Computing Triplet and Quartet Distances Between Trees

Gerth Stølting Brodal, Morten Kragelund Holt, Jens Johansen
Aarhus University

Rolf Fagerberg
University of Southern Denmark

Thomas Mailund, Christian N. S. Pedersen, Andreas Sand
Aarhus University, Bioinformatics Research Center

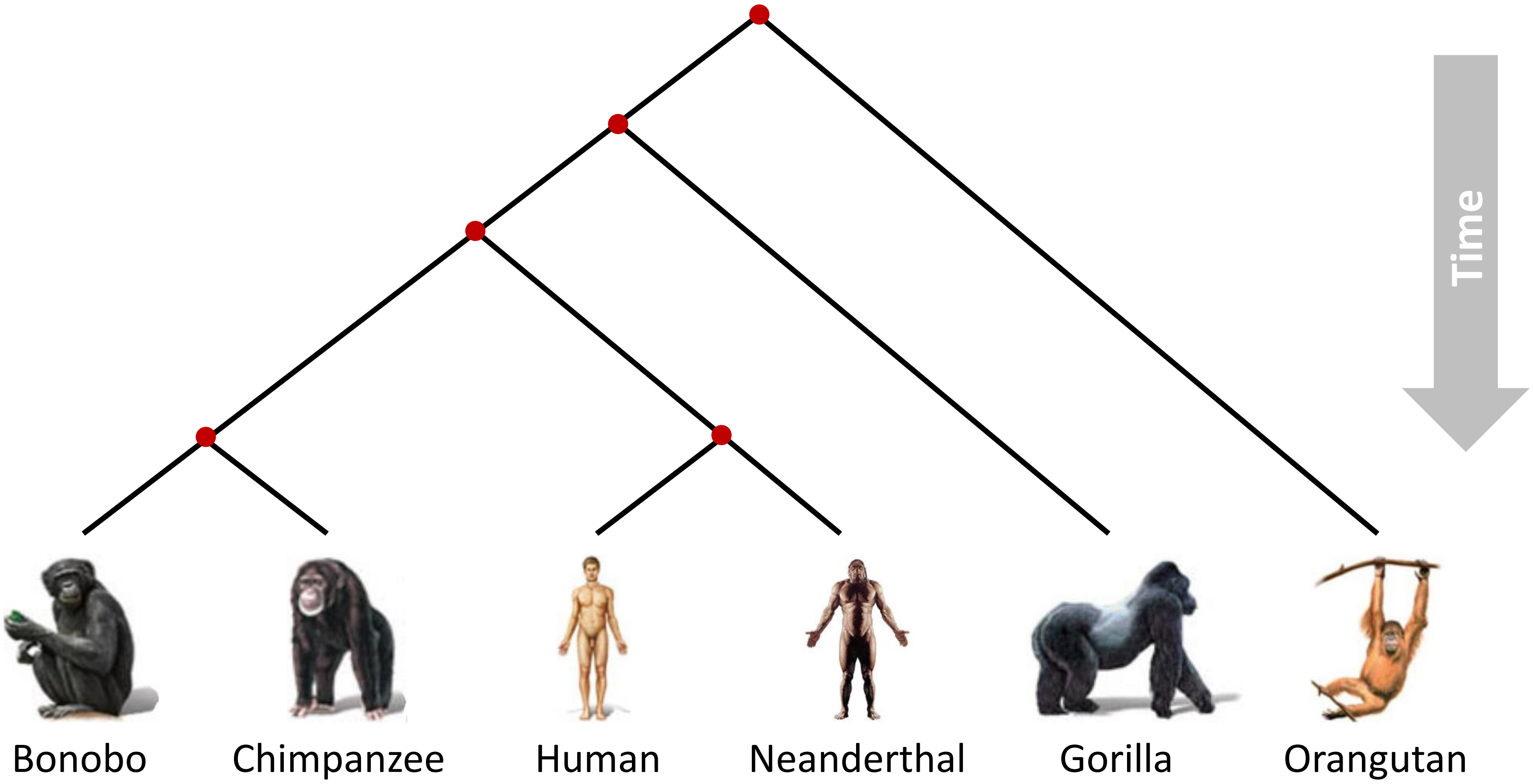
Work presented at SODA 2013 and ALENEX 2014

Computer Science Institute, Charles University, Prague, Czech Republic, 18 September 2014

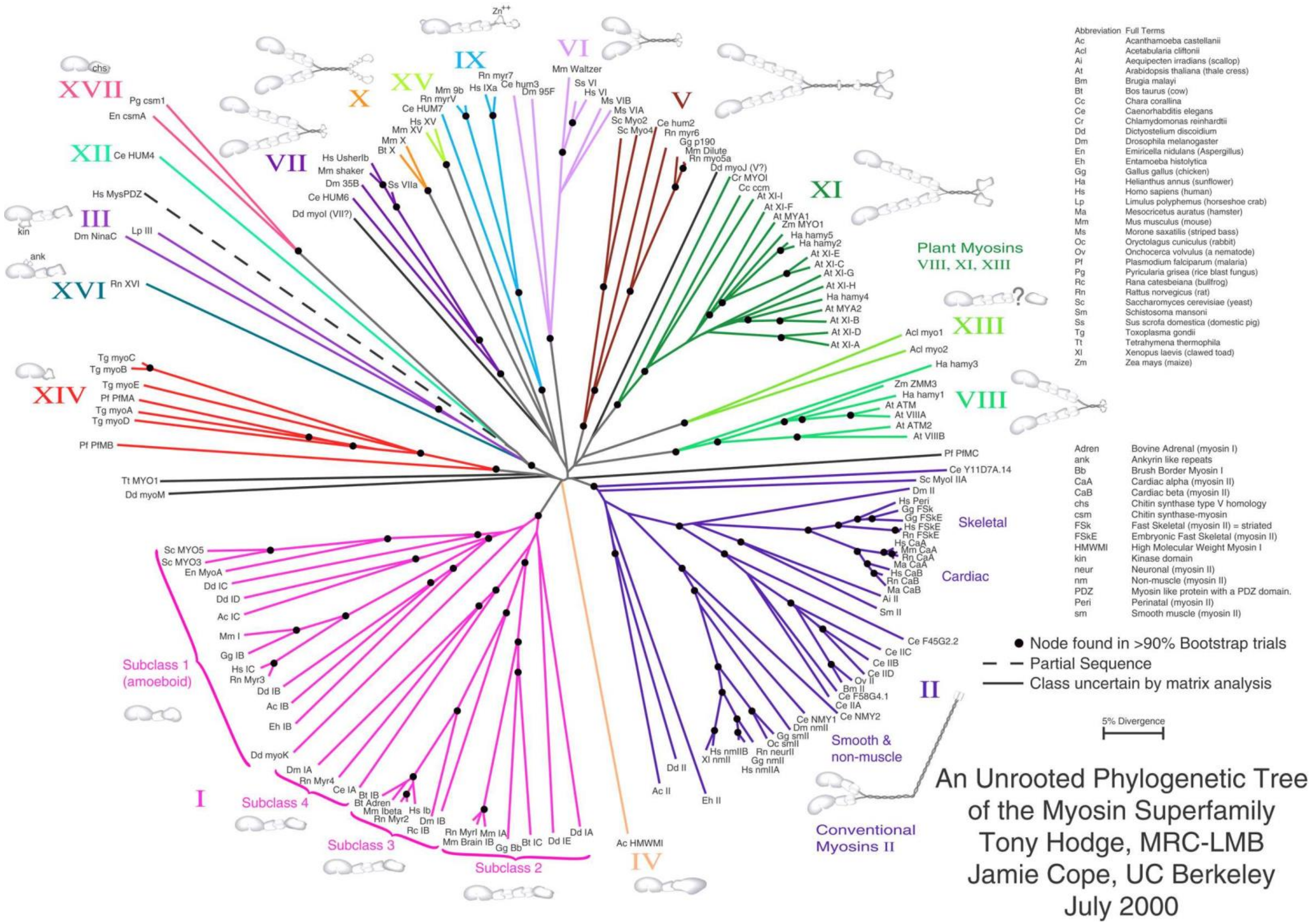
Outline

- Evolutionary **trees**
  - **rooted** vs. **unrooted**, binary vs. arbitrary degree
- Tree distances
  - Robinson-Foulds, **triplet**, **quartet**
- Results and previous work
  - **triplet**, **quartet** distances
- Algorithms
  - **triplet (quartet)**
- Experimental results (ALENEX 2014)

