Fast Phylogenetic Approaches

Xiaofan Zhou
Integrative Microbiology Research Centre
South China Agriculture University

Why is tree inference so diffcult?

- Too many trees to look at
- Too many calculations to do

Too many trees

No. of binary unrooted trees with n tips:

$$= 1 \times 3 \times 5 \cdots \times (2n-5)$$

$$= (2n-3)!/\left(2^{(n-2)}(n-2)!\right)$$

Tips	Binary unrooted trees
5	15
10	2,027,025
20	2.22×10^{18}
30	8.69×10^{36}
40	1.31×10^{55}
50	2.84×10^{74}
⋮	⋮

Sunway Taihu-Light
World's fastest and largest supercomputer
Peak Flops: 125.4×10^{15}

~3.31×10^{21} billion years

Too many calculations

- Branch length estimation
- Model parameter optimization
- ……

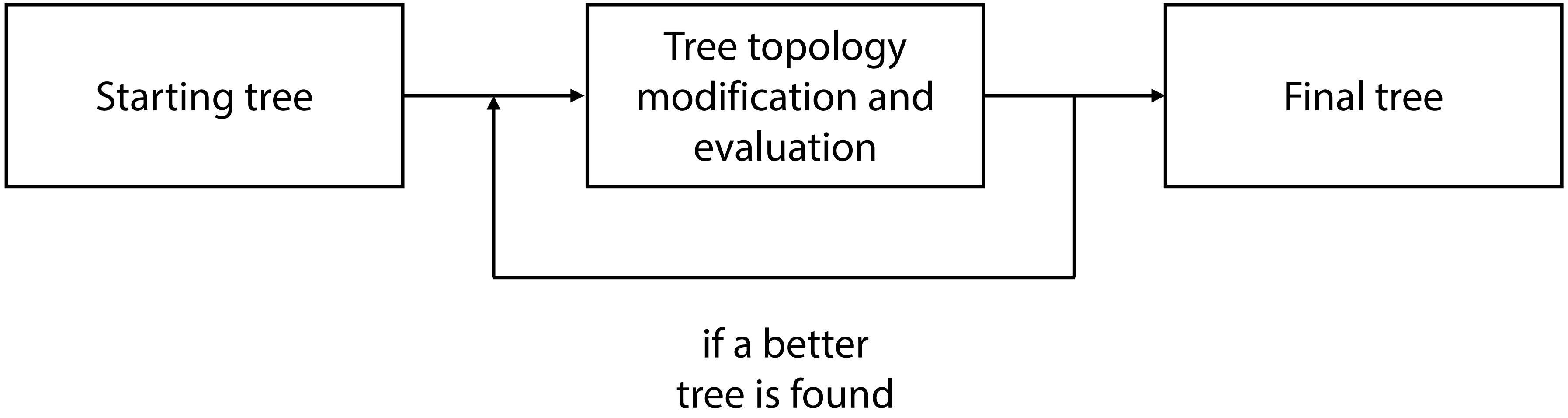
Fast phylogenetic approaches

- Too many trees to look at
  - Heuristic search of the tree space

- Too many calculations to do
  - Approximate estimation of branch length

- Other techniques for fast phylogenetics

Heuristic tree search



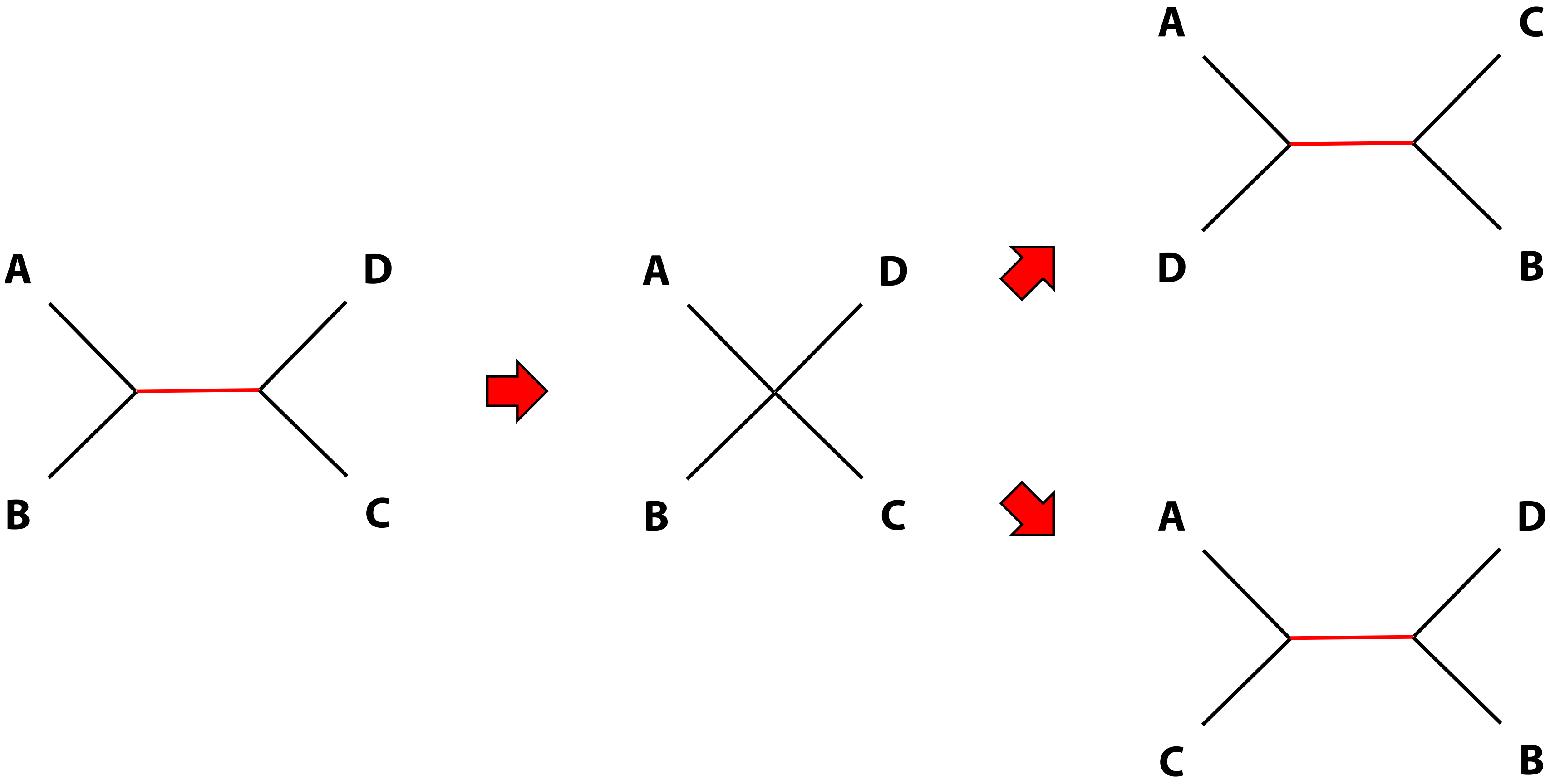
Starting tree

Tree topology modification and evaluation

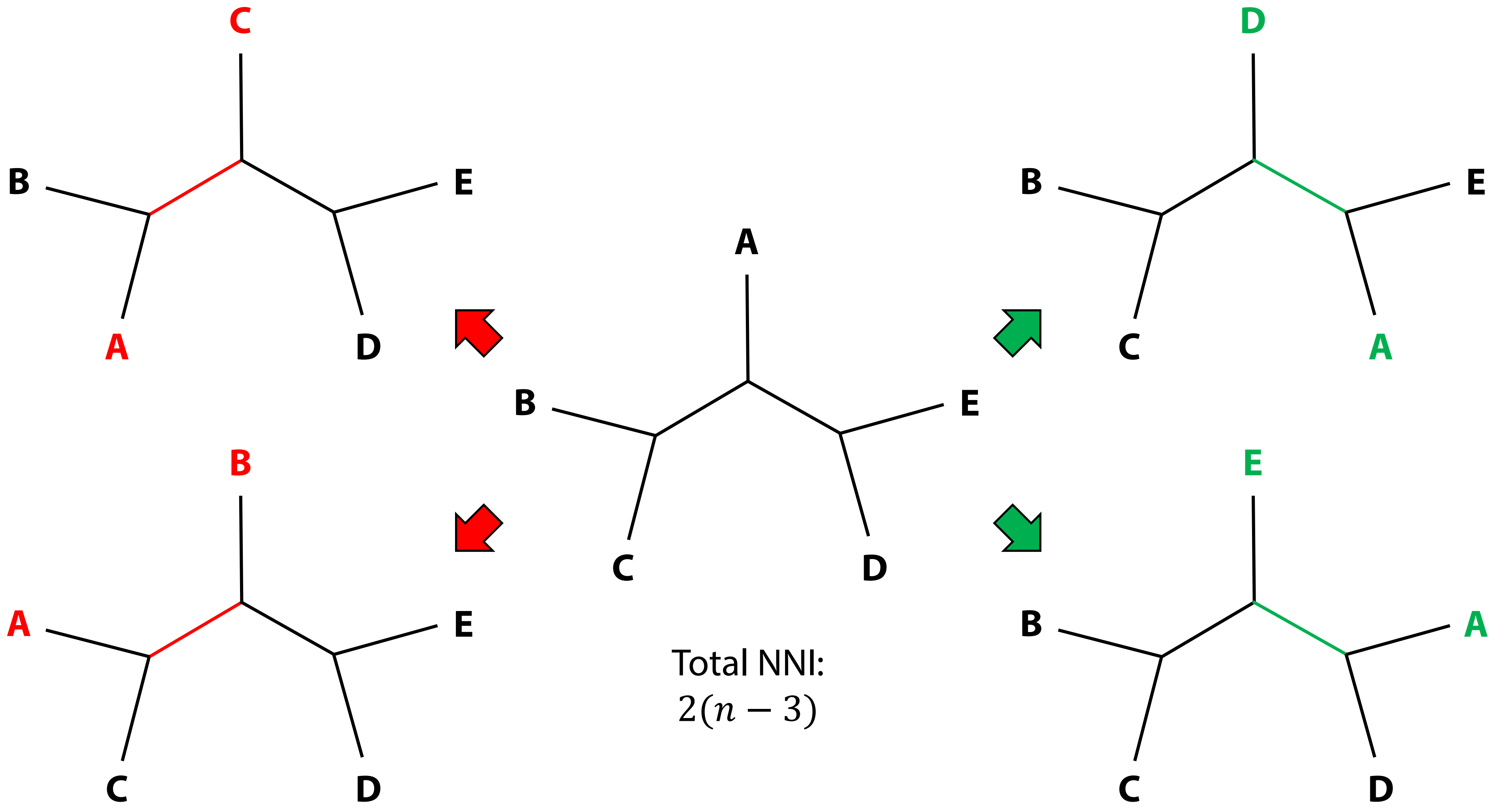
Final tree

if a better tree is found

Nearest Neighbor Interchange (NNI)

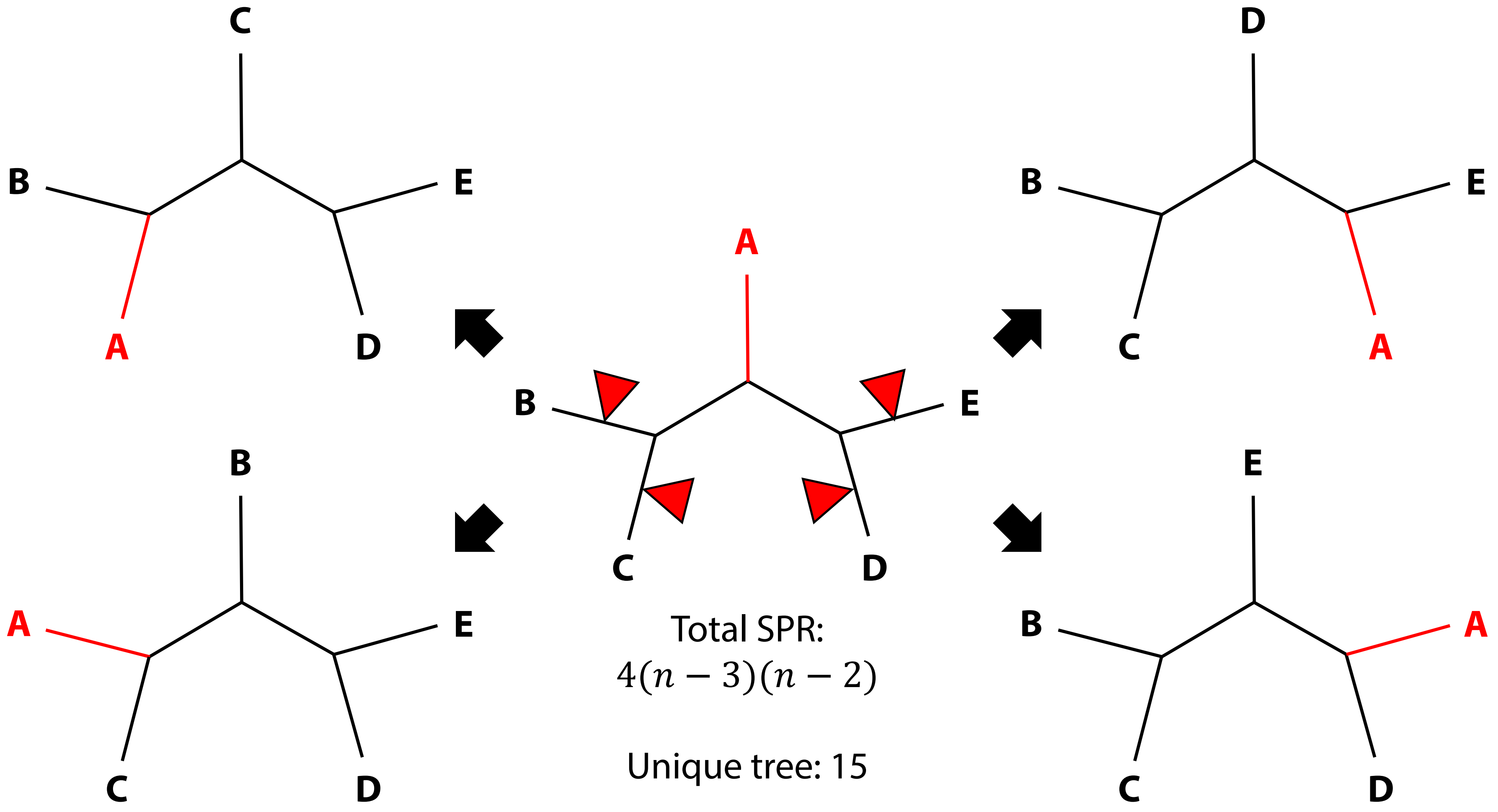


Nearest Neighbor Interchange (NNI)



