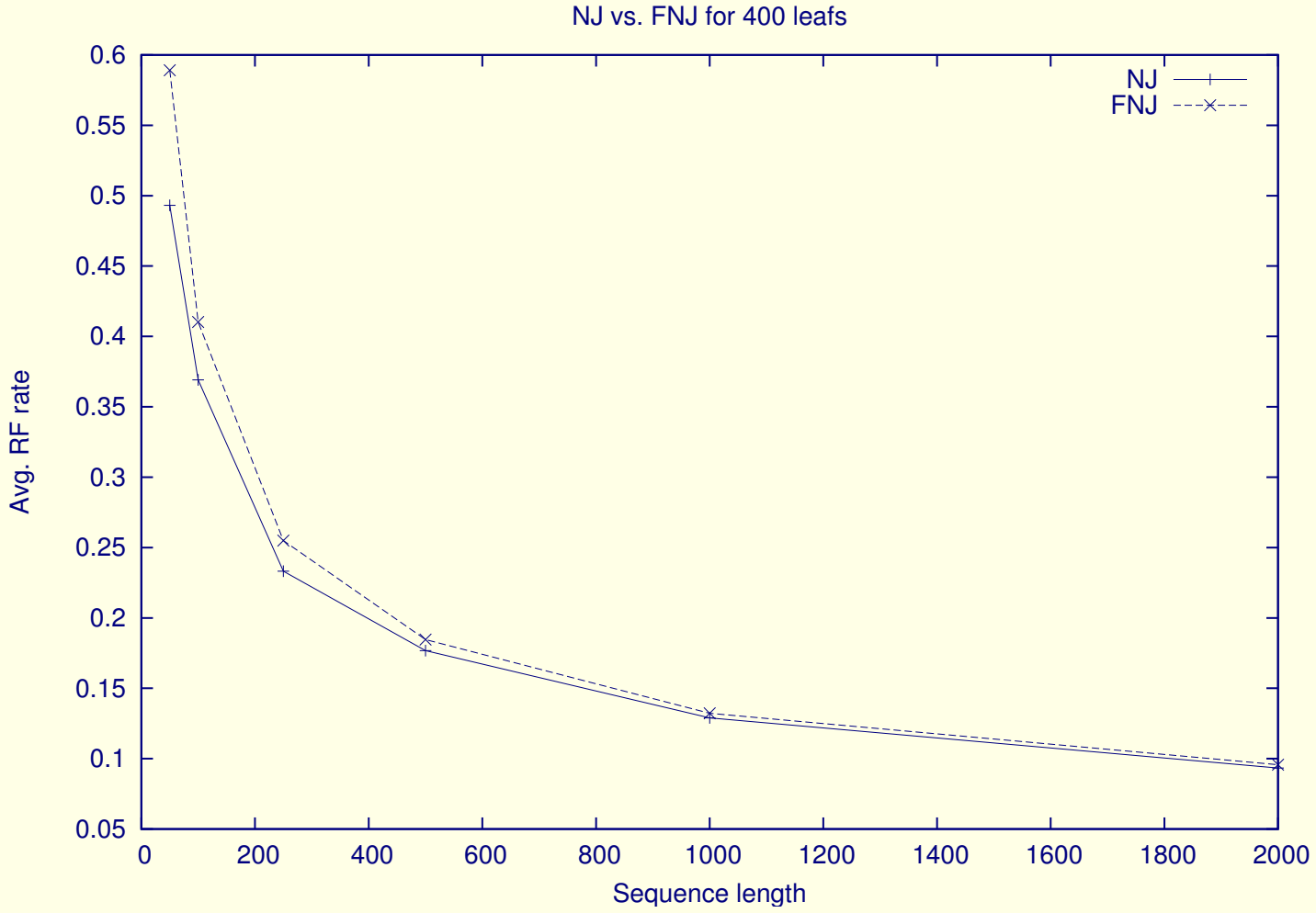
DNA sequences $\longrightarrow$ Estimated Distance Matrix $\longrightarrow$ Tree

Consistent long sequences $\longrightarrow$ good estimates $\longrightarrow$ correct tree

Estimation Accuracy

1. Build random tree T.
2. Model sequence evolution in the tree.
3. Compute distance matrix D from the sampled sequences.
4. Compute tree $NJ(D) = S$
5. Measure Robinson-Foulds distance between S and T.

In Practice


NJ vs. FNJ for 400 leafs
Avg. RF rate
Sequence length
NJ
FNJ
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
200
400
600
800
1000
1200
1400
1600
1800
2000

Convergence Rate

How long sequences are need to with high probability reconstruct the correct tree?

NJ requires exponentially long sequences.

We would like to have a method that reconstructs the tree from short sequences!

Convergence Rate

How long sequences are need to with high probability reconstruct the correct tree?

NJ requires exponentially long sequences.

We would like to have a method that reconstructs the tree from short sequences!

DCM uses NJ as a subrutine and has polynomial convergence rate.