Rooted Evolutionary Tree

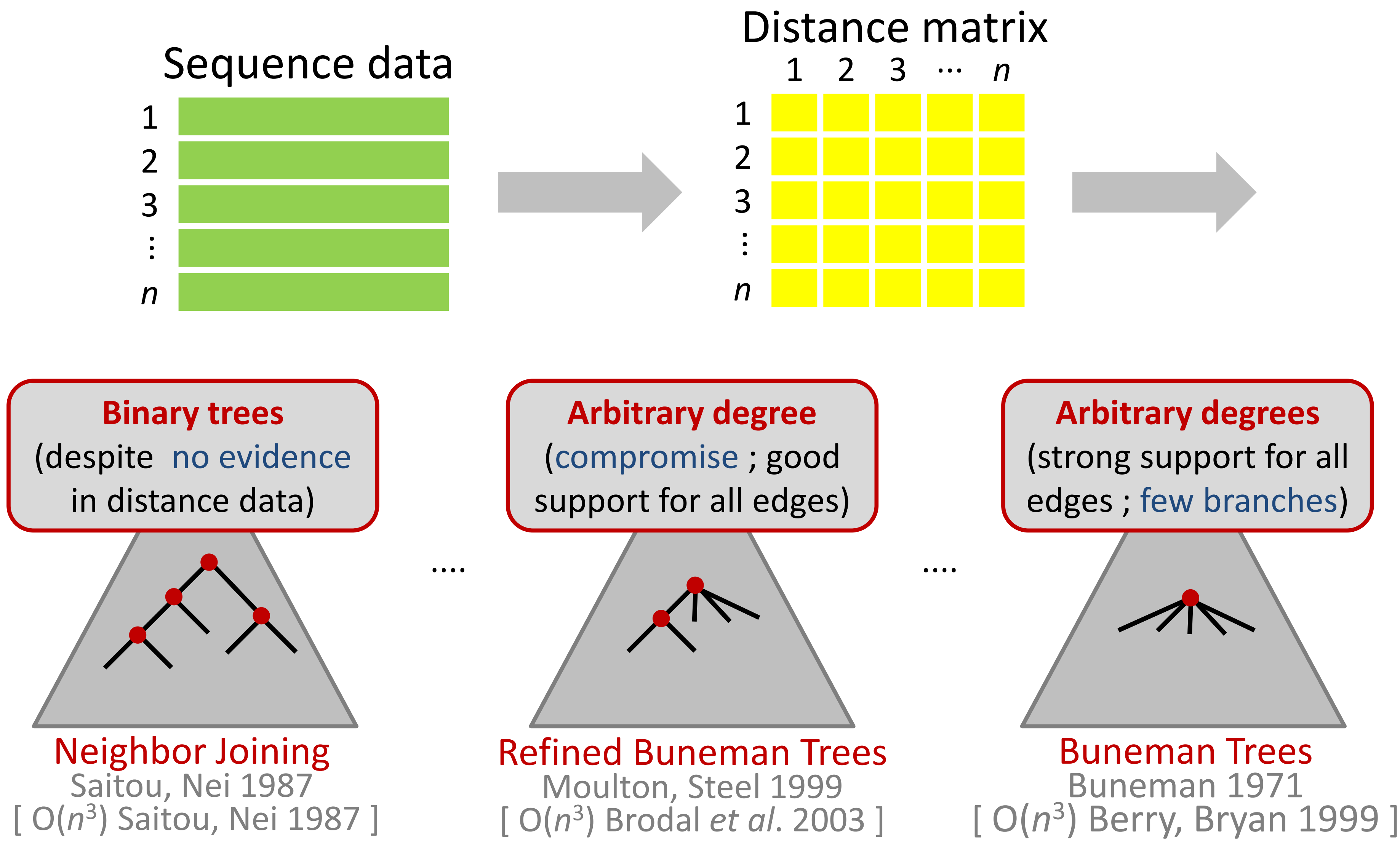


<u>Unrooted</u> Evolutionary Tree



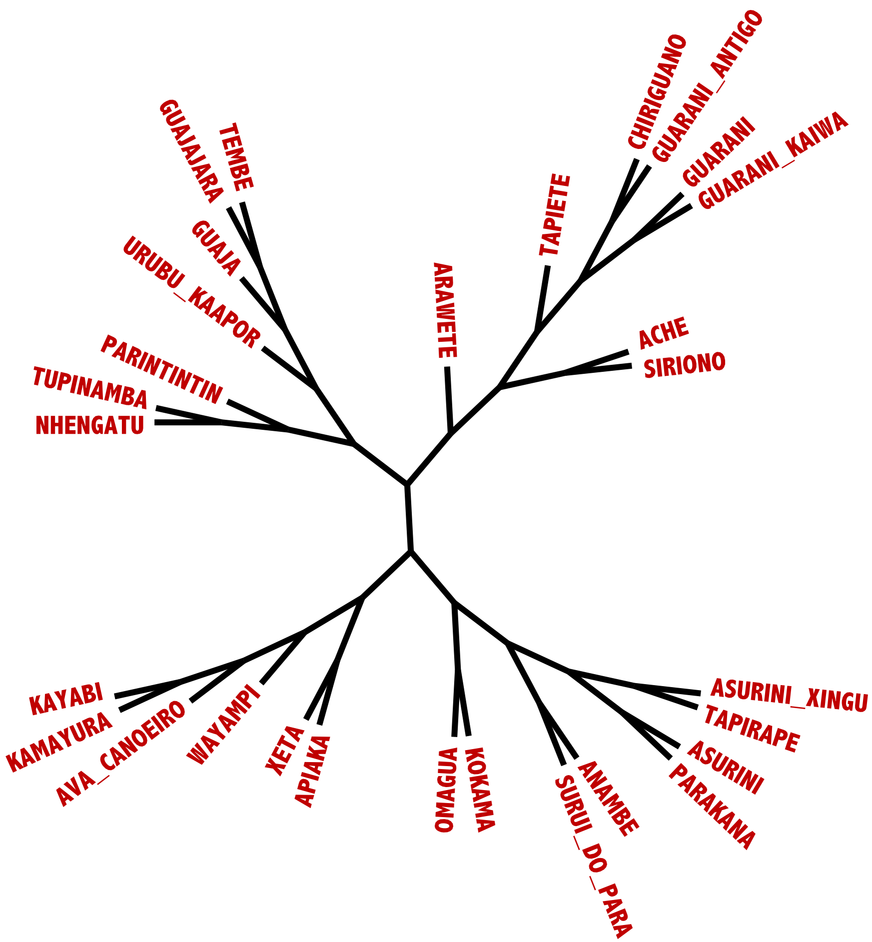
Dominant modern approach to study evolution is from DNA analysis

Constructing Evolutionary Trees – Binary or Arbitrary Degrees ?

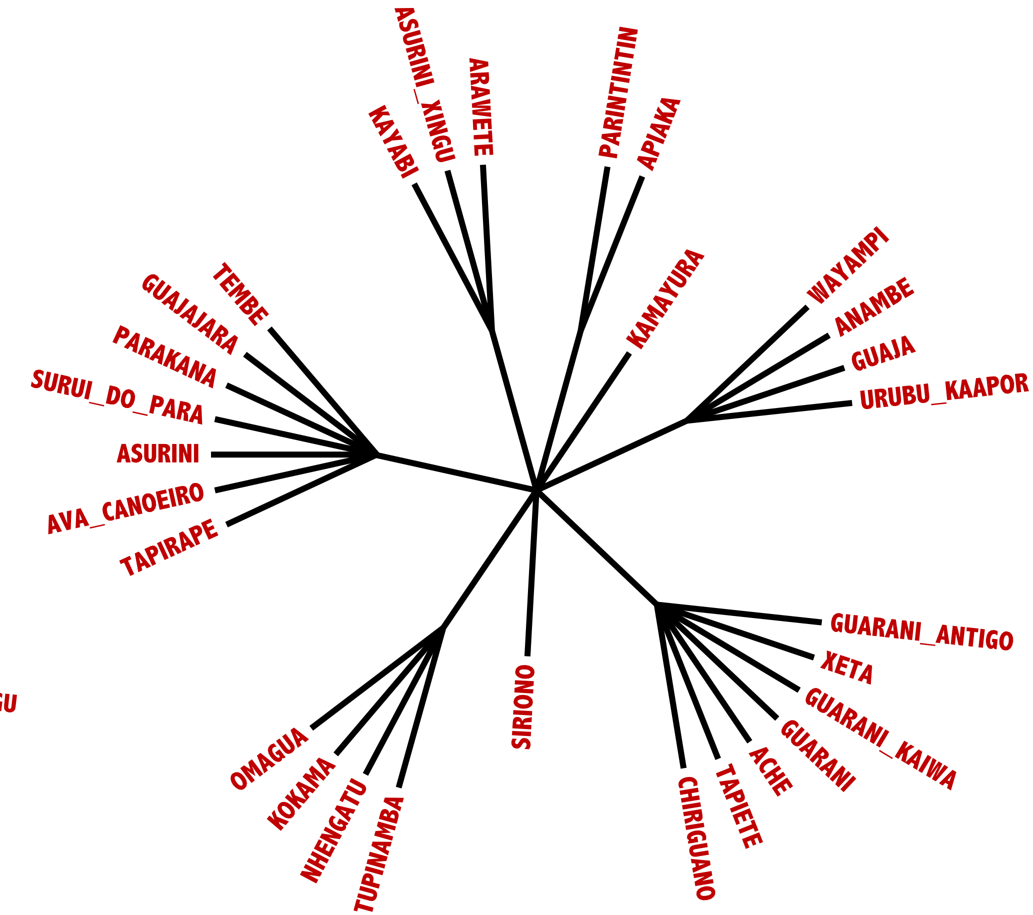


Data Analysis vs Expert Trees – Binary vs Arbitrary Degrees ?

Cultural Phylogenetics of the Tupi Language Family in Lowland South America.
R. S. Walker, S. Wichmann, T. Mailund, C. J. Atkisson. PLoS One. 7(4), 2012.