Total NNI:
$2(n-3)$

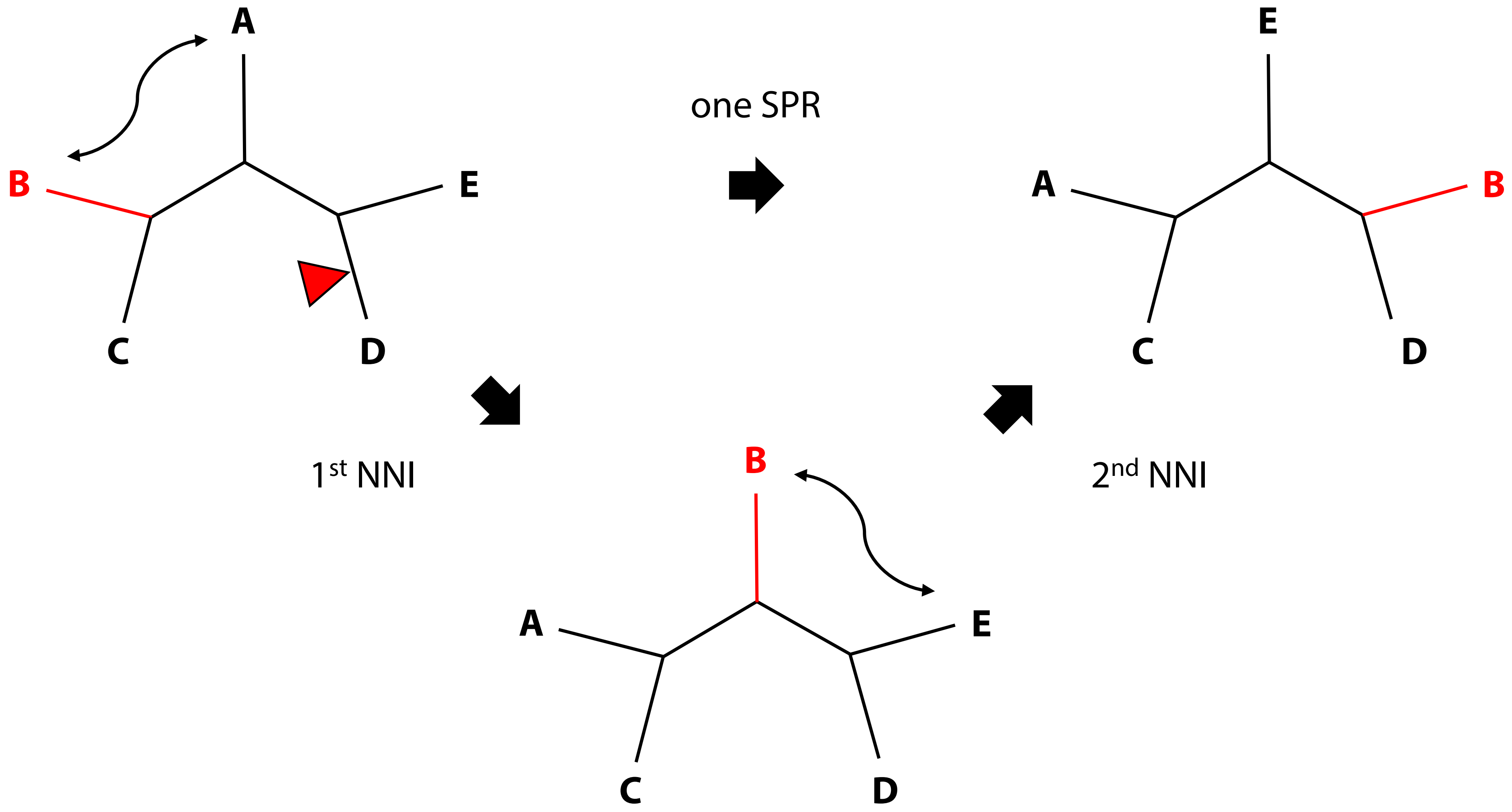
Subtree Pruning and Re-grafting (SPR)