DCM+NJ has running time $O(n^5)$.

DCM+FNJ has running time $O(n^4)$.

Other Interesting Results

Daskalakis et al. Optimal Phylogenetic Reconstruction.
If the mutation probability $p < 0.146$ on all edges of the tree, then the tree can be recovered from sequences of length $O(\log n)$. The algorithm reconstructs ancestral sequences.

Mihaescu et al. Why neighbor-joining works.
NJ and FNJ reconstructs the correct tree if the input matrix is *quartet consistent* and *quartet additive*. NJ and FNJ have edge-radius $1/4$.

Gronau et al. Pivotal Neighbor Joining Algorithms for Inferrring Phylogenies via LCA-Distances.
An $O(n^2)$ algorithm with edge-radius $1/2$. The algorithm is also a 3-approximation under L_∞.

Acknowledgments

Dr. Luay Nakhleh

and

Prof. Tandy Warnow

Thanks!

Part 1. The Nearly Additive Case

D nearly additive distance matrix and (a, b) sibling pair then

$$\forall c \neq b \;\; S_D(a,c) - S_D(a,b) \;\; > \;\; 0$$

$$S_D(a,c) - S_{D_T}(a,c) + S_{D_T}(a,b) - S_D(a,b) \;\; > \;\; -S_{D_T}(a,c) + S_{D_T}(a,b)$$

Part 1. The Nearly Additive Case

D nearly additive distance matrix and (a, b) sibling pair then

$$\forall c \neq b \;\; S_D(a, c) - S_D(a, b) \quad > \quad 0$$

$$S_D(a, c) - S_{D_T}(a, c) + S_{D_T}(a, b) - S_D(a, b) \quad > \quad -2(n-3)\mu(T)$$

Part 1. The Nearly Additive Case

D nearly additive distance matrix and (a, b) sibling pair then

$$\forall c \neq b \;\; S_D(a, c) - S_D(a, b) \quad > \quad 0$$

$$S_D(a, c) - S_{D_T}(a, c) + S_{D_T}(a, b) - S_D(a, b) \quad > \quad -2(n-3)\mu(T)$$

Let $D(i, j) - D_T(i, j) = \varepsilon_{i,j}$ and bound the right hand side

$$S_D(x, y) \triangleq (n-2)D(x, y) - \sum_z (D(z, x) + D(z, y))$$

Part 1. The Nearly Additive Case

D nearly additive distance matrix and (a, b) sibling pair then

$$\forall c \neq b \;\; S_D(a, c) - S_D(a, b) \quad > \quad 0$$

$$S_D(a, c) - S_{D_T}(a, c) + S_{D_T}(a, b) - S_D(a, b) \quad > \quad -2(n-3)\mu(T)$$

Let $D(i, j) - D_T(i, j) = \varepsilon_{i,j}$ and bound the right hand side

$$(n-2)(\varepsilon_{a,c} - \varepsilon_{a,b}) - \sum_m (\varepsilon_{a,m} + \varepsilon_{c,m} - \varepsilon_{a,m} - \varepsilon_{b,m})$$

$$> -(n-2)(\mu/2 + \mu/2) - \sum_m (\mu/2 + \mu/2) > -2(n-3)\mu(T)$$

Part 2. The Nearly Additive Case

If D nearly additive and (c, d) is not a sibling pair $\Longrightarrow \exists (a, b)$ s.t.

$$S_D(c, d) - S_D(a, b) \quad > \quad 0$$

$$S_D(c, d) - S_{D_T}(c, d) + S_{D_T}(a, b) - S_D(a, b) \quad > \quad -S_{D_T}(c, d) + S_{D_T}(a, b)$$

Part 2. The Nearly Additive Case

If D nearly additive and (c, d) is not a sibling pair $\Longrightarrow \exists (a, b)$ s.t.

$$S_D(c, d) - S_D(a, b) \quad > \quad 0$$

$$S_D(c, d) - S_{D_T}(c, d) + S_{D_T}(a, b) - S_D(a, b) \quad > \quad -3(n-4)\mu(T)$$

Part 2. The Nearly Additive Case

If D nearly additive and (c,d) is not a sibling pair $\Longrightarrow \exists (a,b)$ s.t.

$$S_D(c,d) - S_D(a,b) \quad > \quad 0$$

$$S_D(c,d) - S_{D_T}(c,d) + S_{D_T}(a,b) - S_D(a,b) \quad > \quad -3(n-4)\mu(T)$$

Let $D(i,j) - D_T(i,j) = \varepsilon_{i,j}$ and bound right hand side

$$(n-2)(\varepsilon_{c,d} - \varepsilon_{a,b}) - \sum_m (\varepsilon_{c,m} + \varepsilon_{d,m} - \varepsilon_{a,m} - \varepsilon_{b,m})$$

$$> -(n-2)(\mu/2 + \mu/2) - \sum_m (\mu/2 + \mu/2 + \mu/2 + \mu/2) > -3(n-4)\mu(T)$$