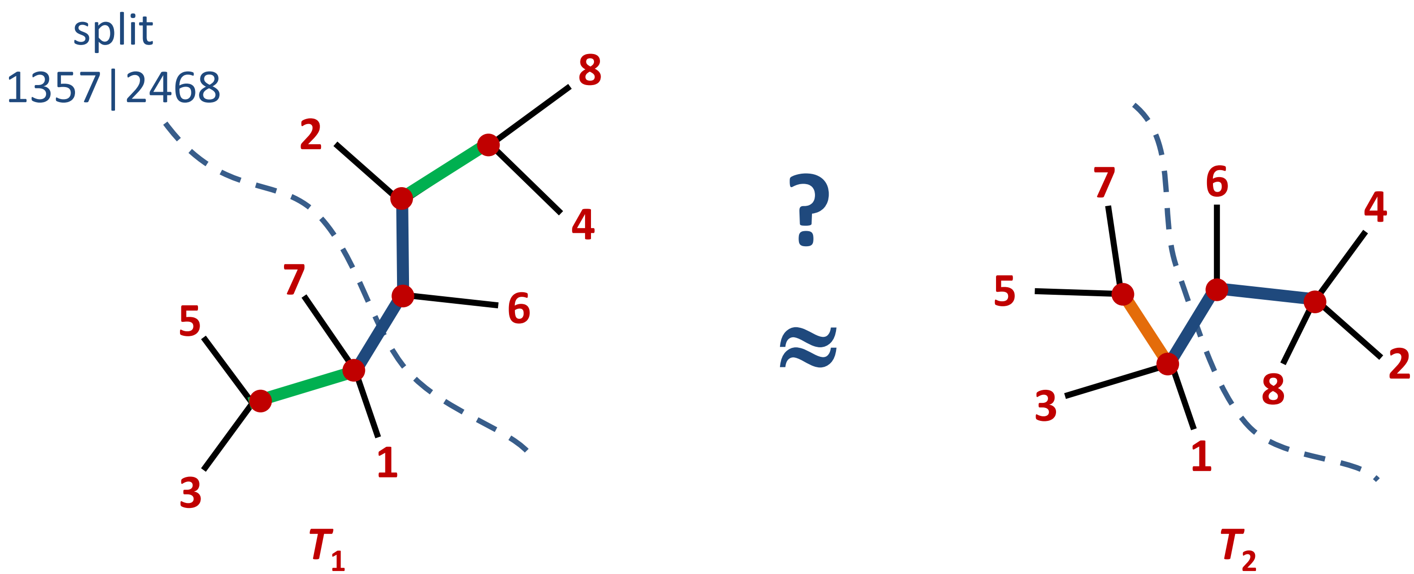


Neighbor Joining on linguistic data



Linguistic expert classification (Aryon Rodrigues)

Evolutionary Tree Comparison



Common	Only T_1	Only T_2
1357|2468	35|124678	57|123468
13567|248	48|123567	

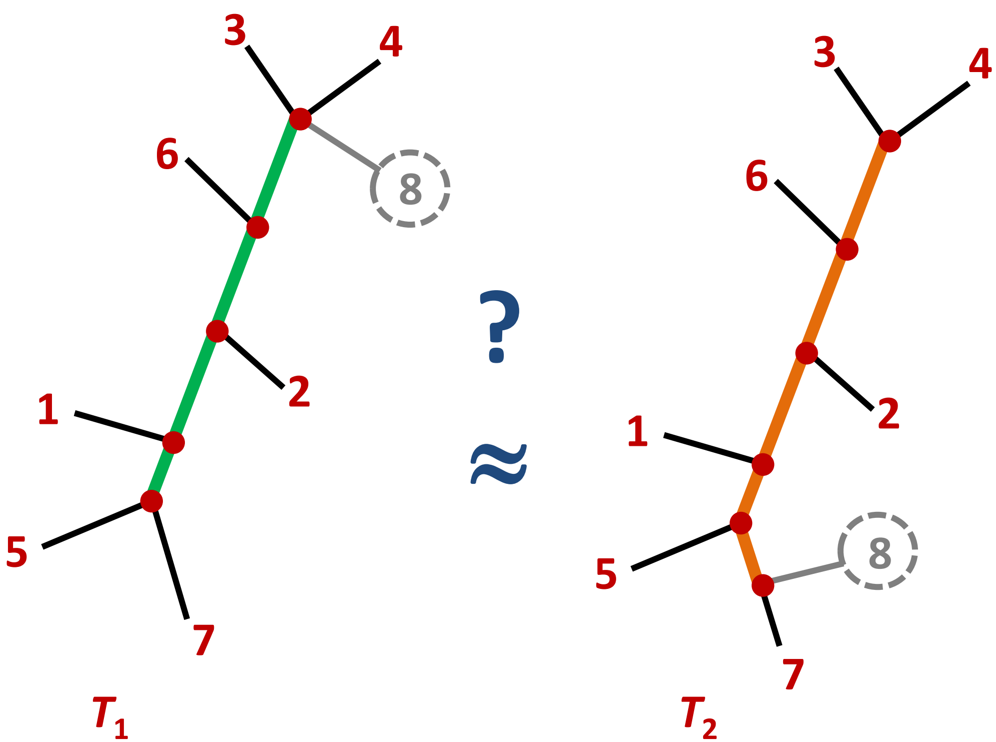
Robinson-Foulds distance = # non-common splits = **2** + **1** = **3**

D. F. Robinson and L. R. Foulds. Comparison of weighted labeled trees.
In *Combinatorial mathematics, VI*, Lecture Notes in Mathematics, pages 119–126. Springer, 1979.

[Day 1985] O(n) time algorithm using 2 x DFS + radix sort

Robinson-Foulds Distance (unrooted trees)

D. F. Robinson and L. R. Foulds. Comparison of weighted labeled trees. In *Combinatorial mathematics, VI*, Lecture Notes in Mathematics, pages 119–126. Springer, 1979.



Common	Only T_1	Only T_2
(none)	12567|348	125678|34
	1257|3468	12578|346
	157|23468	1578|2346
	57|123468	578|12346
		78|123456

RF-dist(T_1 , T_2) = **4** + **5** = **9**

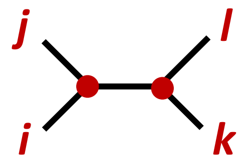
RF-dist($T_1\setminus\{8\}$, $T_2\setminus\{8\}$) = **0**

Robinson-Foulds very sensitive to outliers

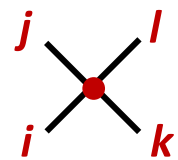
Quartet Distance (unrooted trees)

G. Estabrook, F. McMorris, and C. Meacham. Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units. *Systematic Zoology*, 34:193-200, 1985.

Consider all $\binom{n}{4}$ quartets, i.e. topologies of subsets of 4 leaves $\{i,j,k,l\}$

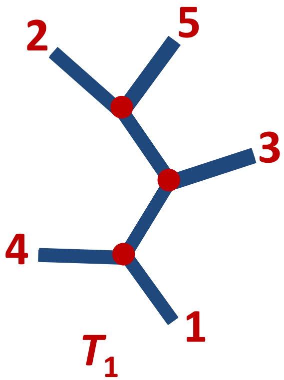


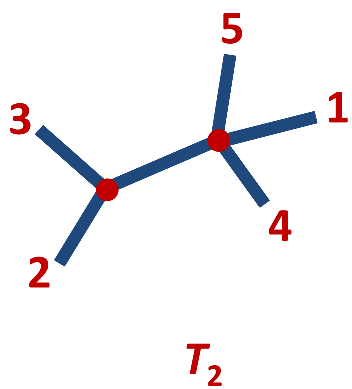
resolved : $ij|kl$



unresolved : $ijkl$

(only non-binary trees)





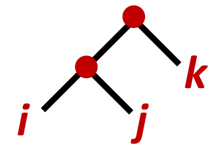
Quartet	T_1	T_2
{1,2,3,4}	14|23	14|23
{1,2,3,5}	13|25	15|23
{1,2,4,5}	14|25	1245
{1,3,4,5}	14|35	1345
{2,3,4,5}	25|34	23|45

Quartet-dist(T_1, T_2) = $\binom{n}{4}$ - # common quartets = **5** - **1** = **4**

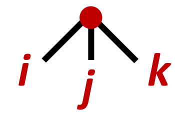
Triplet Distance (rooted trees)

D. E. Critchlow, D. K. Pearl, C. L. Qian: The triples distance for rooted bifurcating phylogenetic trees. *Systematic Biology*, 45(3):323-334, 1996.

Consider all $\binom{n}{3}$ triplets, i.e. topologies of subsets of 3 leaves $\{i,j,k\}$

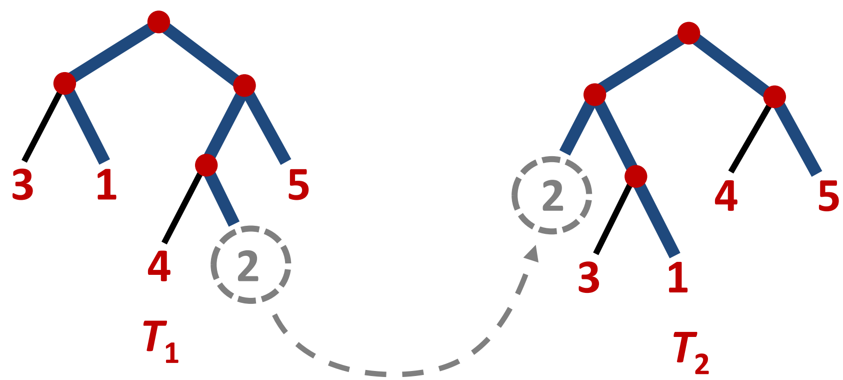


resolved : $k|ij$



unresolved : ijk

(only non-binary trees)

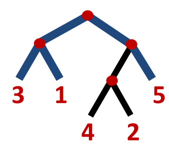


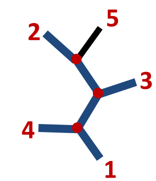
Triplet	T_1	T_2
{1,2,3}	2|13	2|13
{1,2,4}	1|24	4|12
{1,2,5}	1|25	5|12
{1,3,4}	4|13	4|13
{1,3,5}	5|13	5|13
{1,4,5}	1|45	1|45
{2,3,4}	3|24	4|23
{2,3,5}	3|25	5|23
{2,4,5}	5|24	2|45
{3,4,5}	3|45	3|45

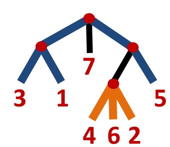
Triplet-dist$(T_1, T_2) = \binom{n}{3}$ - # common triplets = **10** - **5** = **5**

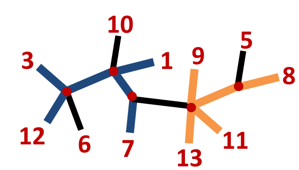
Computational Results

	Rooted Triplet distance		Unrooted Quartet distance	
Binary	$O(n^2)$	CPQ 1996	$O(n^3)$	D 1985
	$O(n \cdot \log^2 n)$	SBFPM 2013	$O(n^2)$	BTKL 2000
	$O(n \cdot \log n)$	[SODA 2013]	$O(n \cdot \log^2 n)$	BFP 2001
			$O(n \cdot \log n)$	BFP 2003
Arbitrary degrees	$O(n^2)$	BDF 2011	$O(d^9 \cdot n \cdot \log n)$	SPMBF 2007
	$O(n \cdot \log n)$	[SODA 2013]	$O(n^{2.688})$	NKMP 2011
			$O(d \cdot n \cdot \log n)$	[SODA 2013] [ALENEX 2014]