Total SPR:
$4(n-3)(n-2)$

Unique tree: 15

SPR as a chain of NNI

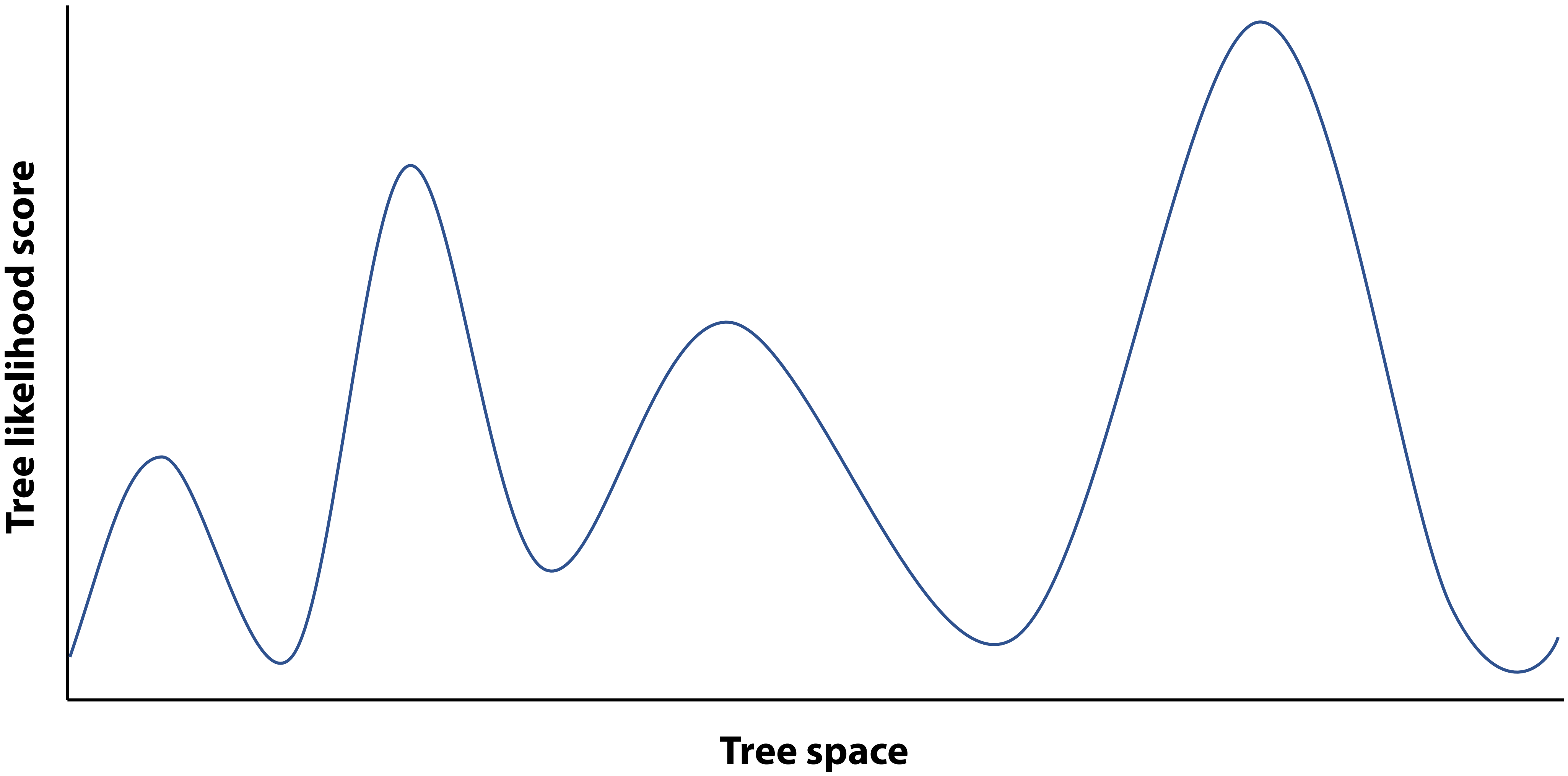


one SPR

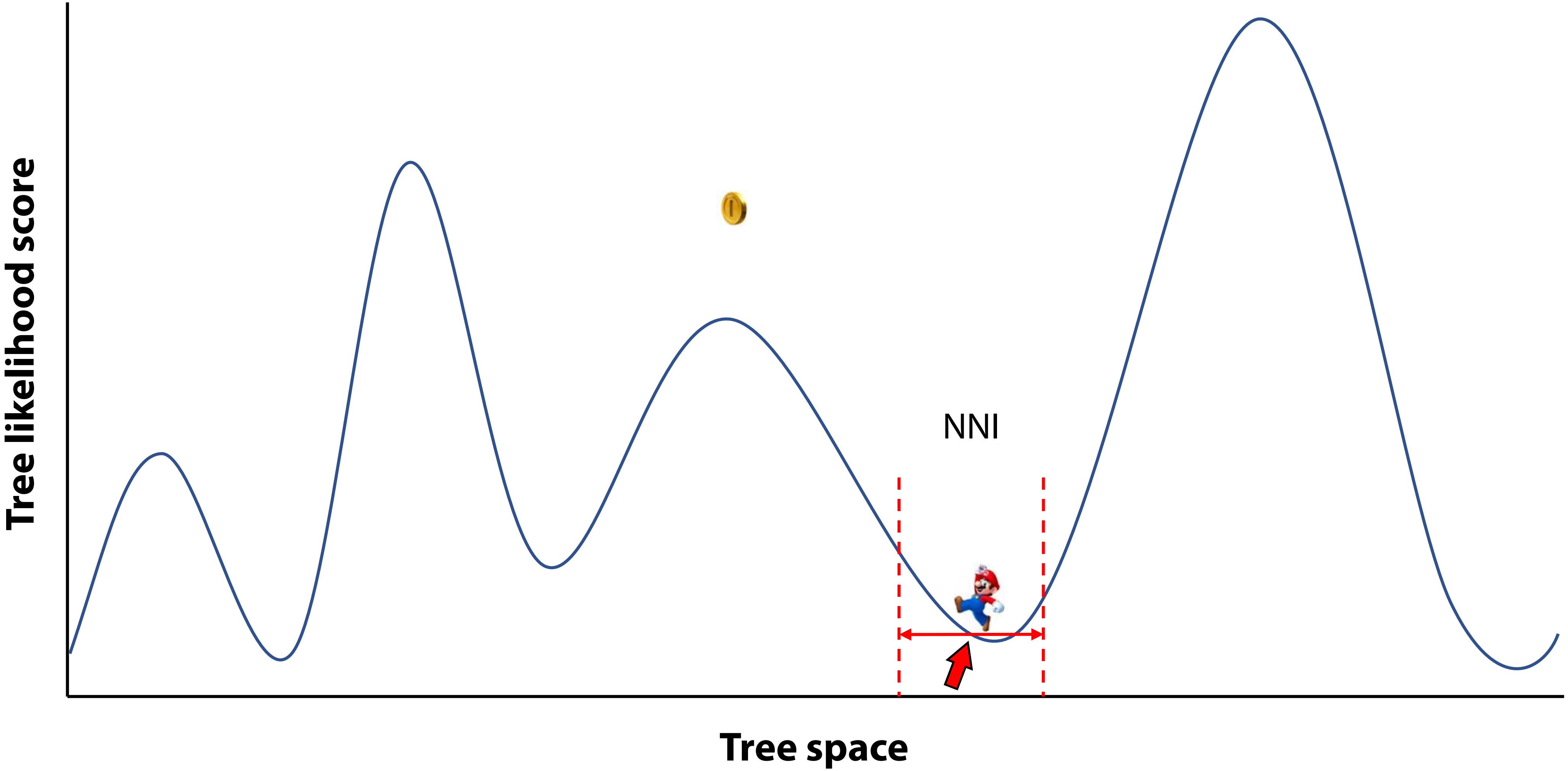
1st NNI

2nd NNI

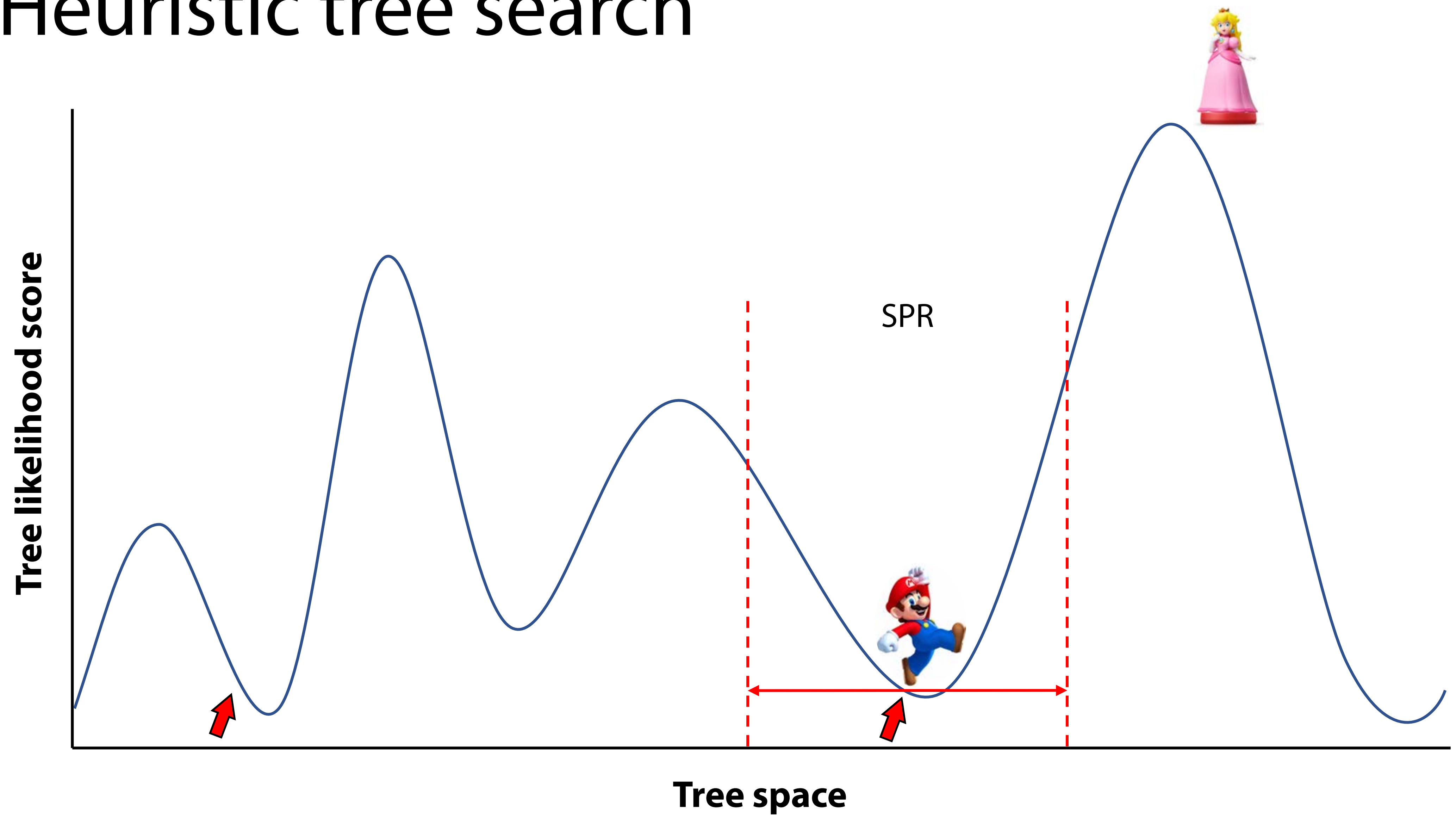
Heuristic search of tree space



Heuristic tree search



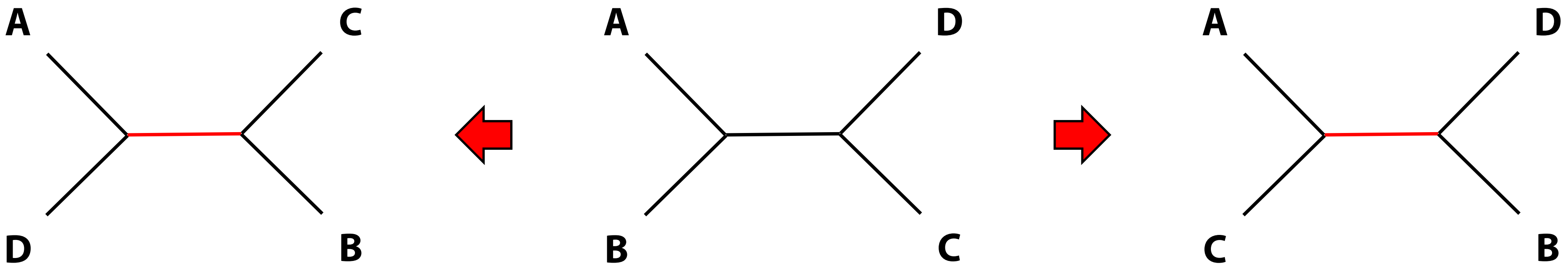
Heuristic tree search



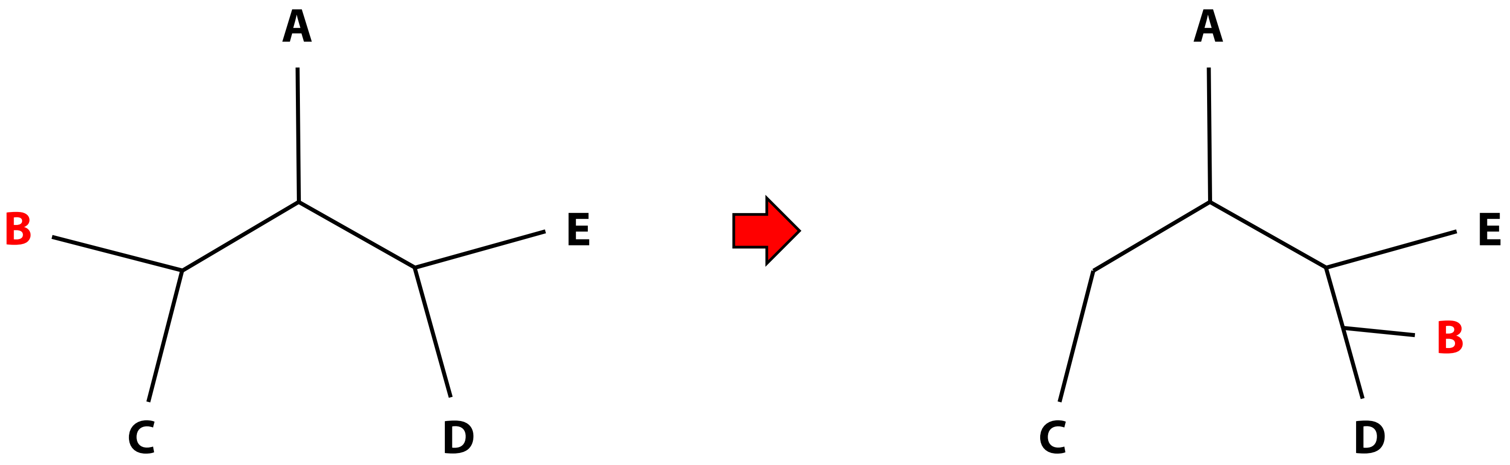
Approximate likelihood calculation

- Global optimization vs. local optimization

NNI



SPR



Approximate likelihood calculation

- Global optimization vs. local optimization

- Exhaustive optimization vs. approximate optimization
  - Diminished return from extra efforts
  - Subsequent topological changes can invalidate extra efforts

Other techniques for fast phylogenetics

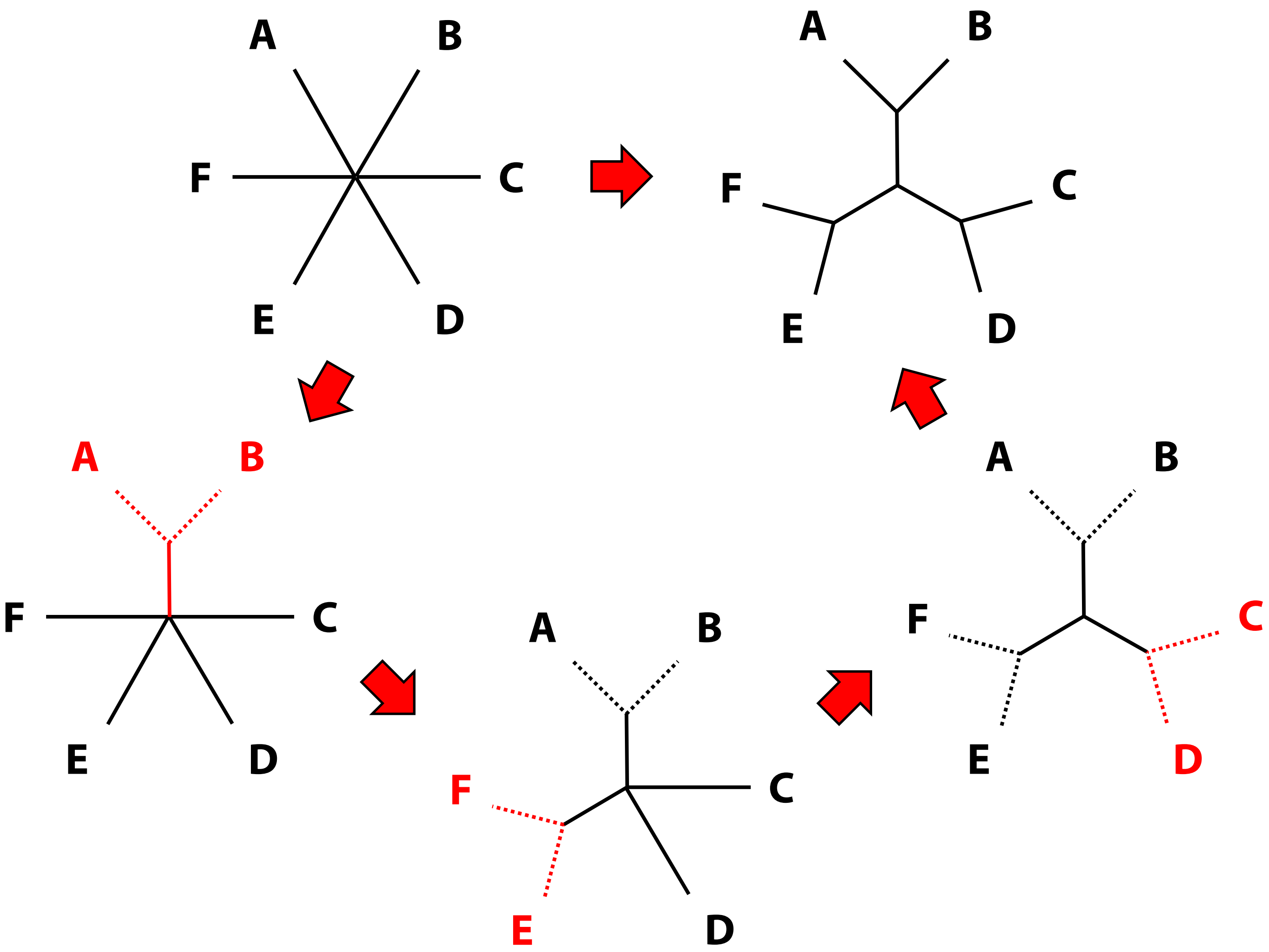
- GAMMA vs CAT
- Fast approaches for node support
- Parallelization
- Memory saving

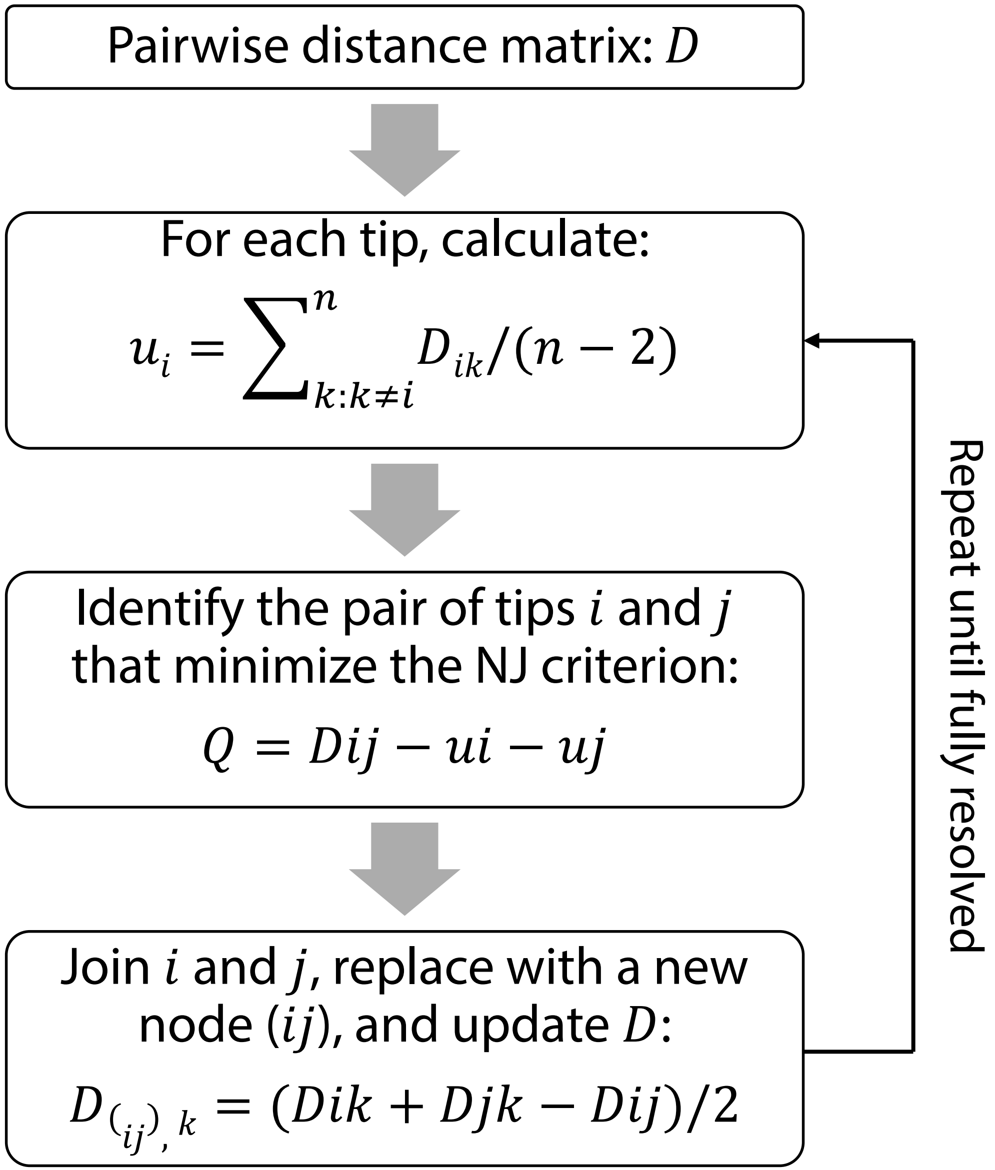
Outline

- Popular fast phylogenetic programs
  - FastTree, RAxML, PhyML, IQ-TREE
  - Main algorithm and development

- Empirical evaluation using recent phylogenomic datasets

NJ – Neighbor Joining





Pairwise distance matrix: D

For each tip, calculate:

$$u_i = \sum_{k:k\neq i}^{n} D_{ik}/(n-2)$$

Identify the pair of tips i and j that minimize the NJ criterion:

$$Q = Dij - ui - uj$$

Join i and j, replace with a new node (ij), and update D:

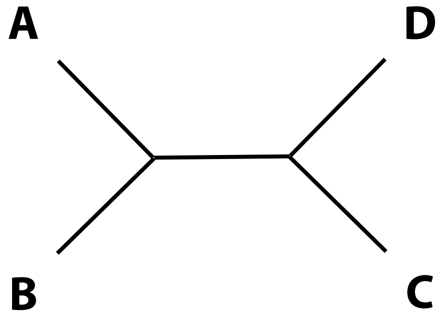
$$D_{(ij),k} = (Dik + Djk - Dij)/2$$

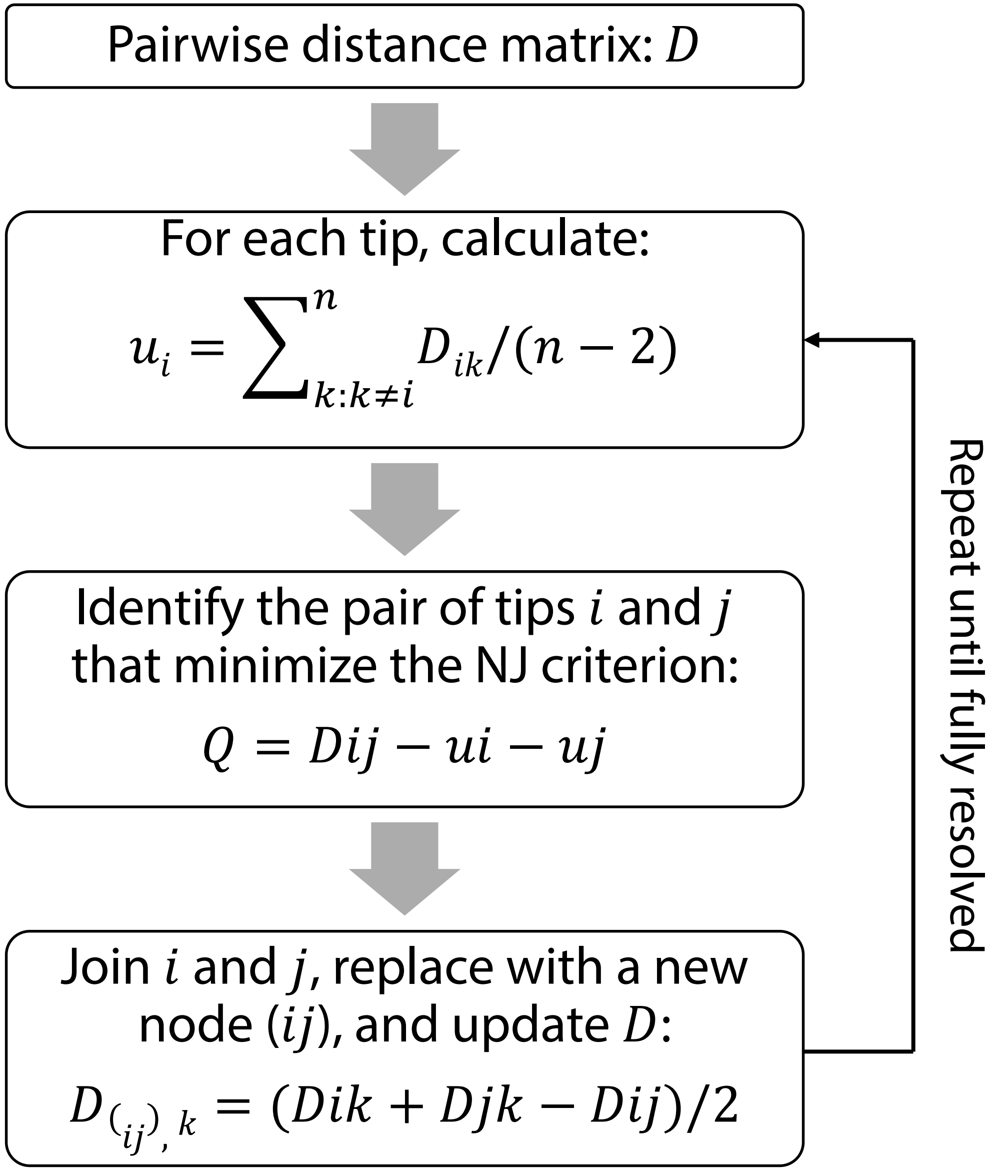
Repeat until fully resolved

What is NJ optimizing?

- NJ is a greedy algorithm optimizing "balance" tree length:

$$L = \sum_{i,j} 2^{1-pij} D_{ij}$$





Pairwise distance matrix: D

For each tip, calculate:

$$u_i = \sum_{k:k\neq i}^{n} D_{ik}/(n-2)$$

Identify the pair of tips i and j that minimize the NJ criterion:

$$Q = Dij - ui - uj$$

Join i and j, replace with a new node (ij), and update D:

$$D_{(ij),k} = (Dik + Djk - Dij)/2$$

Repeat until fully resolved

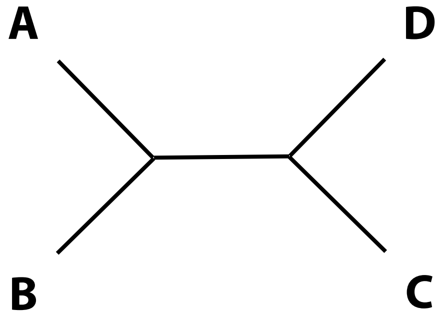
What is NJ optimizing?