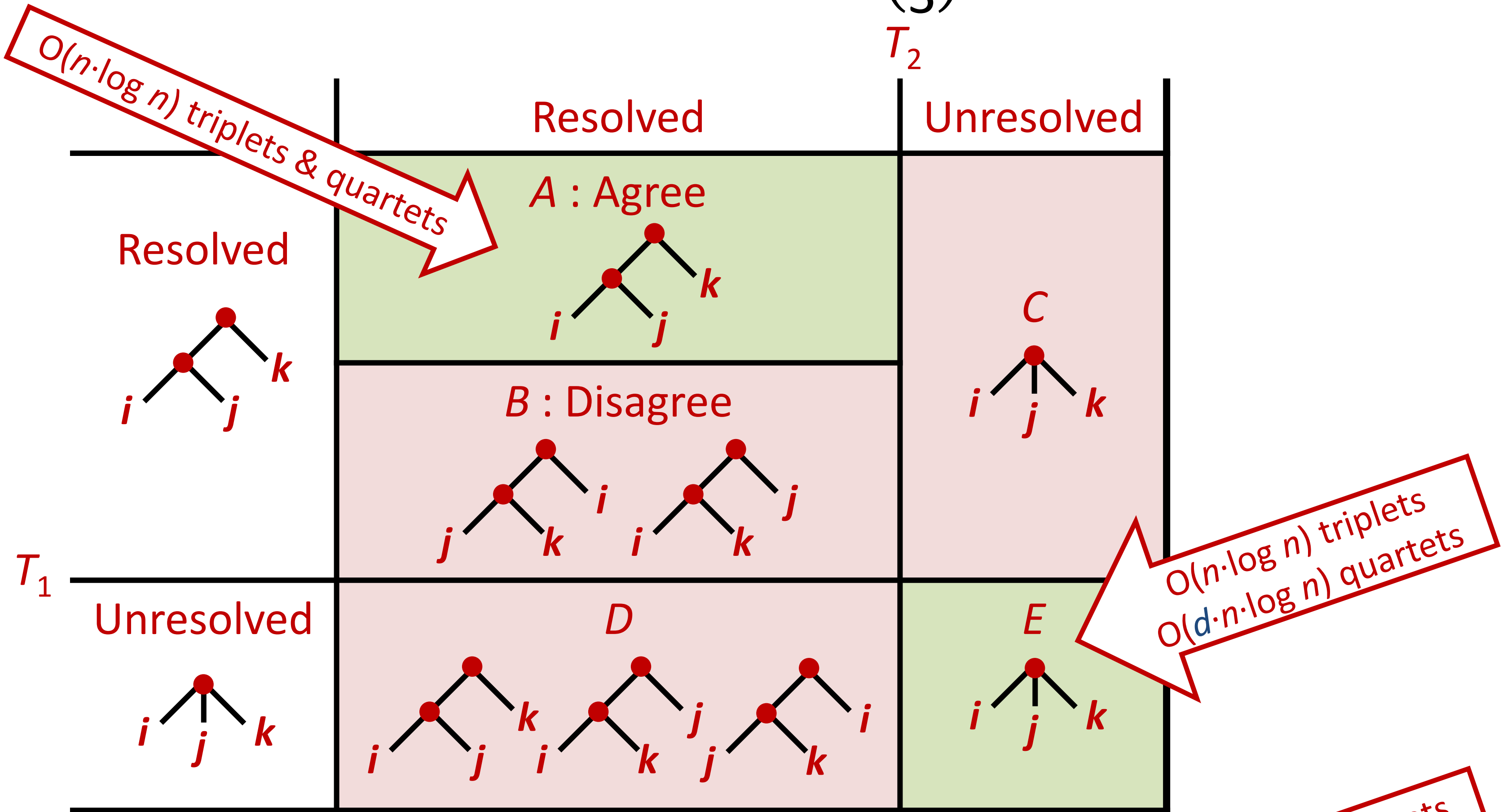


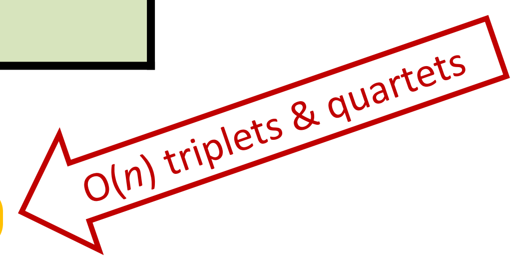




Distance Computation

$$\text{Triplet-dist}(T_1, T_2) = B + C + D = \binom{n}{3} - A - E$$





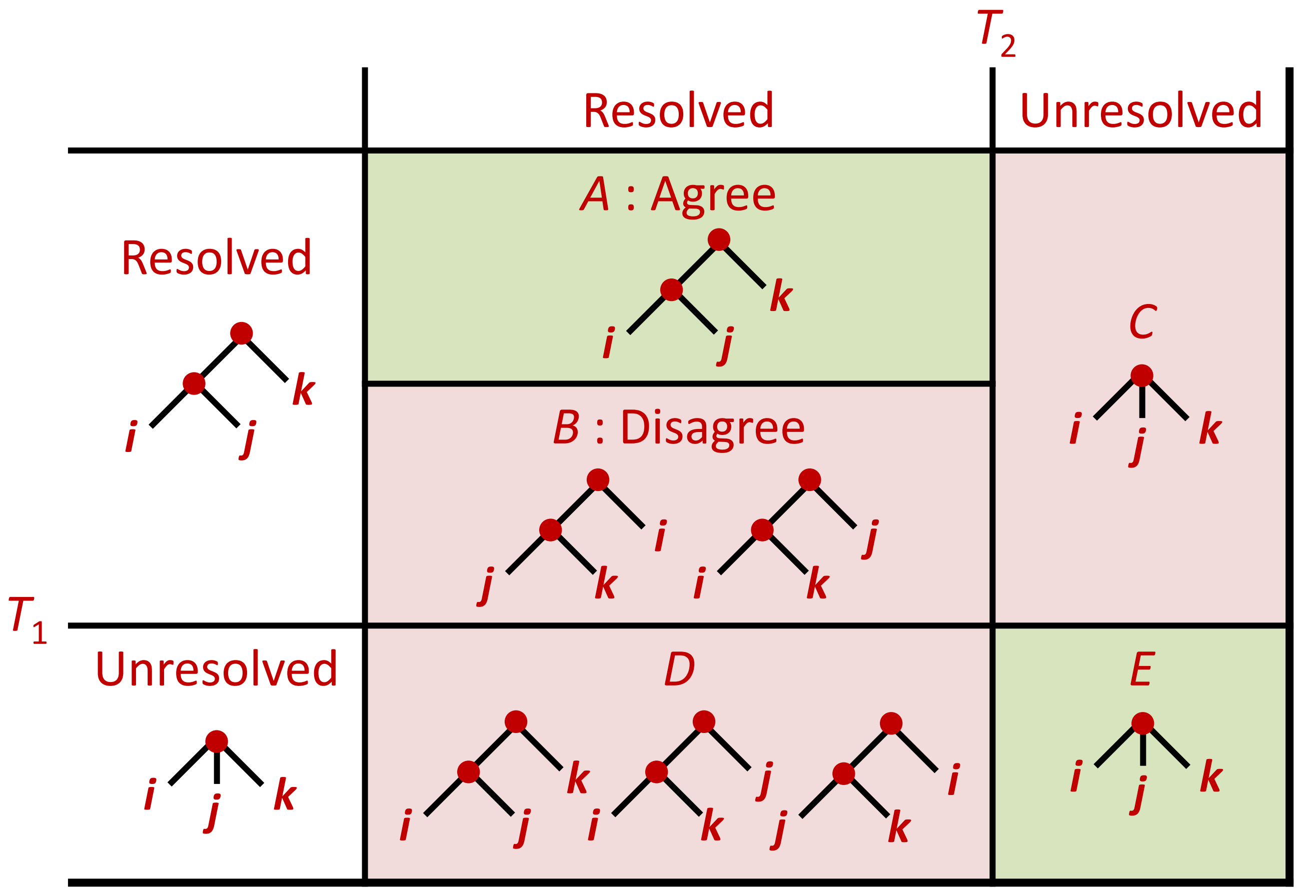
Sufficient to compute *A* and *E*

D + *E* and *C* + *E* unresolved in one tree

(For binary trees C, D and E are all zero)

Parameterized Triplet & Quartet Distances

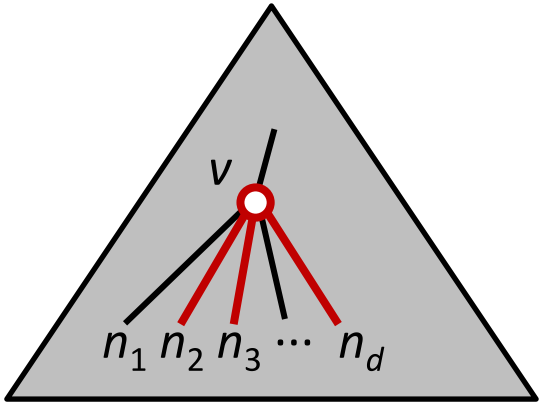
$$B + \alpha \cdot (C + D)\ , \quad 0 \le \alpha \le 1$$



BDF 2011 O(n^2) for triplet, NKMP 2011 O($n^{2.688}$) for quartet
[SODA 2013/ALENEX 2014] O($n \cdot \log n$) and O($d \cdot n \cdot \log n$), respectively

Counting Unresolved Triplets in One Tree

$$\sum_{v} \sum_{i<j<k} n_i \cdot n_j \cdot n_k$$

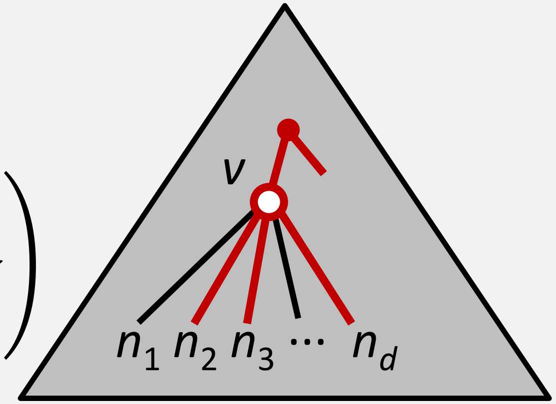


Triplet anchored at *v*

Computable in O(*n*) time using DFS + dynamic programming

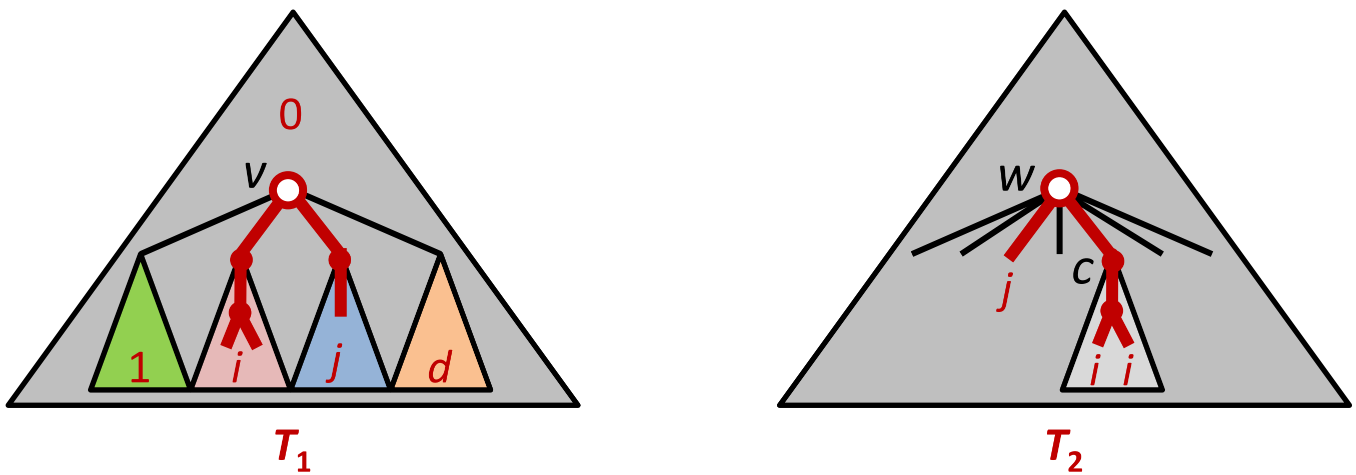
Quartets (root tree arbitrary)

$$\sum_{v} \left(\sum_{i<j<k<l} n_i \cdot n_j \cdot n_k \cdot n_l + \left(n - \sum_{l} n_l \right) \sum_{i<j<k} n_i \cdot n_j \cdot n_k \right)$$



Quartet anchored at *v*

Counting Agreeing Triplets (Basic Idea)

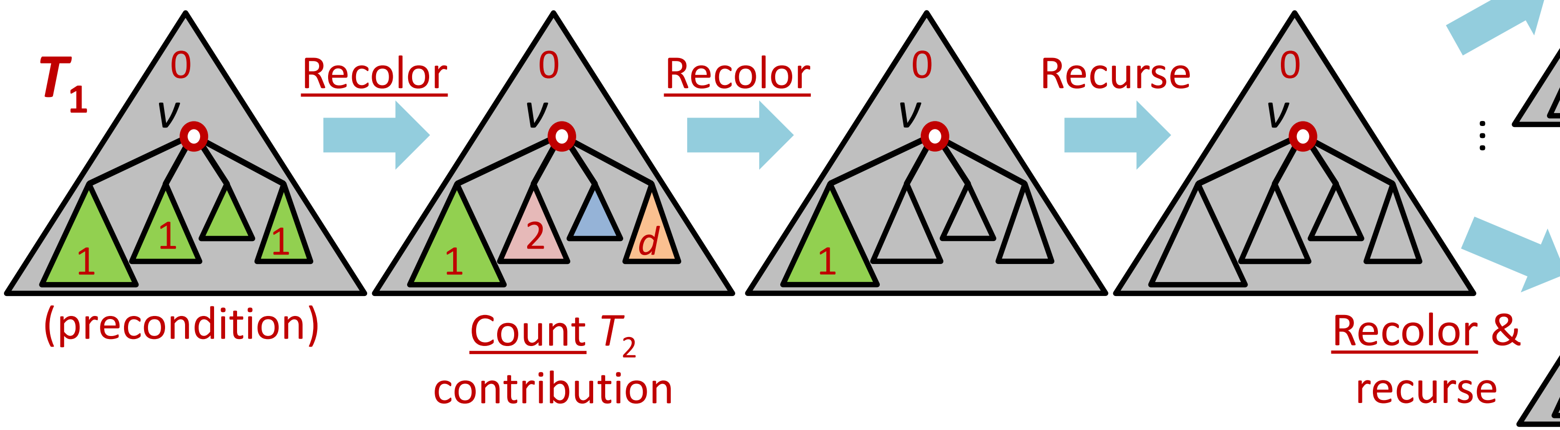


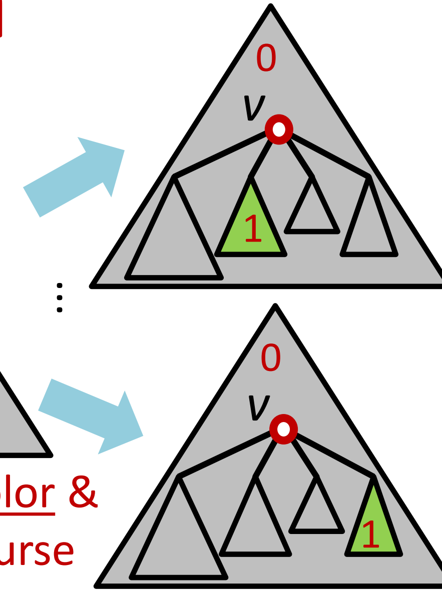
$$\sum_{v \in T_1} \sum_{w \in T_2} \sum_{c} \sum_{1 \le i \le d} \binom{n_i^{\,c}}{2} \left(n^w - n^c - n_i^{\,w} + n_i^{\,c}\right)$$

$$\sum_{1 \le i \le d} n_i^{\,w}$$

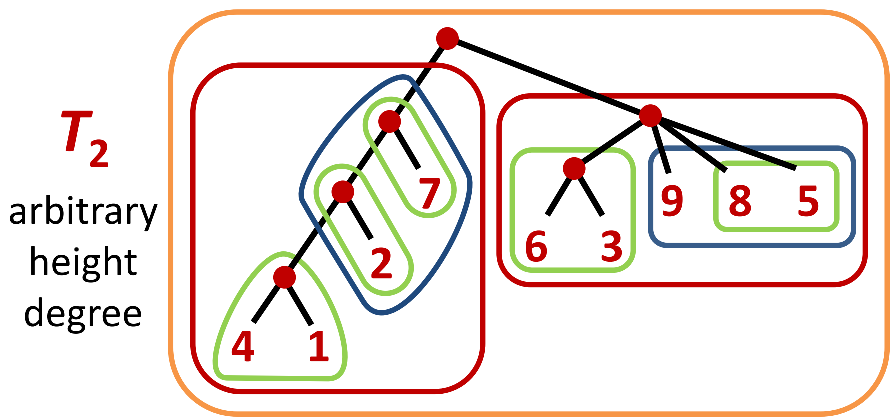
Efficient Computation

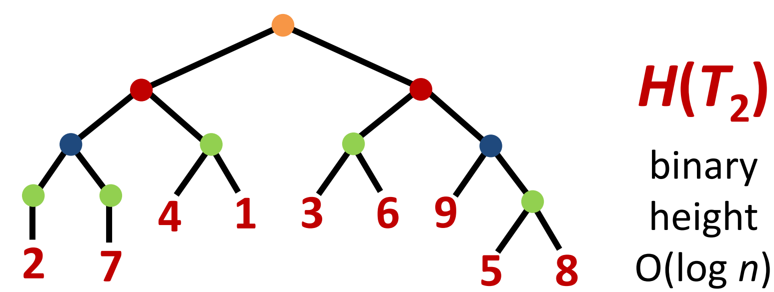
Limit recolorings in T_1 (and T_2) to $O(n \cdot \log n)$