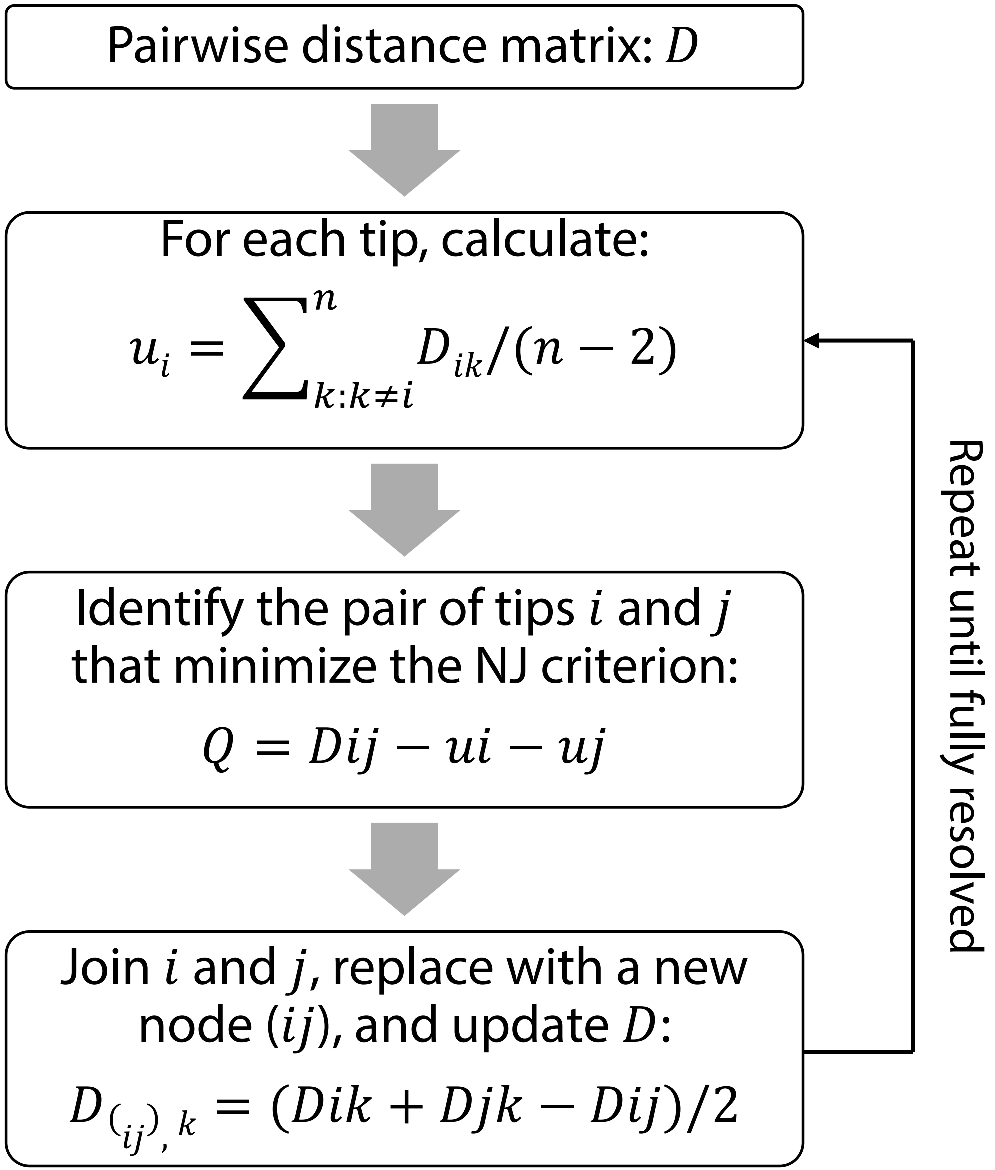
- Balanced Minimum Evolution
  - FastME:

  → stepwise addition starting tree

  → NNI

  → SPR (FastME 2)





Pairwise distance matrix: D

For each tip, calculate:

$$u_i = \sum_{k:k\neq i}^{n} D_{ik}/(n-2)$$

Identify the pair of tips i and j that minimize the NJ criterion:

$$Q = Dij - ui - uj$$

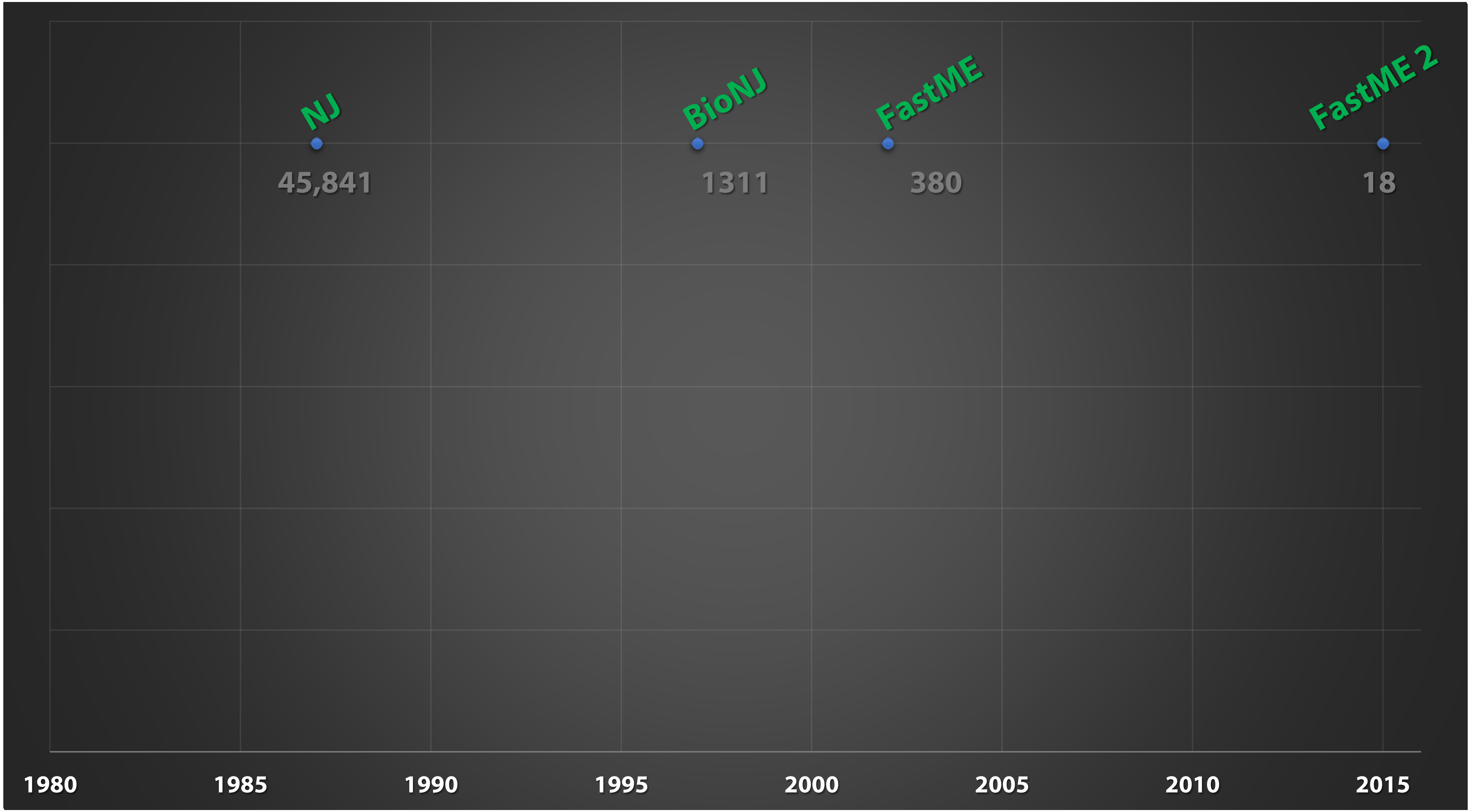
Join i and j, replace with a new node (ij), and update D:

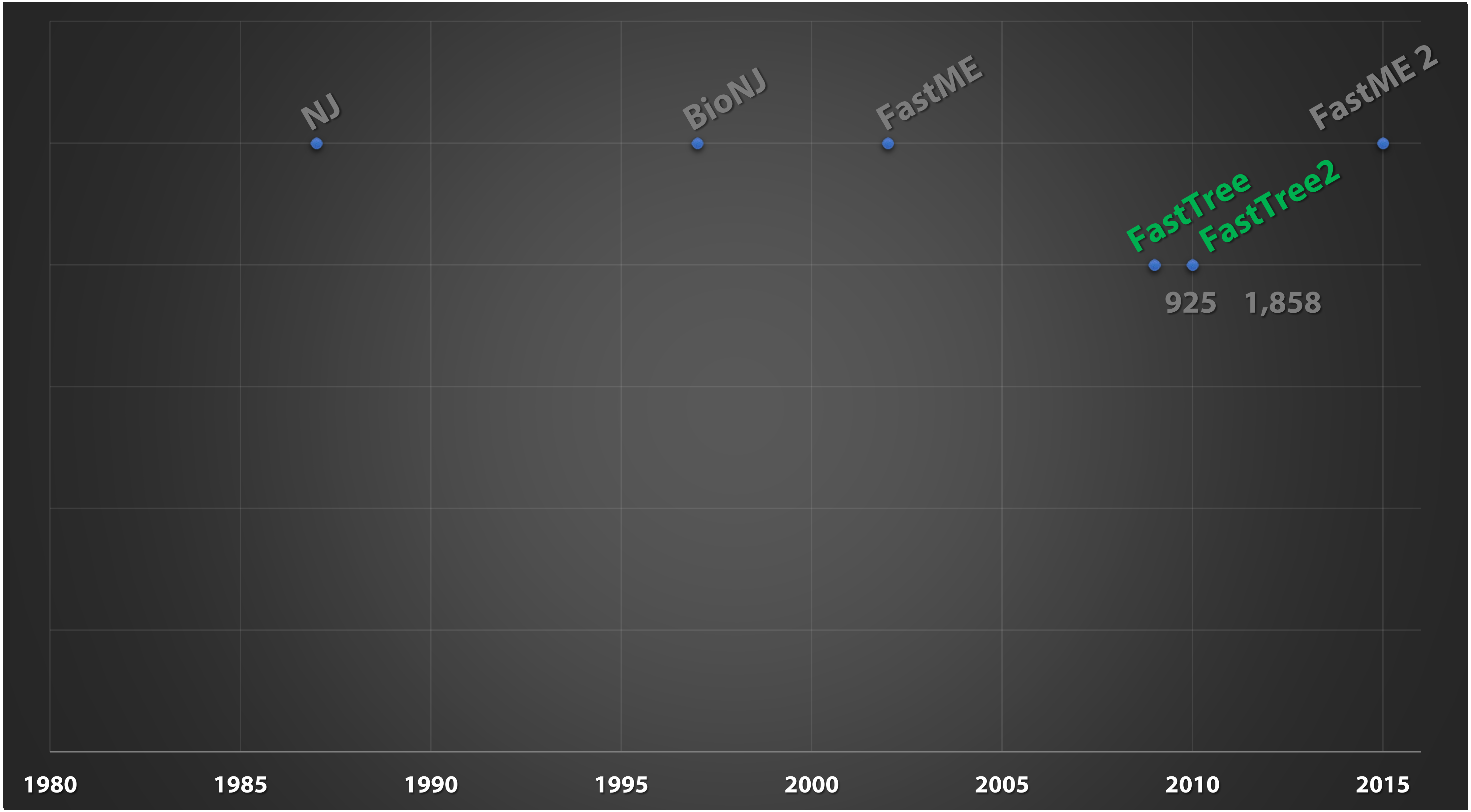
$$D_{(ij),k} = (Dik + Djk - Dij)/2$$

Repeat until fully resolved

Variants of NJ

- BIONJ/WEIHBOR
  - Take variance/co-variance into consideration
- Relaxed Neighbor Joining
  - Join the pair of neighbors firstly found, instead of the one that minimizing Q
- Fast Neighbor Joining
  - Look for a subset of the pairs


NJ
45,841
BioNJ
1311
FastME
380
FastME 2
18
1980
1985
1990
1995
2000
2005
2010
2015


NJ
BioNJ
FastME
FastME 2
FastTree
FastTree2
925
1,858
1980
1985
1990
1995
2000
2005
2010
2015

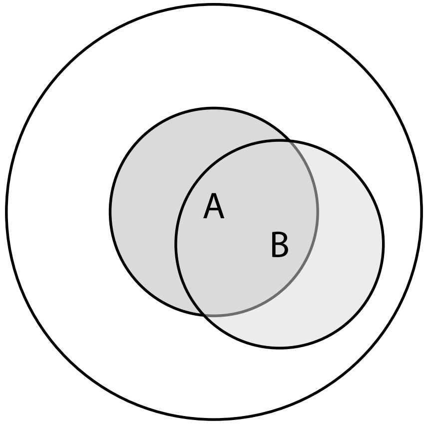
FastTree

- Sequence profile instead of distance matrix
  - Reduce memory requirements
  - Use an average profile of all active nodes to calculate the NJ criterion, instead of actually doing all pairwise computing

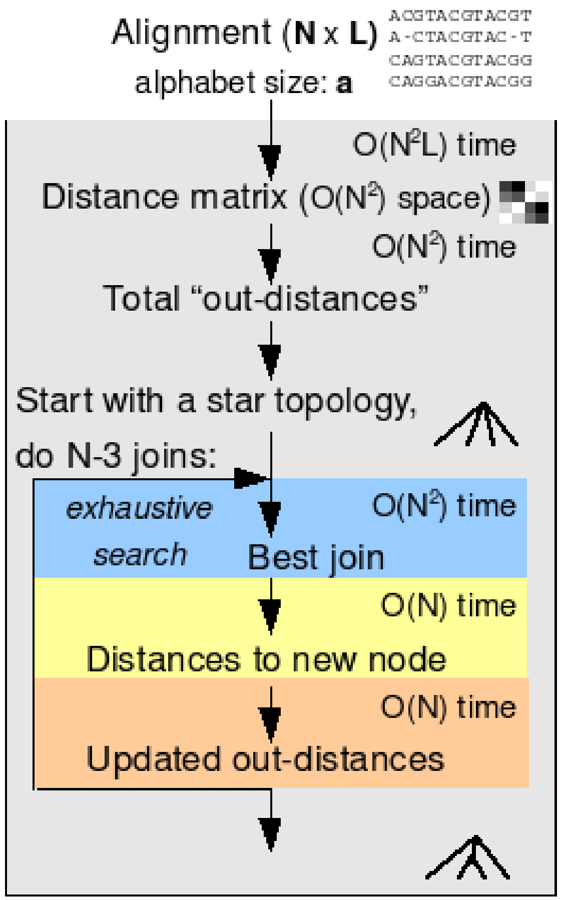
A	C	G	T	A
A	-	C	T	A
C	A	G	T	A
C	A	G	G	A

A	**0.5**	**0.66**	**0**	**0**	**1**
T	**0**	**0**	**0**	**0.75**	**0**
C	**0.5**	**0.33**	**0.25**	**0**	**0**
G	**0**	**0**	**0.75**	**0.25**	**0**
-	**0**	**0.25**	**0**	**0**	**0**

- Three main heuristics
  - Top-hits: neighbors of neighbors are also likely to be neighbors
  - Fast neighbor joining: remembering the best join candidate for each node
  - Relaxed neighbor joining: hill-climbing search for best join



Traditional Neighbor Joining with Bootstrap



Alignment (**N** x **L**)
alphabet size: **a**

ACGTACGTACGT
A-CTACGTAC-T
CAGTACGTACGG
CAGGACGTACGG

O(N^2L) time

Distance matrix (O(N^2) space)

O(N^2) time

Total "out-distances"