Reduce recoloring cost in T_2 from $O(n^2)$ *to* $O(n \cdot \log^2 n)$

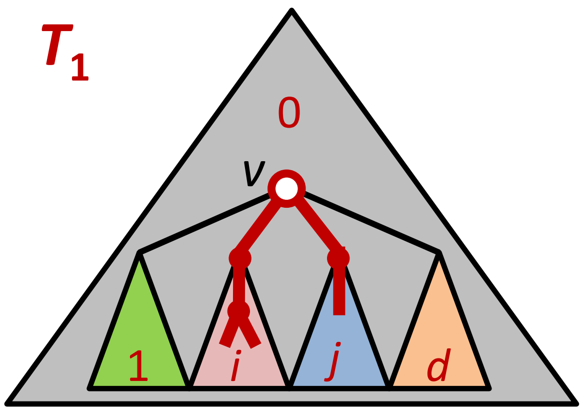




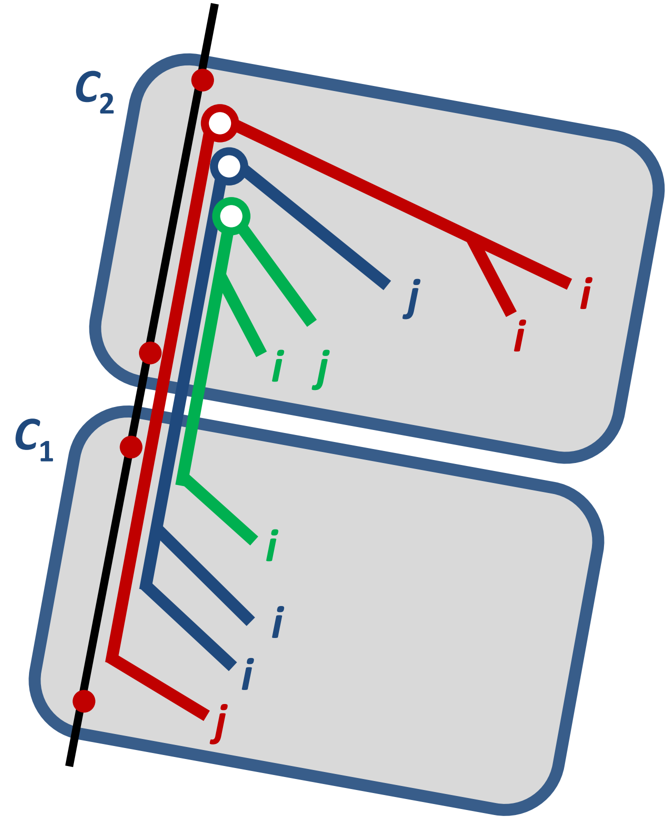
Reduce recoloring cost in T_2 from $O(n \cdot \log^2 n)$ *to* $O(n \cdot \log n)$

- Contract T_2 and reconstruct $H(T_2)$ during recursion

Counting Agreeing Triplets (II)



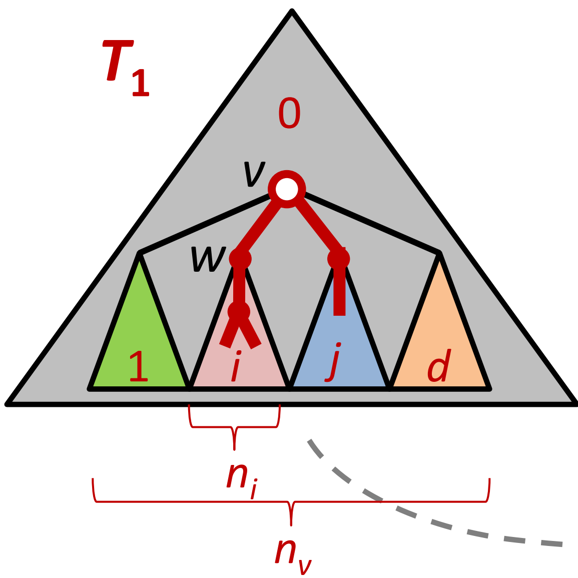
node in $H(T_2)$ =
component
composition in T_2

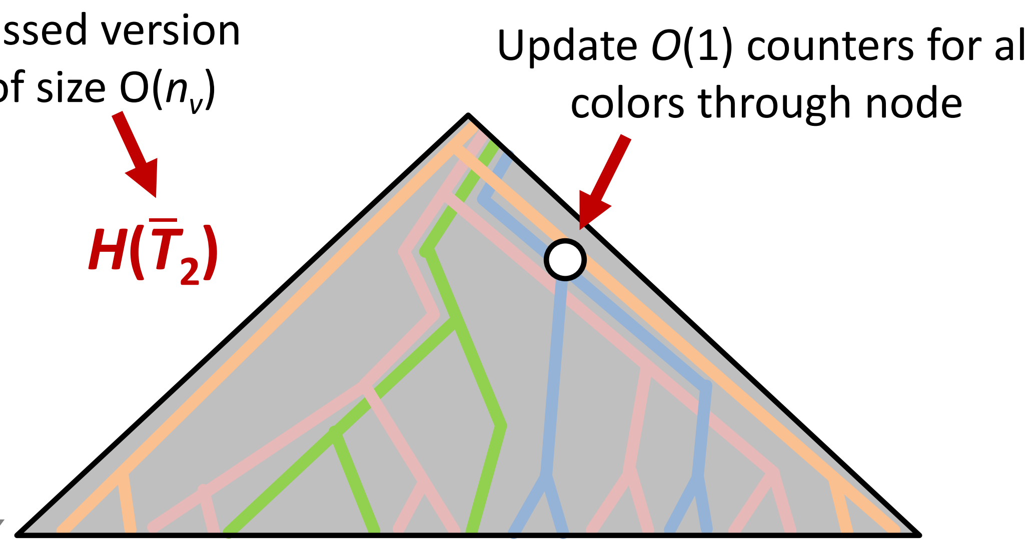


Contribution to agreeing triplets at node in $H(T_2)$

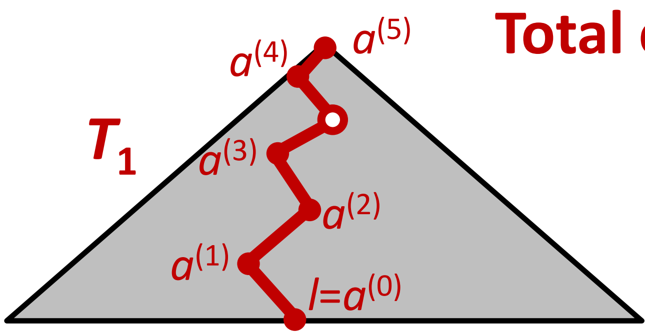
$$\sum_{1\le i\le d} n_i^{C_1}\cdot n_{i\uparrow *}^{C_2} + \sum_{1\le i\le d} \binom{n_i^{C_1}}{2}\left(n_*^{C_2} - n_i^{C_2}\right) + \sum_{1\le i\le d}\left(n_*^{C_1} - n_i^{C_1}\right) n_{(ii)}^{C_2}$$

From O(n·log² *n*) to O(*n*·log *n*)





Colored path lengths $\displaystyle\sum_{2\leq i\leq d} \log \binom{|\overline{T}_2|}{n_i} = \sum_{2\leq i\leq d} n_i \cdot \log \frac{n^v}{n_i}$



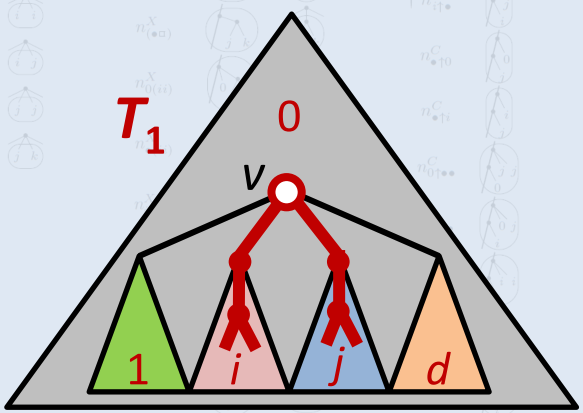
Total cost for updating counters

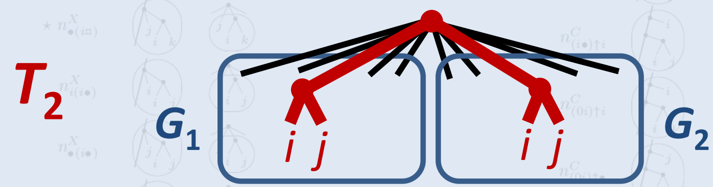
$$\sum_{\text{leaf } l\in T_1} \; \sum_{\substack{\text{ancestor } a^{(j)} \\ \text{not heavy child}}} \log \frac{n^{a^{(j+1)}}}{n^{a^{(j)}}} = n\cdot \log n$$

Counting Quartets...

- Root T_1 and T_2 arbitrary
- Keep up to $7d^2 + 97d + 29$ different counters per node in $H(T_2)$...

Bottleneck in computing disagreeing resolved-resolved quartets



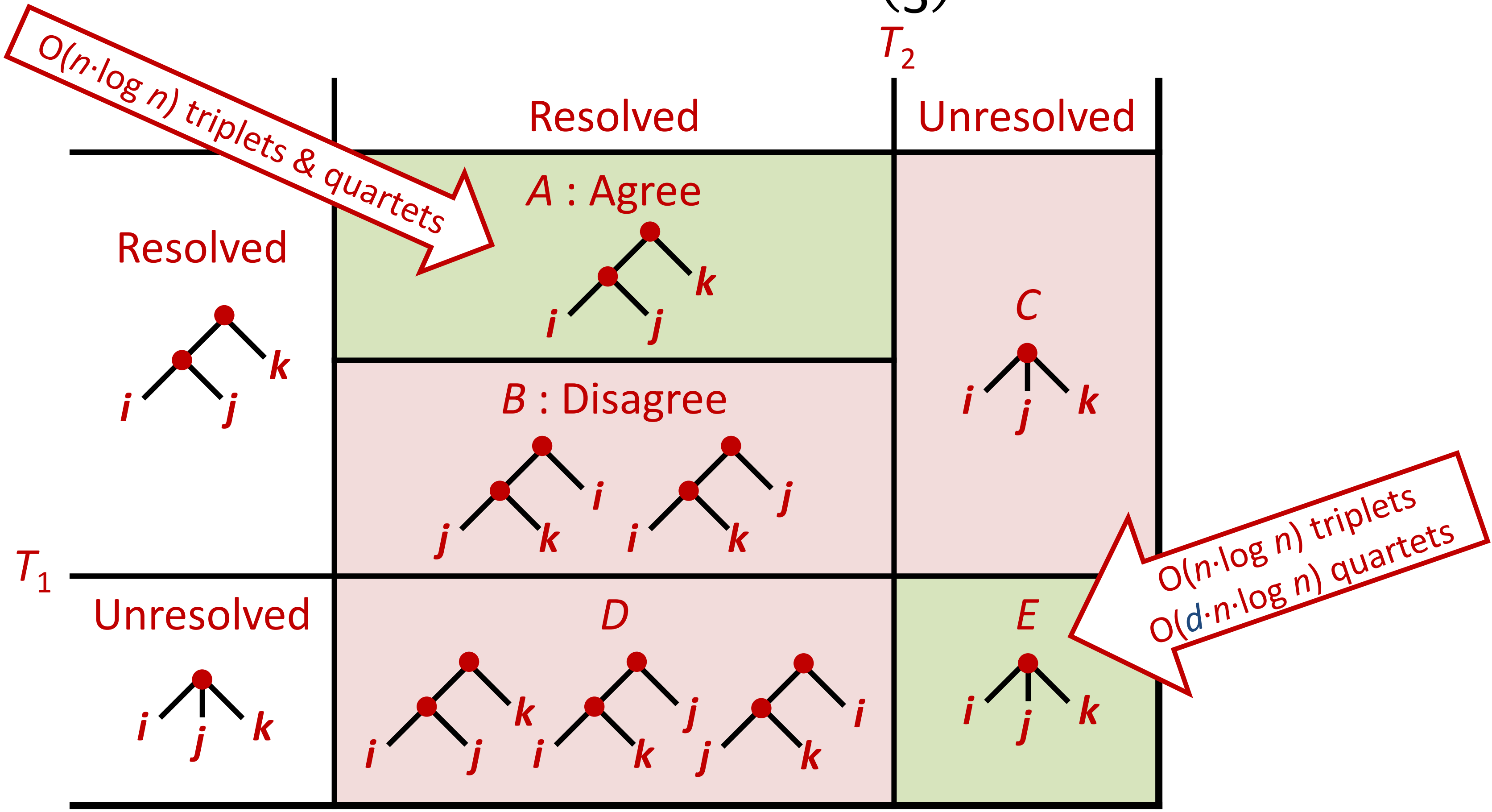


$$\sum_{1 \le i < d} \; \sum_{i < j \le d} n_{(ij)}^{G_1} \cdot n_{(ij)}^{G_2}$$

double-sum ⇒ factor d time

Distance Computation

$$\text{Triplet-dist}(T_1, T_2) = B + C + D = \binom{n}{3} - A - E$$



Sufficient to compute *A* and *E*

ALENEX 2014: Implementation

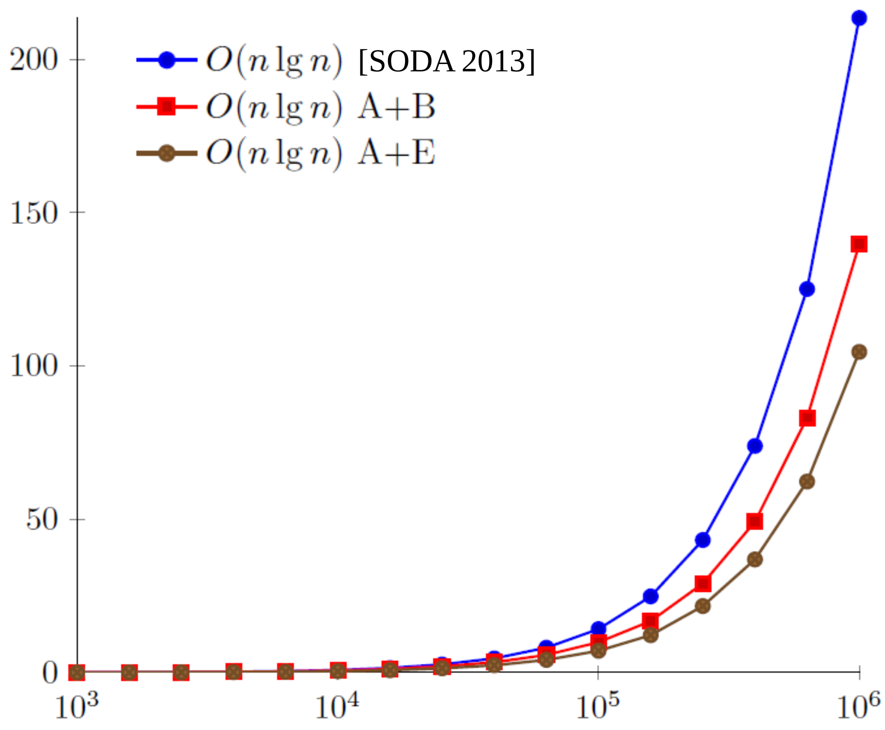
(M.Sc. thesis Morten Kragelund Holt and Jens Johansen)

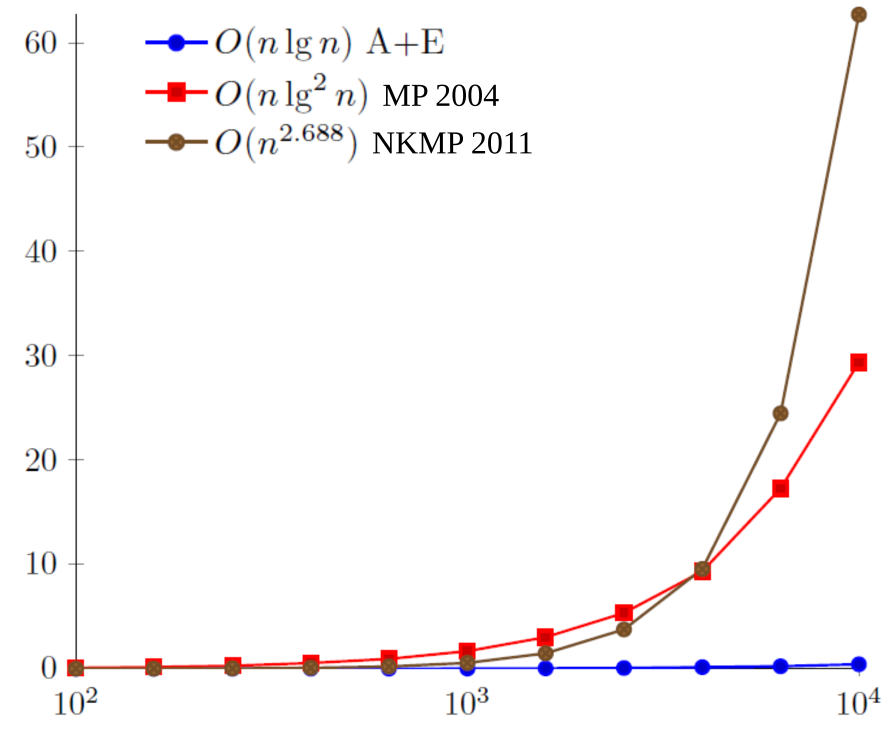
	Binary		Arbitrary degree	
	time	counters	time	counters
Triplet	O(n log n)	6	O(n log n)	$4d+2$
Quartet	O(n log n)	40	O(max(d_1, d_2) n log n)	$2d^2 + 79d + 22$ (B, with $T_1 \leftrightarrow T_2$)
			O(min(d_1, d_2) n log n)	$7d^2 + 97d + 29$ (B, no swap)
				$d^2 + 12d + 12$ (E, no swap)

Worst-case #counters per node in HDT(T_2)

- First implementation for triplets for arbitrary degree
- Space usage ≈10 KB per node for quartet (binary trees)
  ⇨ can handle ≈ 1,000,000 leaves
- 64 bit integers, except 128 bit integers for values > n^3
  ⇨ quartet distance of up to ≈ 2,000,000 leaves

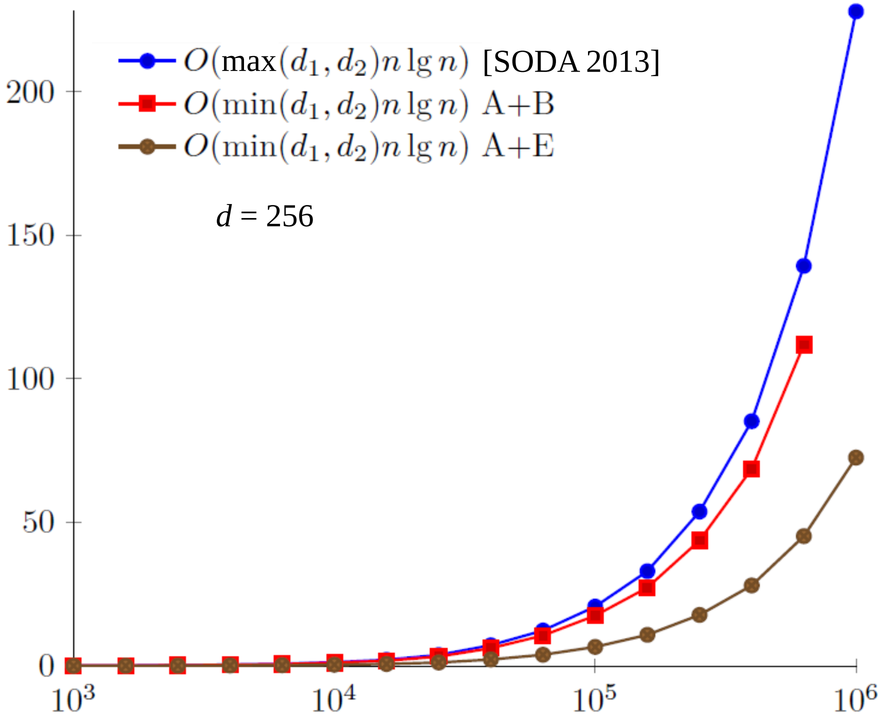
Experimental Results
Quartet Distance – Binary Trees