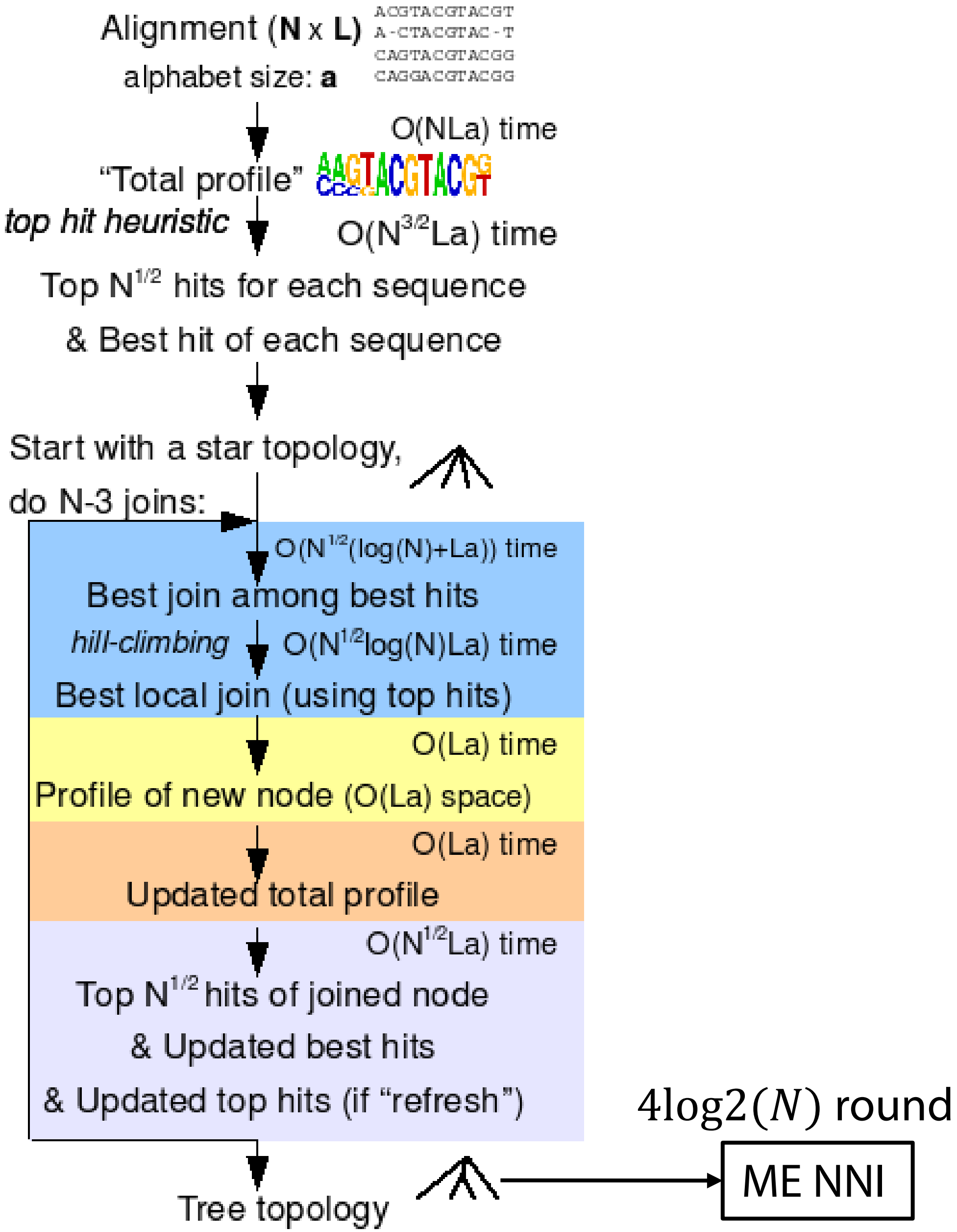
Start with a star topology,
do N-3 joins:

exhaustive search O(N^2) time
Best join

O(N) time
Distances to new node

O(N) time
Updated out-distances

FastTree with Local Support



Alignment (**N** x **L**)
alphabet size: **a**

ACGTACGTACGT
A-CTACGTAC-T
CAGTACGTACGG
CAGGACGTACGG

O(NLa) time

"Total profile"

top hit heuristic O($N^{3/2}$La) time

Top $N^{1/2}$ hits for each sequence
& Best hit of each sequence

Start with a star topology,
do N-3 joins:

O($N^{1/2}$(log(N)+La)) time
Best join among best hits

hill-climbing O($N^{1/2}$log(N)La) time
Best local join (using top hits)

O(La) time
Profile of new node (O(La) space)

O(La) time
Updated total profile

O($N^{1/2}$La) time
Top $N^{1/2}$ hits of joined node
& Updated best hits
& Updated top hits (if "refresh")

Tree topology

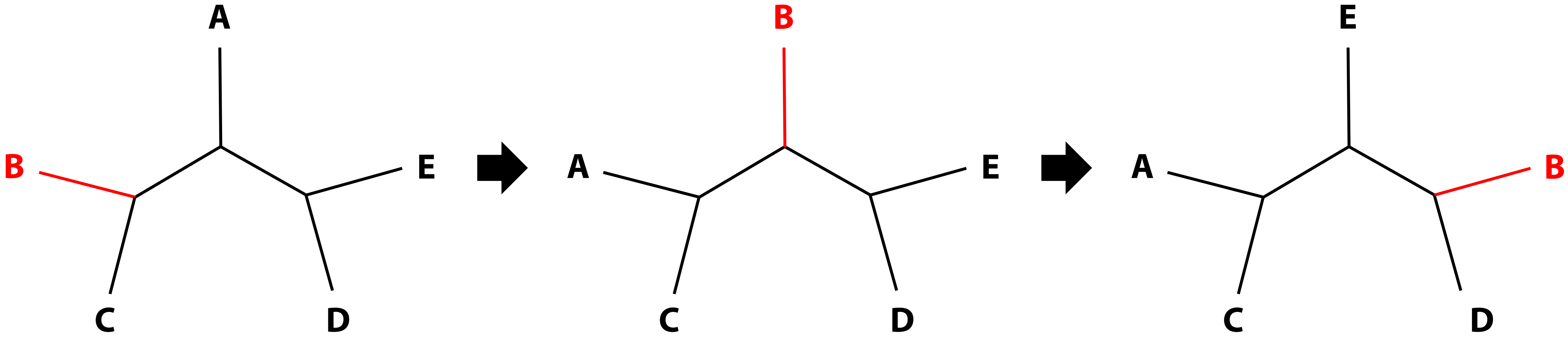
$4\log2(N)$ round

ME NNI

http://www.microbesonline.org/fasttree/FastTreeScheme.png

FastTree2

- ME SPR
  - Extends SPR only along the best path up to length of 10
  - Only two rounds of SPR for each subtree

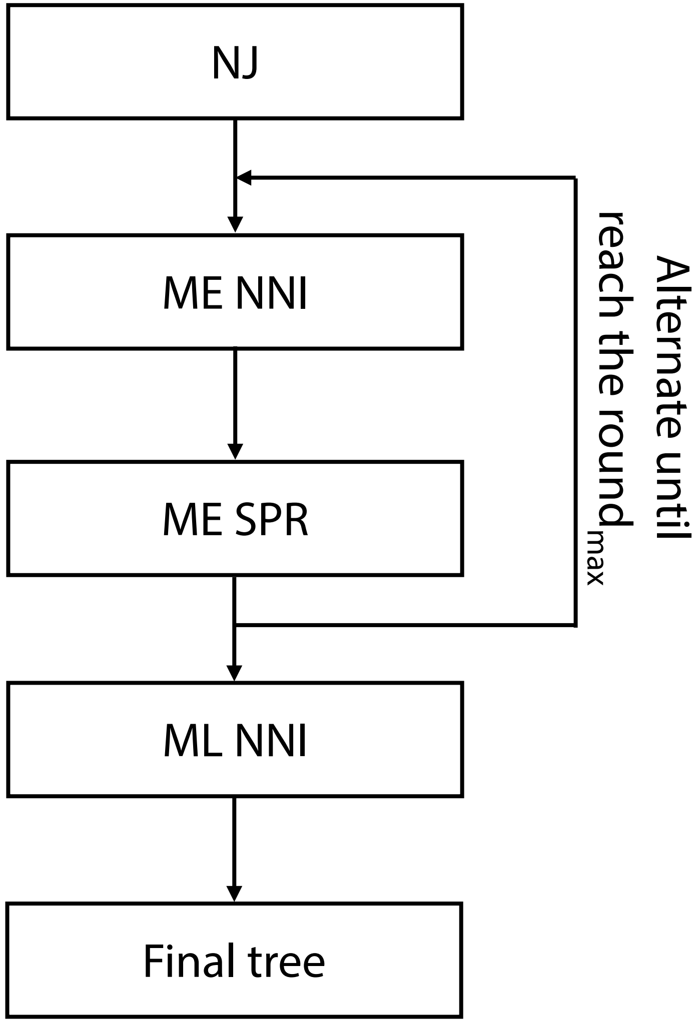


FastTree2

- ME SPR
  - Extends SPR only along the best path up to length of 10
  - Only two rounds of SPR for each subtree

- ML NNI
  - $2\log_2 N$quick rounds + 1 final thorough rounds
  - Heuristics:
    - Branch-length estimation
      - Skip the topology if significantly worse than current tree
    - Star-topology test
      - Pick the winner when one of the quartet tree is significantly better than a star topology
    - Subtree skipping
      - Skip subtrees without significant improvement in recent two rounds

FastTree2



NJ

ME NNI

ME SPR

ML NNI

Final tree

Alternate until reach the round$_{max}$

FastTree: performance

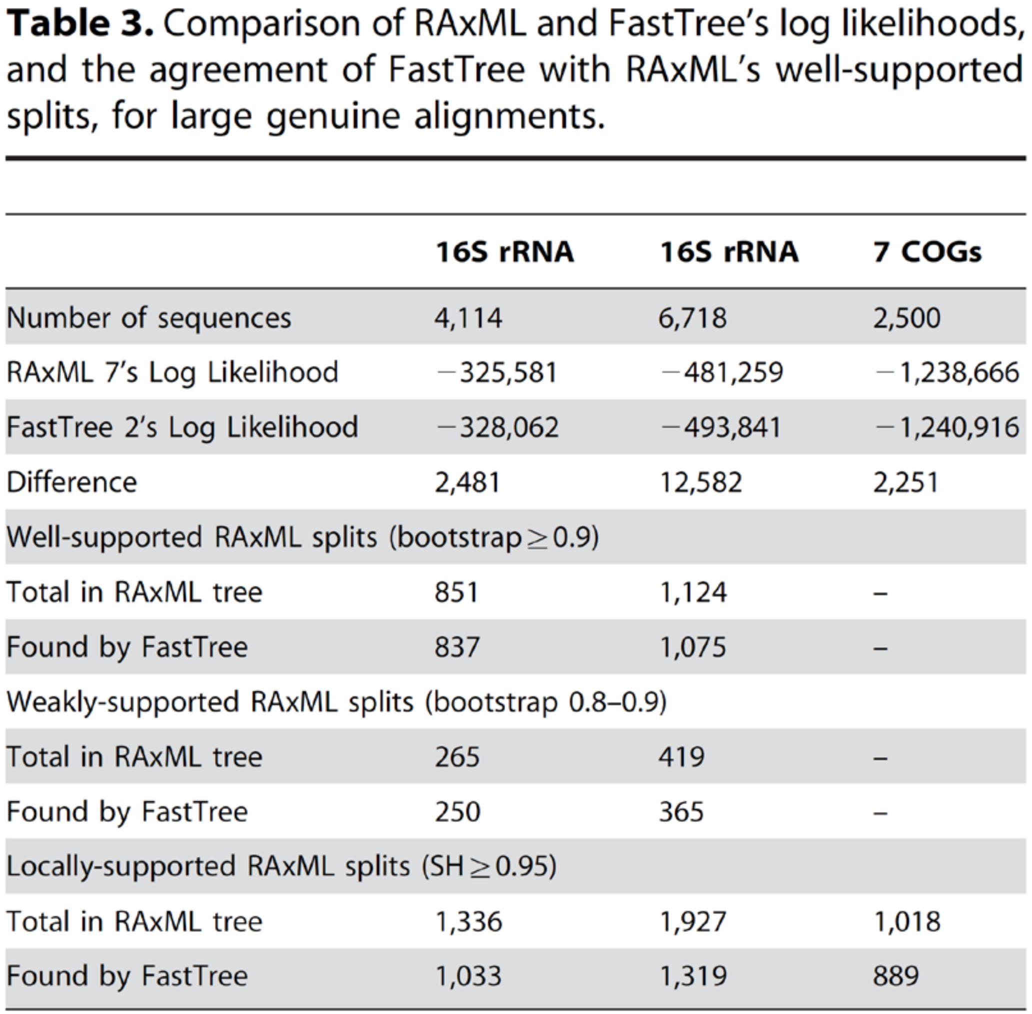


Table 3. Comparison of RAxML and FastTree's log likelihoods, and the agreement of FastTree with RAxML's well-supported splits, for large genuine alignments.

	16S rRNA	**16S rRNA**	**7 COGs**
Number of sequences	4,114	6,718	2,500
RAxML 7's Log Likelihood	−325,581	−481,259	−1,238,666
FastTree 2's Log Likelihood	−328,062	−493,841	−1,240,916
Difference	2,481	12,582	2,251
Well-supported RAxML splits (bootstrap ≥ 0.9)			
Total in RAxML tree	851	1,124	–
Found by FastTree	837	1,075	–
Weakly-supported RAxML splits (bootstrap 0.8–0.9)			
Total in RAxML tree	265	419	–
Found by FastTree	250	365	–
Locally-supported RAxML splits (SH ≥ 0.95)			
Total in RAxML tree	1,336	1,927	1,018
Found by FastTree	1,033	1,319	889

Price et al (2010) PLoS One

FastTree: performance

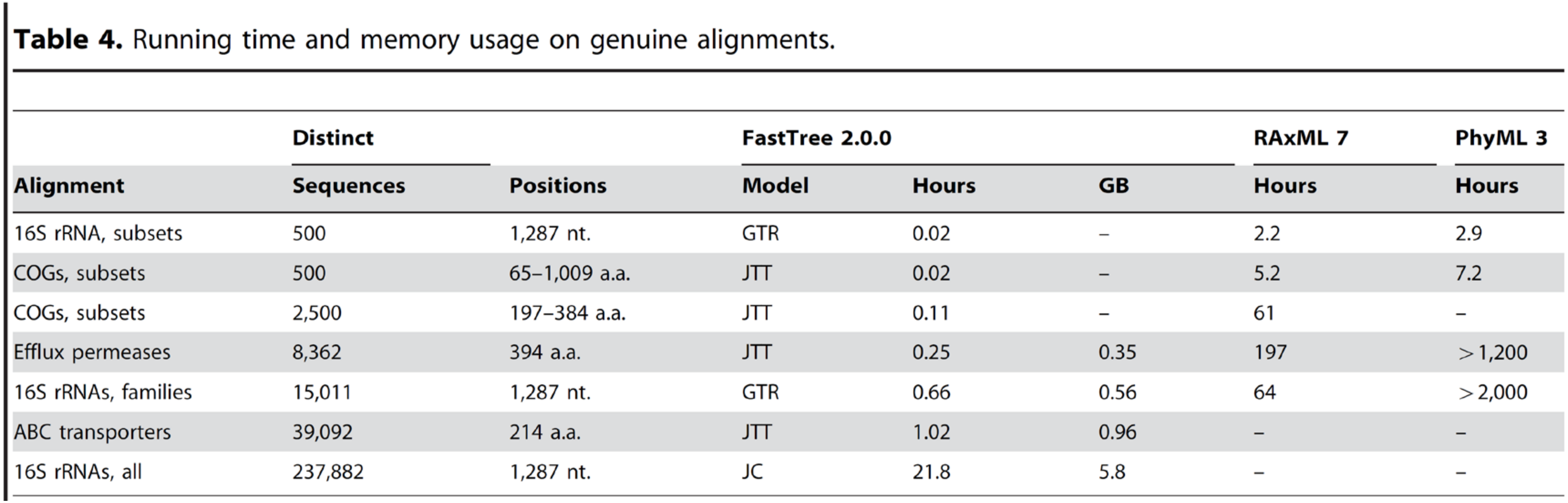
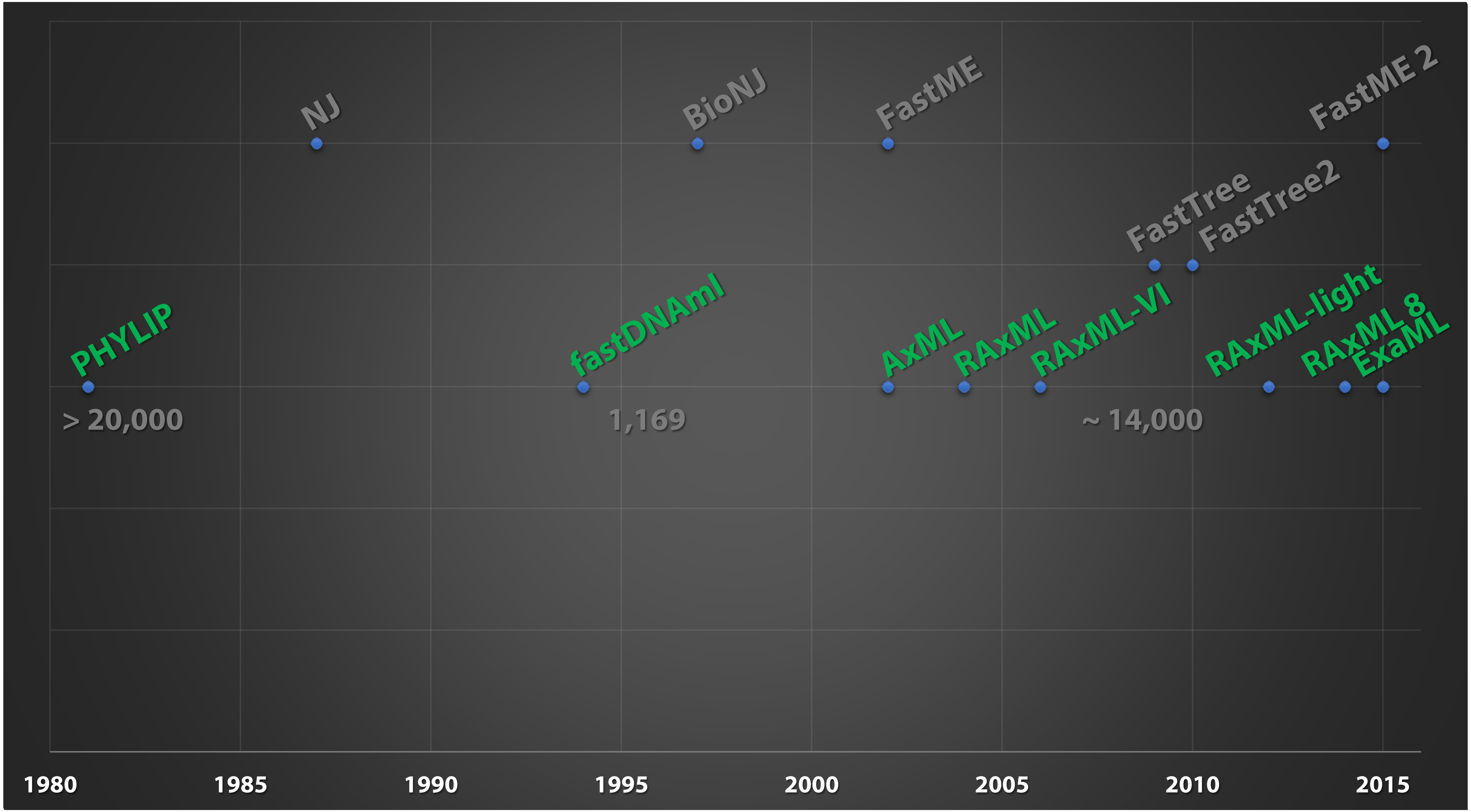


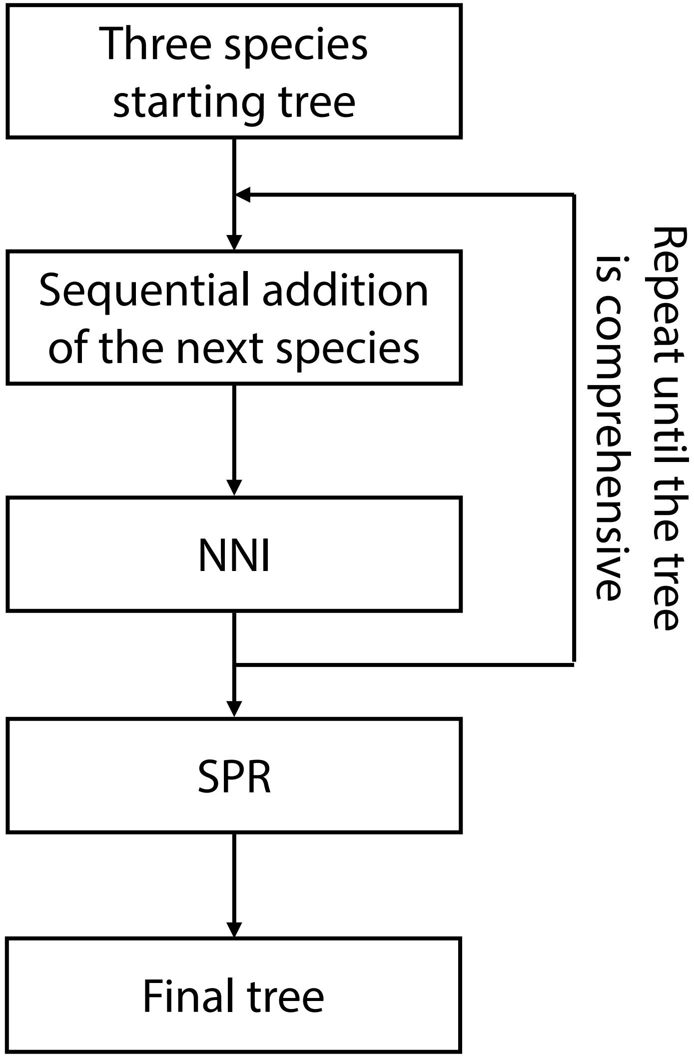
Table 4. Running time and memory usage on genuine alignments.

	Distinct		**FastTree 2.0.0**			**RAxML 7**	**PhyML 3**
Alignment	**Sequences**	**Positions**	**Model**	**Hours**	**GB**	**Hours**	**Hours**
16S rRNA, subsets	500	1,287 nt.	GTR	0.02	–	2.2	2.9
COGs, subsets	500	65–1,009 a.a.	JTT	0.02	–	5.2	7.2
COGs, subsets	2,500	197–384 a.a.	JTT	0.11	–	61	–
Efflux permeases	8,362	394 a.a.	JTT	0.25	0.35	197	>1,200
16S rRNAs, families	15,011	1,287 nt.	GTR	0.66	0.56	64	>2,000
ABC transporters	39,092	214 a.a.	JTT	1.02	0.96	–	–
16S rRNAs, all	237,882	1,287 nt.	JC	21.8	5.8	–	–

Price et al (2010) PLoS One


NJ
BioNJ
FastME
FastME 2
FastTree
FastTree2
PHYLIP
fastDNAml
AxML
RAxML
RAxML-VI
RAxML-light
RAxML 8
ExaML
> 20,000
1,169
~ 14,000
1980
1985
1990
1995
2000
2005
2010
2015

PHYLIP – PHYLogeny Inference Package



Three species starting tree

Sequential addition of the next species

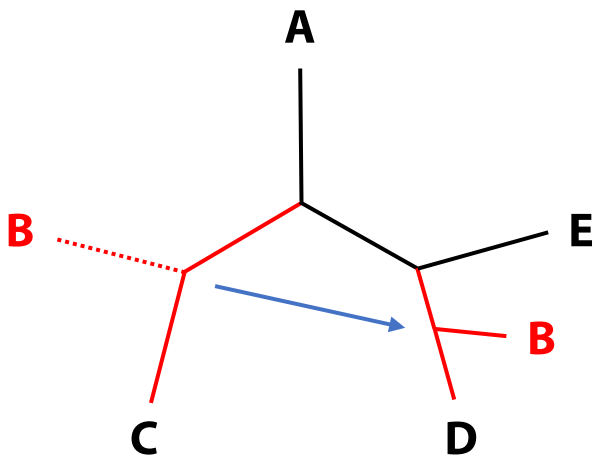
NNI

SPR

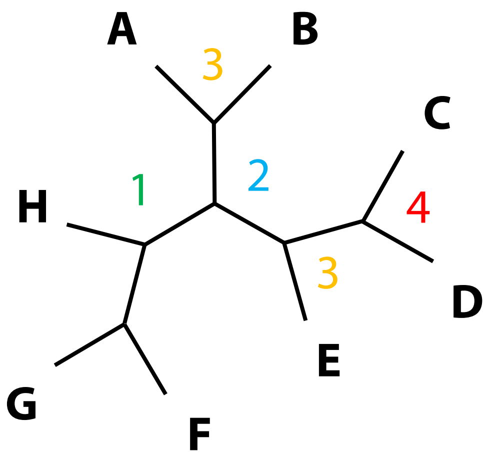
Final tree

Repeat until the tree is comprehensive

fastDNAml



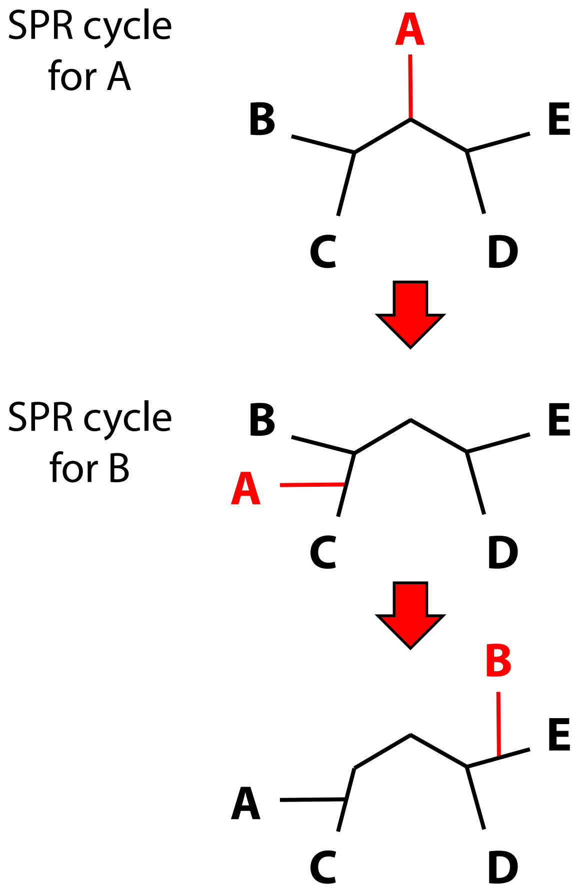
- Algorithm largely the same as PHYLIP
- Less exhaustive branch-length optimization
  - Only optimize the three branches relevant to the sequence addition
- Lazy SPR
  - Only consider re-grafting on branches at most r node away from the pruning position

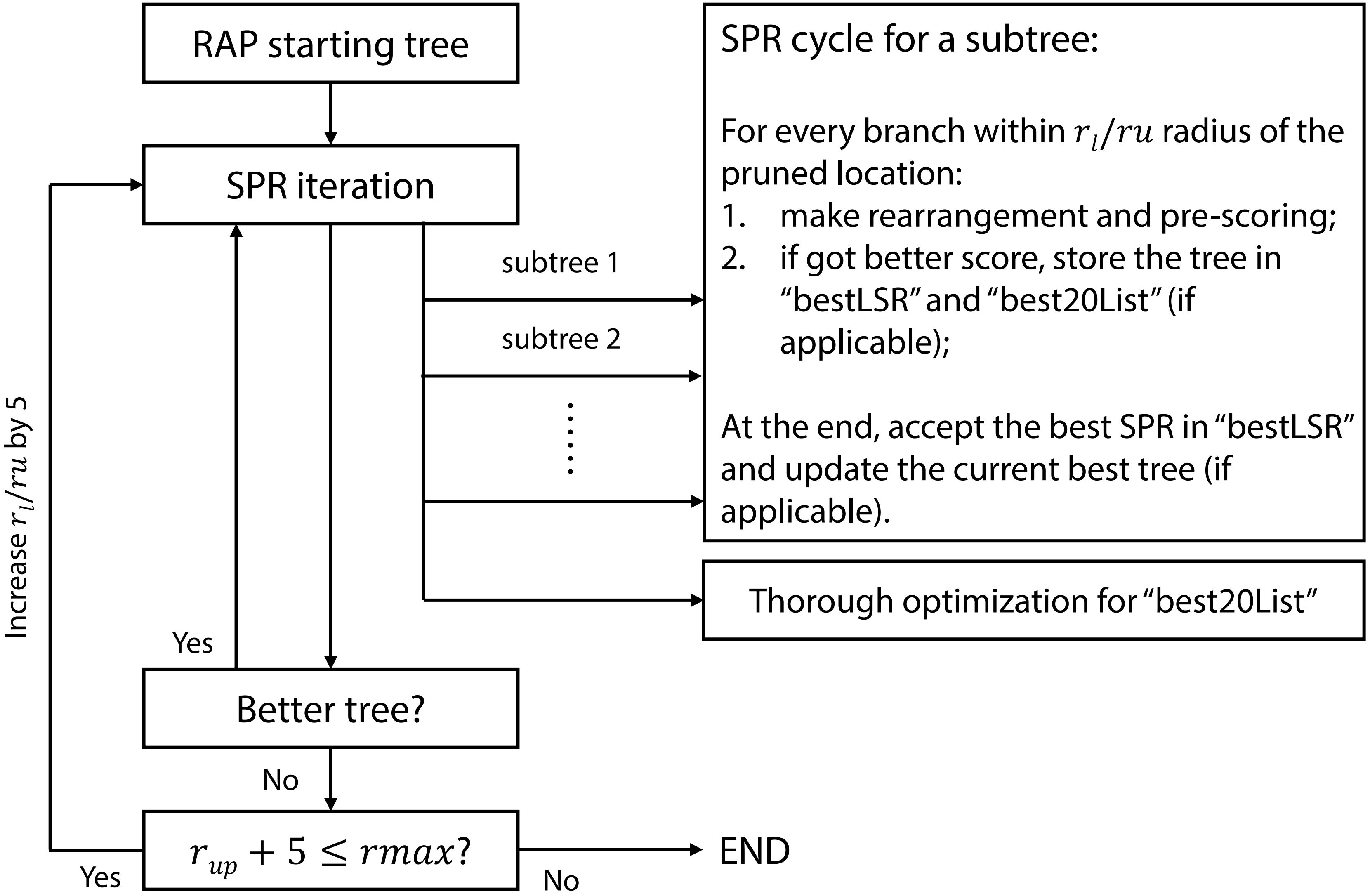


RAxML - Randomized AxML

- Parsimony starting tree:
  - Parsimony is connected with ML
  - Speed and randomization!

- Lazy SPR
  - Only pre-scoring during one SPR iteration
  - SPRs leading to better scores are immediately implied
  - Dynamic adjustment of Lazy SPR radius





RAP starting tree

SPR iteration

subtree 1

subtree 2

SPR cycle for a subtree:

For every branch within r_l/ru radius of the pruned location:

1. make rearrangement and pre-scoring;
2. if got better score, store the tree in "bestLSR" and "best20List" (if applicable);

At the end, accept the best SPR in "bestLSR" and update the current best tree (if applicable).

Thorough optimization for "best20List"

Better tree?