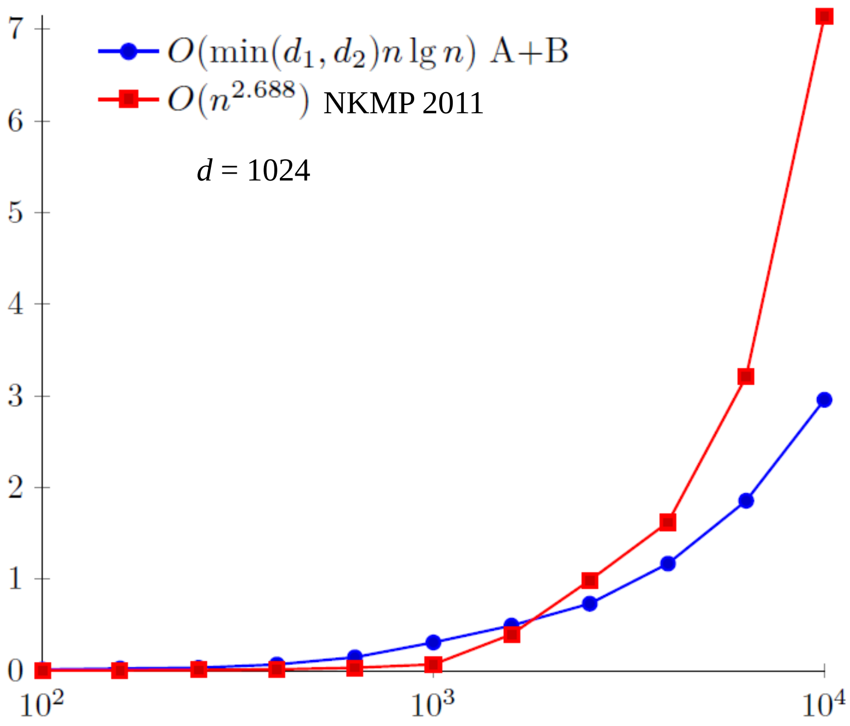




- [ALENEX 2014] are the first O(*n*·log *n*) implementations
- MP 2004 overhead from working with polynomials

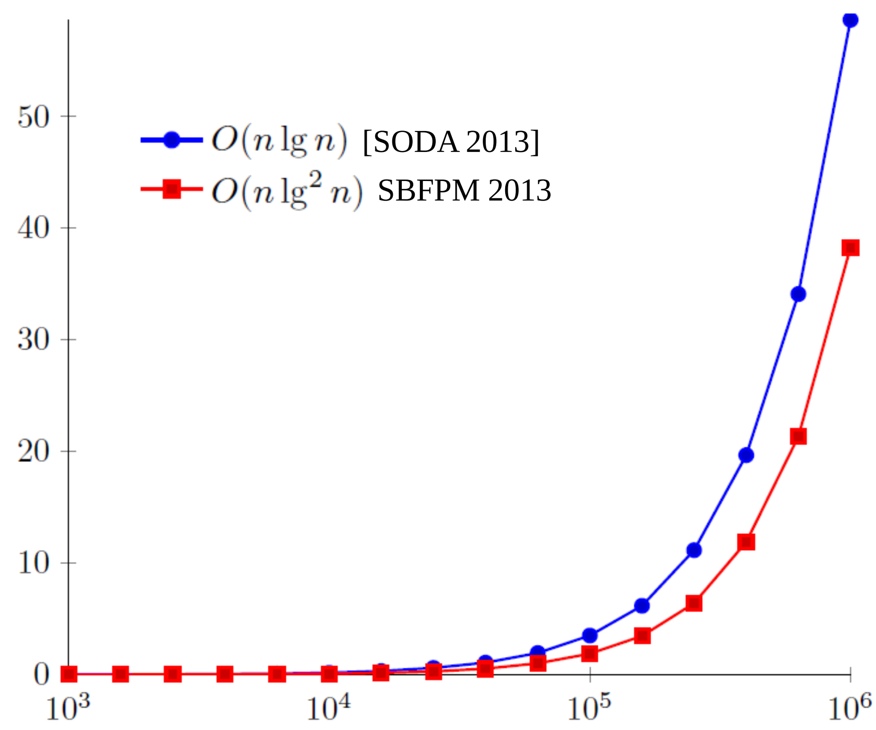
Experimental Results
Quartet Distance – High Degree Trees





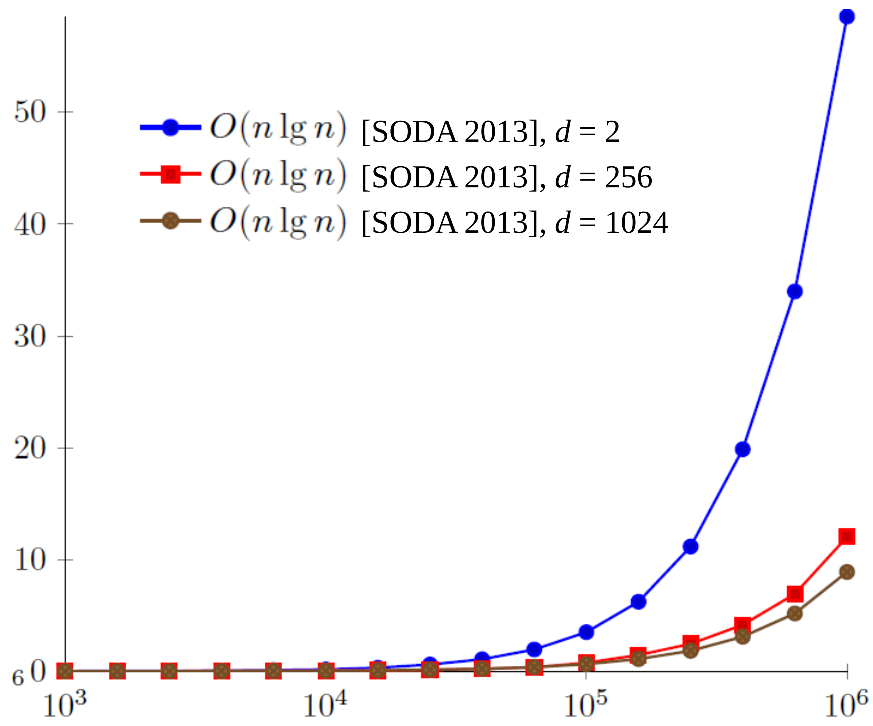
- [ALENEX 2014] are the first *n*·poly(log *n*,*d*) implementation

Experimental Results
Triplet Distance – Binary Trees



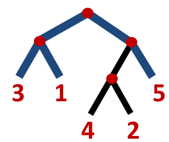
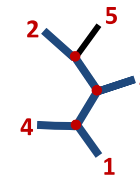
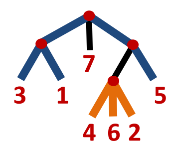
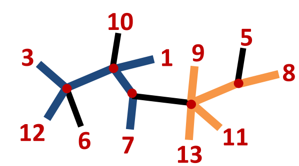
- [ALENEX 2014] are the first O(*n*·log *n*) implementation
- SBFPM 2013 only binary trees, no contractions

Experimental Results
Triplet Distance – High Degree Trees



- [ALENEX 2014] first implementation
- Triplet distance appears hardest for binary trees

Summary

	Rooted Triplet distance	Unrooted Quartet distance
Binary	 $O(n^2)$ CPQ 1996 $O(n \cdot \log^2 n)$ ★ SBFPM 2013 $O(n \cdot \log n)$ [SODA 2013]	 $O(n^3)$ D 1985 $O(n^2)$ BTKL 2000 $O(n \cdot \log^2 n)$ BFP 2001 $O(n \cdot \log n)$ BFP 2003 $O(n \cdot \log n)$ ★ [SODA 2013]
Arbitrary degrees	 $O(n^2)$ BDF 2011 $O(n \cdot \log n)$ ★ [SODA 2013]	 $O(d^9 \cdot n \cdot \log n)$ SPMBF 2007 $O(n^{2.688})$ NKMP 2011 $O(d \cdot n \cdot \log n)$ ★ [SODA 2013] [ALENEX 2014]

$o(n \cdot \log n)$?

$O(n \cdot \log n)$?

d = minimal degree of any node in T_1 and T_2

★ = fastest implementation for large n

References

- ***On the Scalability of Computing Triplet and Quartet Distances.***
  M.K. Holt, J. Johansen, G.S. Brodal. ALENEX 2014.
- *Algorithms for Computing the Triplet and Quartet Distances for Binary and General Trees*.
  A. Sand, M.K. Holt, J. Johansen, R. Fagerberg, G.S. Brodal, C.N.S. Pedersen, T. Mailund.
  Biology - Special Issue on Developments in Bioinformatic Algorithms, 2013.
- *A practical* O(n log^2 n) *time algorithm for computing the triplet distance on binary trees*.
  A. Sand, G.S. Brodal, R. Fagerberg, C.N.S. Pedersen, T. Mailund. BMC Bioinformatics 2013.
- ***Efficient Algorithms for Computing the Triplet and Quartet Distance Between Trees of Arbitrary Degree.*** **G.S. Brodal, R. Fagerberg, C.N.S. Pedersen, T. Mailund, A. Sand.**
  SODA 2013.
- *A sub-cubic time algorithm for computing the quartet distance between two general trees*.
  J. Nielsen, A. K. Kristensen, T. Mailund, C.N.S. Pedersen.
  Algorithms in Molecular Biology 2011.
- *Computing the Quartet Distance Between Evolutionary Trees of Bounded Degree*.
  M. Stissing, C.N.S. Pedersen, T. Mailund, G.S. Brodal, R. Fagerberg. APBC 2007.
- *QDist - Quartet Distance between Evolutionary Trees*.
  T. Mailund and C.N. S. Pedersen. Bioinformatics 2004.
- *Computing the Quartet Distance Between Evolutionary Trees in Time* O(n log n).
  G.S. Brodal, R. Fagerberg, C.N.S. Pedersen. Algorithmica 2004.
- *Computing the Quartet Distance Between Evolutionary Trees in Time* O(n log^2 n).
  G.S. Brodal, R. Fagerberg, C.N.S. Pedersen. ISAAC 2001.

birc.au.dk/software