Yes

No

$r_{up} + 5 \leq rmax?$

Yes

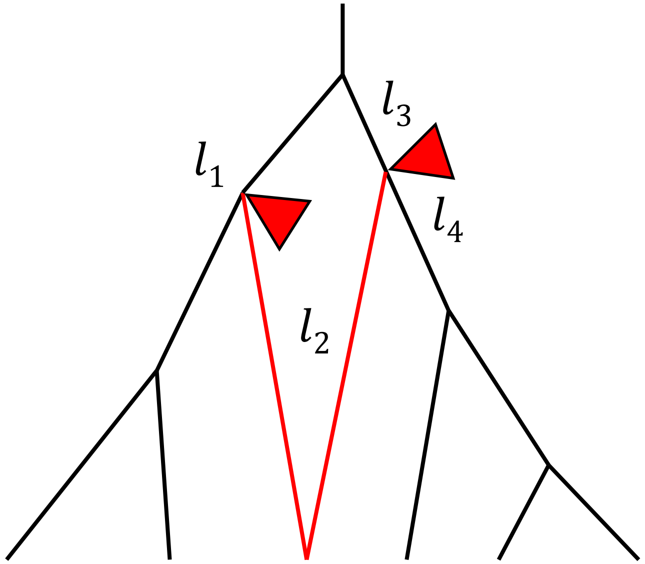
Increase r_l/ru by 5

No

END

RAxML: improvements

- Optimized data structure
  - rearrangement descriptor
  - (pruned position, re-grafting position, l_1, l_2, l_3, l_4)



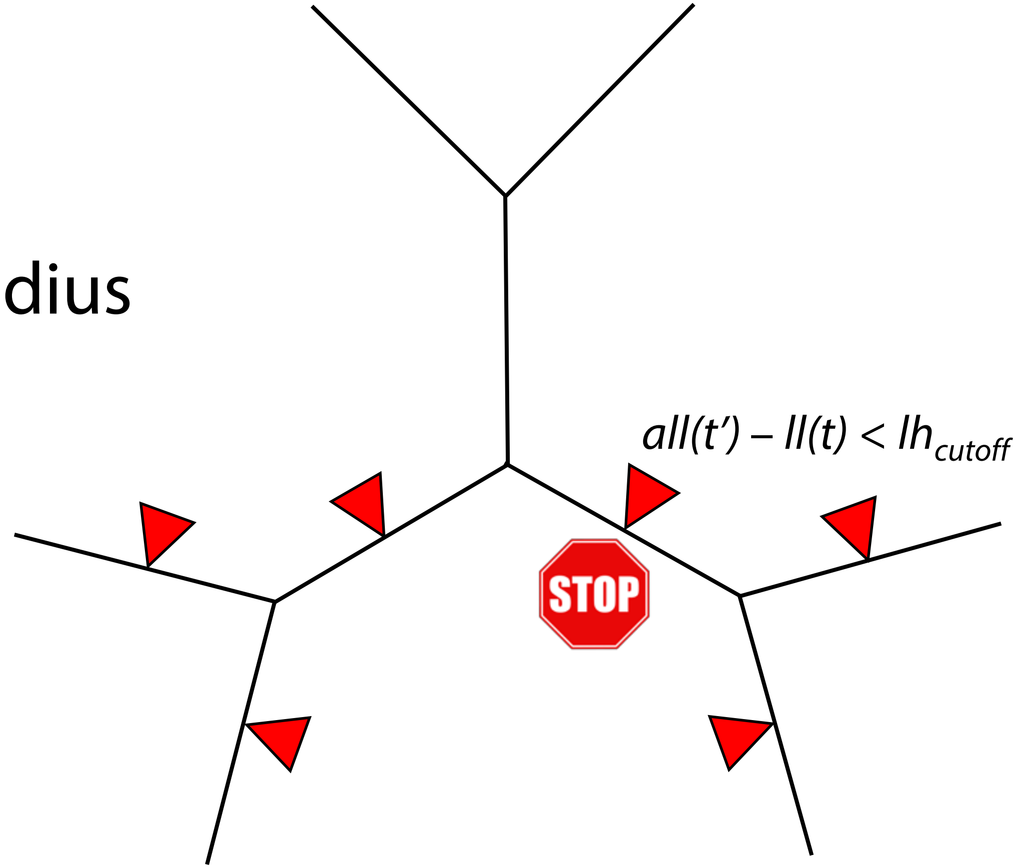
RAxML: improvements

- Optimized data structure

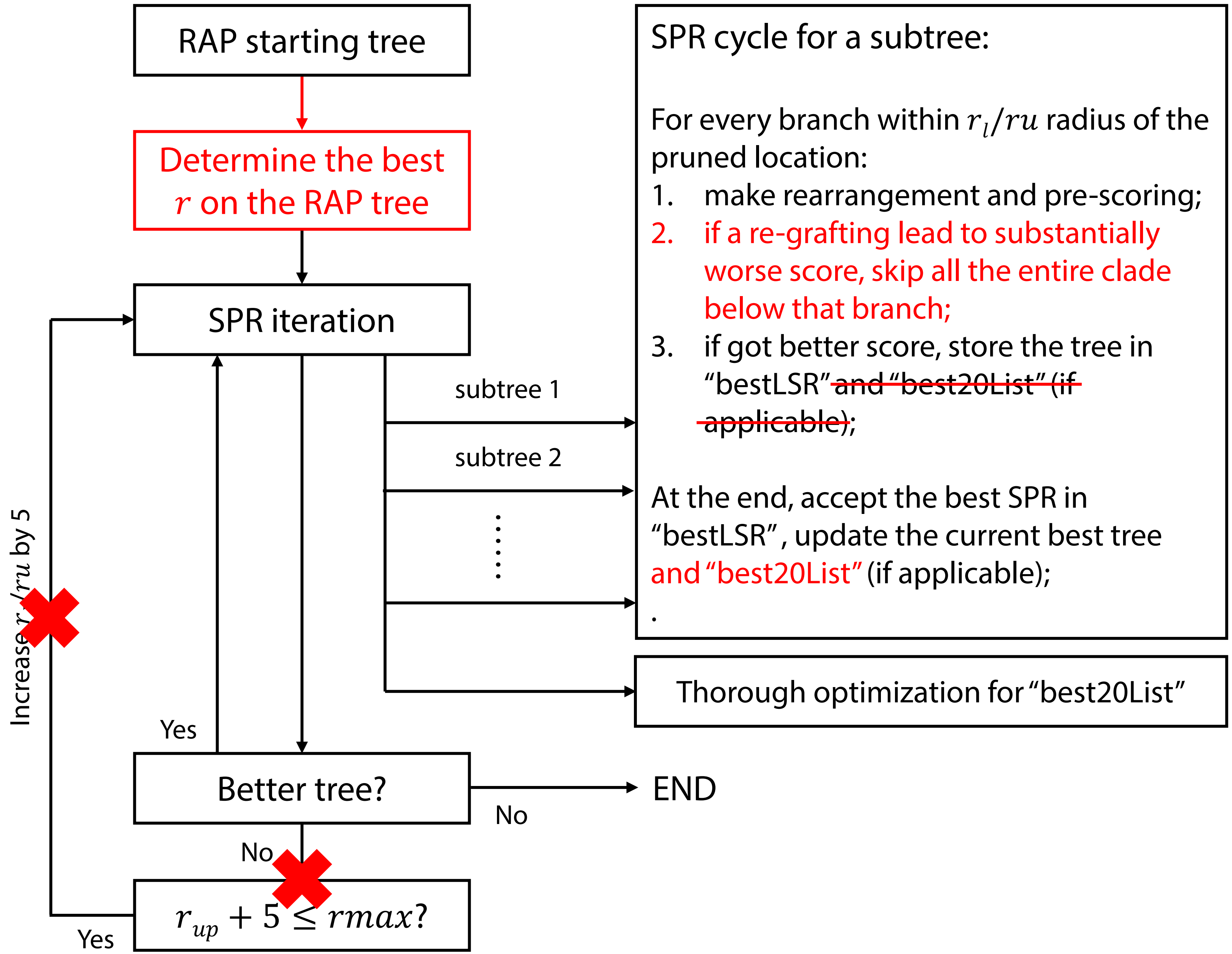
- Adaptive determination of LSR radius
  - SPR on the RAP starting tree with different r value
  - Choose the smallest r giving rise to the best score

RAxML: improvements

- Optimized data structure
- Adaptive determination of LSR radius
- "Subtree skipping"
  - lh_{cutoff} determined dynamically during SPR cycles



~2.5x speed up



RAP starting tree

Determine the best r on the RAP tree

SPR iteration

subtree 1

subtree 2

SPR cycle for a subtree:

For every branch within r_l/ru radius of the pruned location:

1. make rearrangement and pre-scoring;
2. if a re-grafting lead to substantially worse score, skip all the entire clade below that branch;
3. if got better score, store the tree in “bestLSR” ~~and “best20List” (if applicable)~~;

At the end, accept the best SPR in “bestLSR”, update the current best tree and “best20List” (if applicable);

.

Thorough optimization for “best20List”

Better tree?

Yes

No

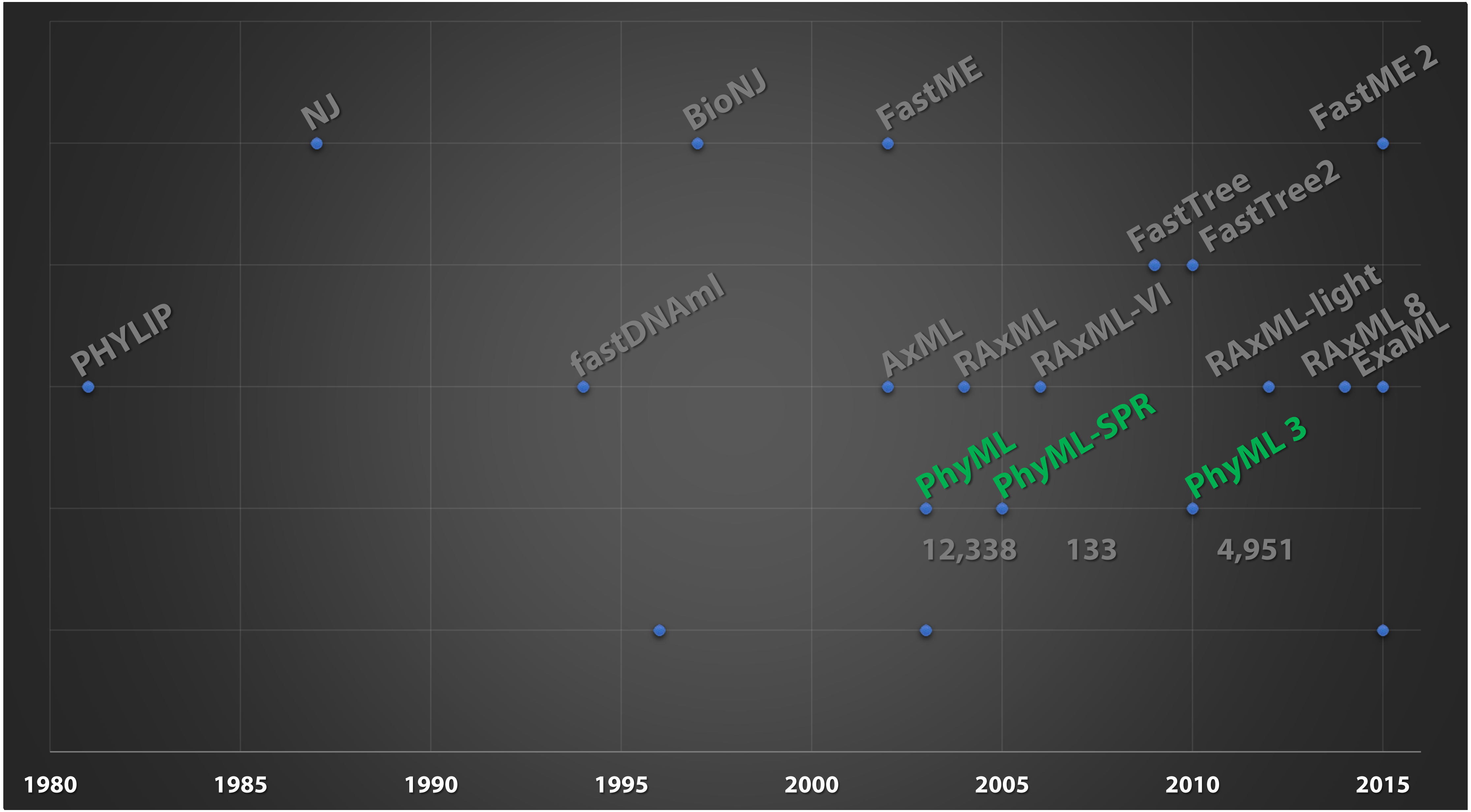
END

No

$r_{up} + 5 \leq rmax?$

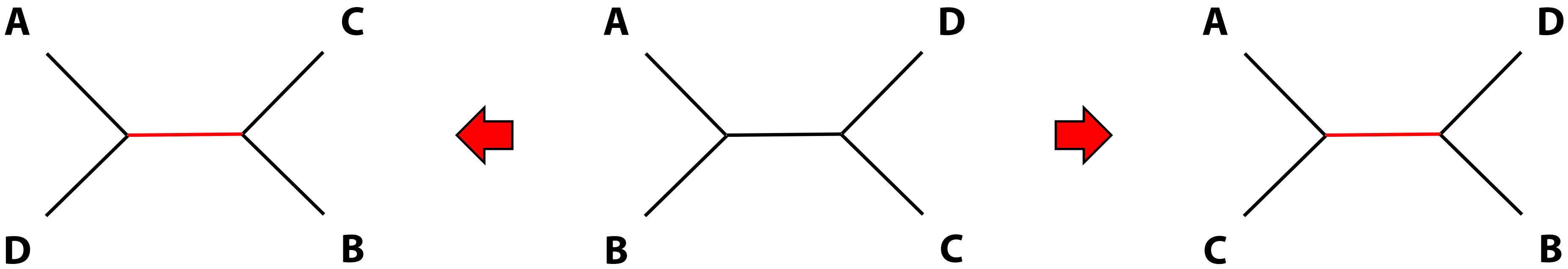
Yes

Increase r_l/ru by 5


NJ
BioNJ
FastME
FastME 2
FastTree
FastTree2
PHYLIP
fastDNAml
AxML
RAxML
RAxML-VI
RAxML-light
RAxML 8
ExaML
PhyML
PhyML-SPR
PhyML 3
12,338
133
4,951
1980
1985
1990
1995
2000
2005
2010
2015

PhyML

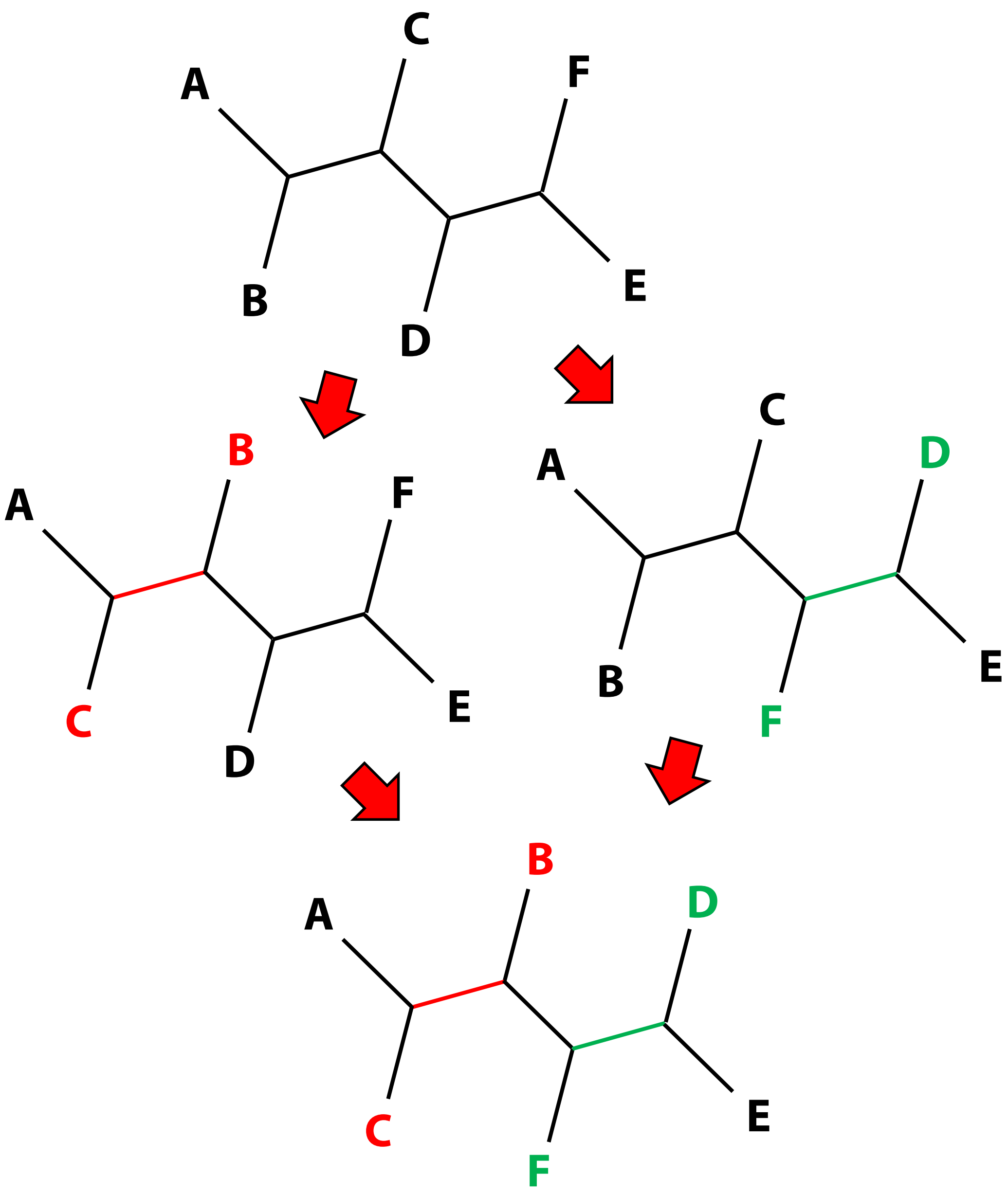
- Single internal branch re-optimization



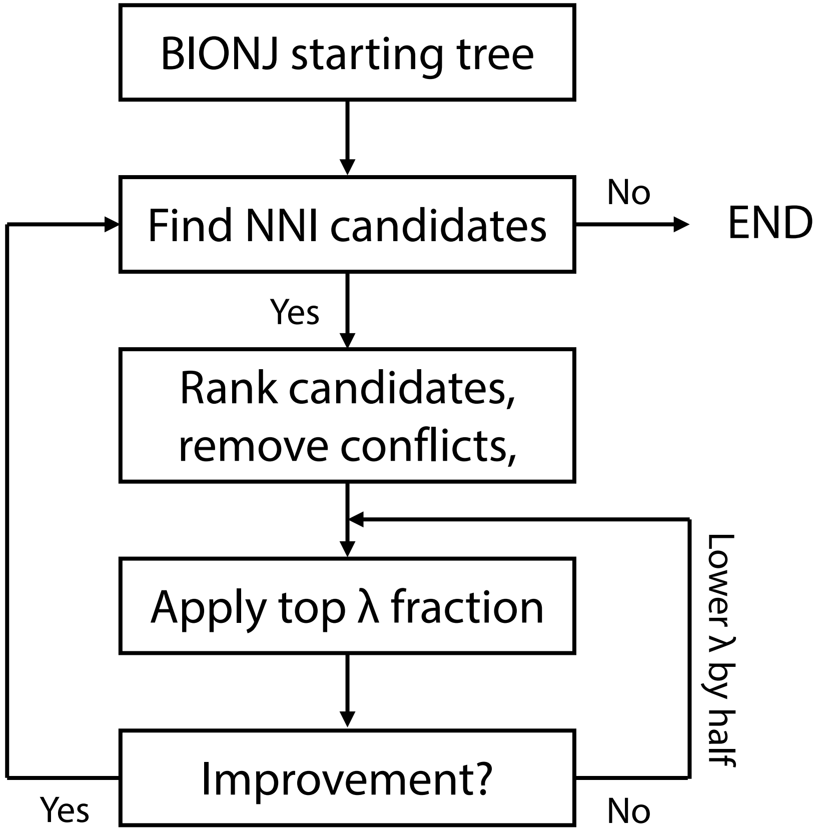
PhyML

- Single internal branch re-optimization

- Simultaneous topology modifications
    - Rank all NNI candidates by LLS
    - Remove conflicting candidates
    - Apply the top λ fraction simultaneously
    - If get worse score, lower λ by half; keep going until the best one



PhyML



BIONJ starting tree

Find NNI candidates

No

END

Yes

Rank candidates,
remove conflicts,

Apply top λ fraction

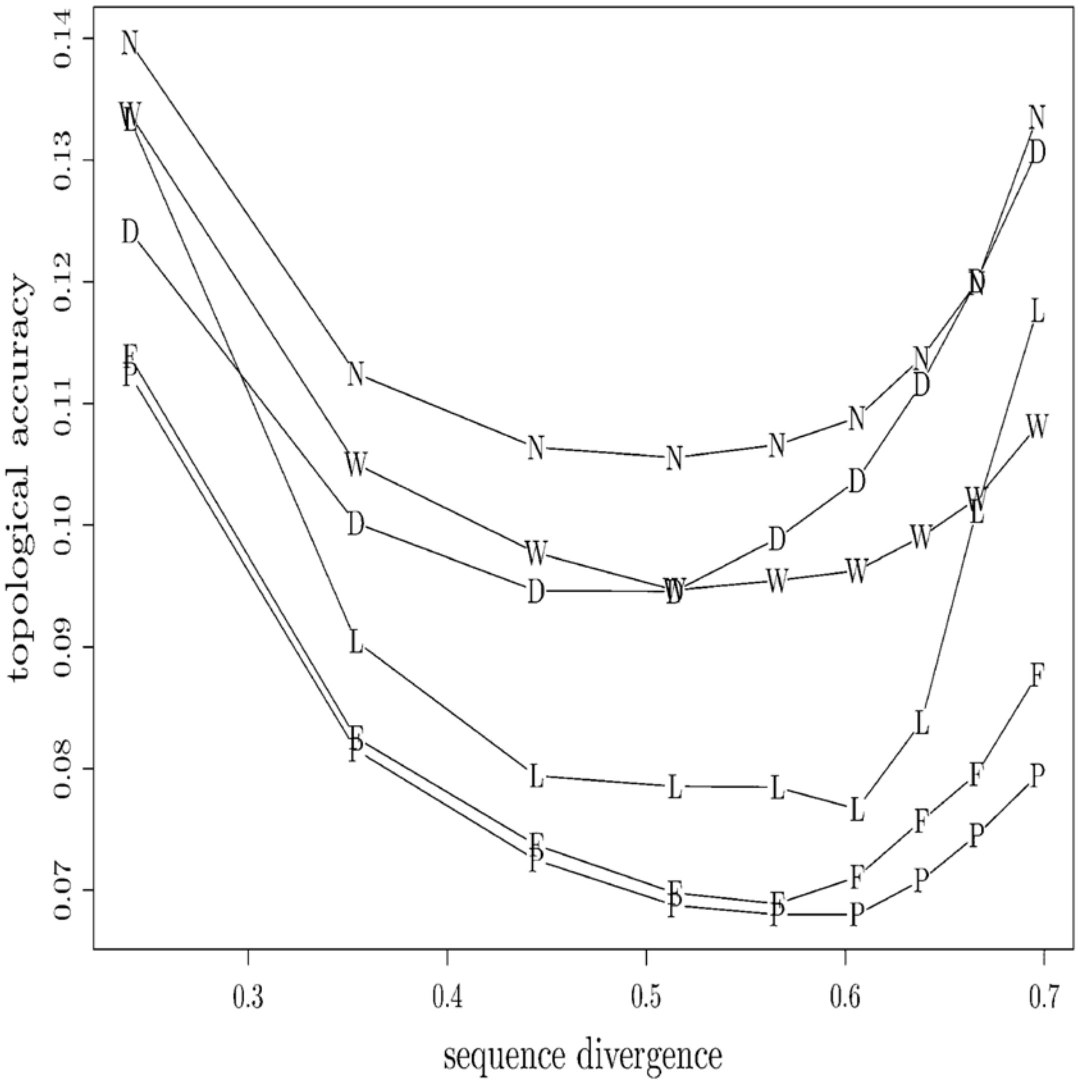
Improvement?

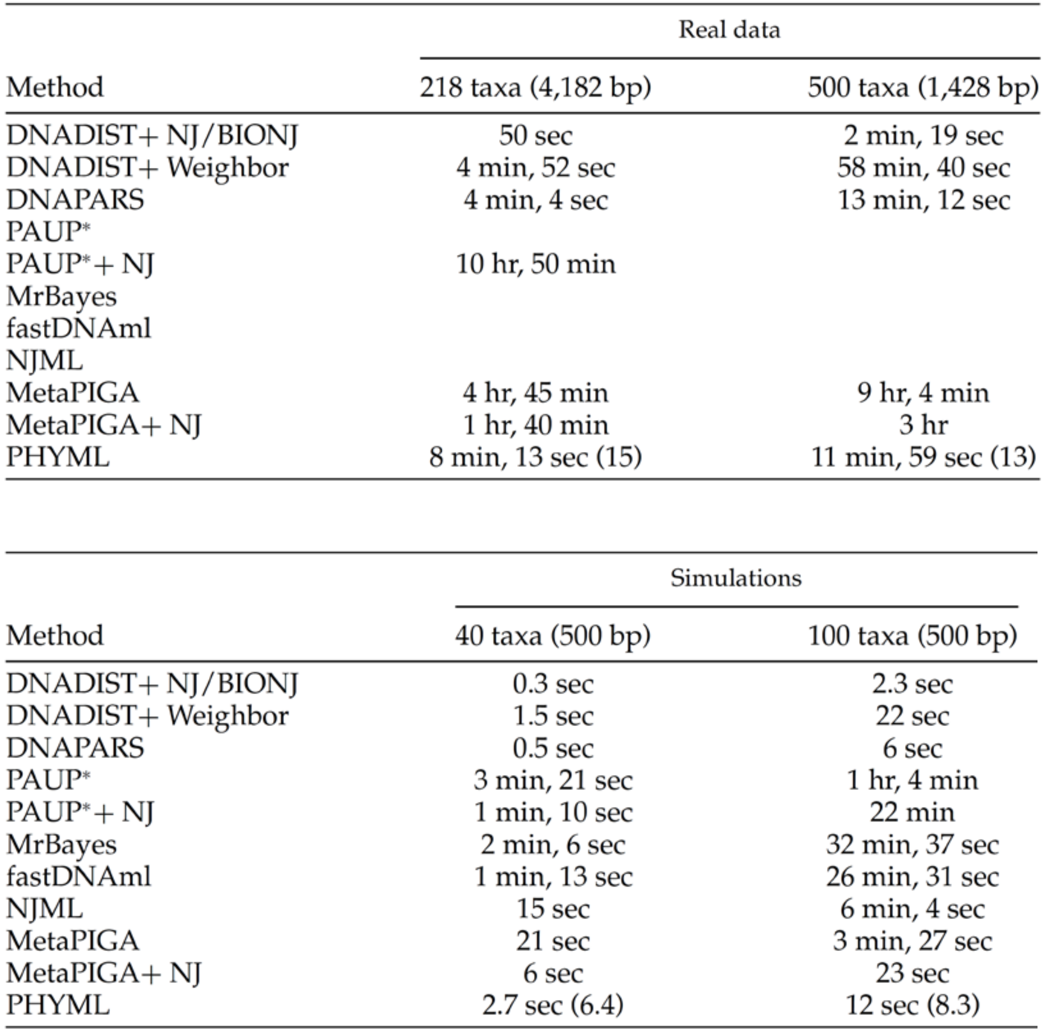
Yes

No

Lower λ by half

PhyML: performance





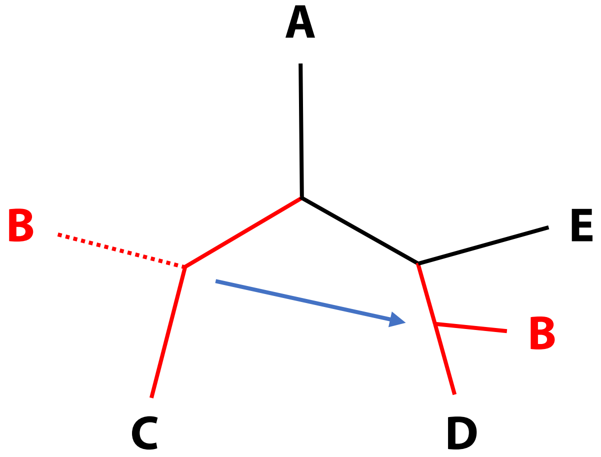
Method	Real data 218 taxa (4,182 bp)	 500 taxa (1,428 bp)
DNADIST+ NJ/BIONJ	50 sec	2 min, 19 sec
DNADIST+ Weighbor	4 min, 52 sec	58 min, 40 sec
DNAPARS	4 min, 4 sec	13 min, 12 sec
PAUP*		
PAUP*+ NJ	10 hr, 50 min	
MrBayes		
fastDNAml		
NJML		
MetaPIGA	4 hr, 45 min	9 hr, 4 min
MetaPIGA+ NJ	1 hr, 40 min	3 hr
PHYML	8 min, 13 sec (15)	11 min, 59 sec (13)

Method	Simulations 40 taxa (500 bp)	 100 taxa (500 bp)
DNADIST+ NJ/BIONJ	0.3 sec	2.3 sec
DNADIST+ Weighbor	1.5 sec	22 sec
DNAPARS	0.5 sec	6 sec
PAUP*	3 min, 21 sec	1 hr, 4 min
PAUP*+ NJ	1 min, 10 sec	22 min
MrBayes	2 min, 6 sec	32 min, 37 sec
fastDNAml	1 min, 13 sec	26 min, 31 sec
NJML	15 sec	6 min, 4 sec
MetaPIGA	21 sec	3 min, 27 sec
MetaPIGA+ NJ	6 sec	23 sec
PHYML	2.7 sec (6.4)	12 sec (8.3)

Guindon and Gascuel (2003) Syst. Biol.

PhyML-SPR

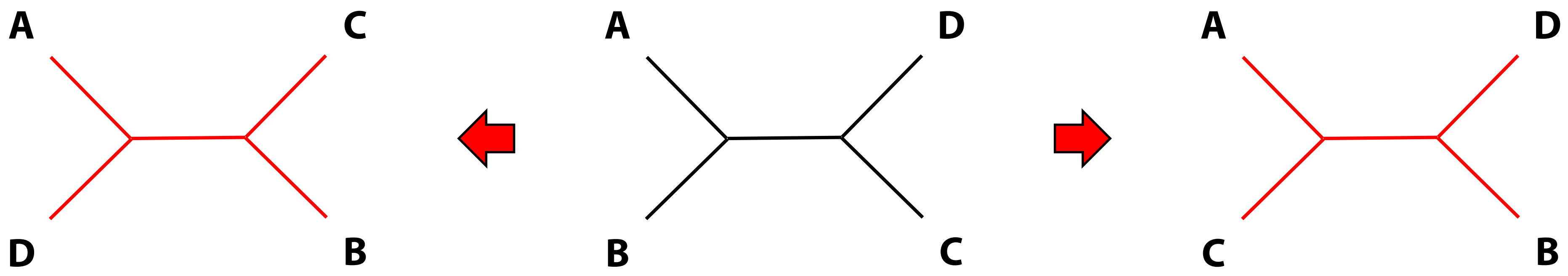
- Filter candidate SPR moves based on distance criterion
- Quick estimation of branch-length using distance-based approach

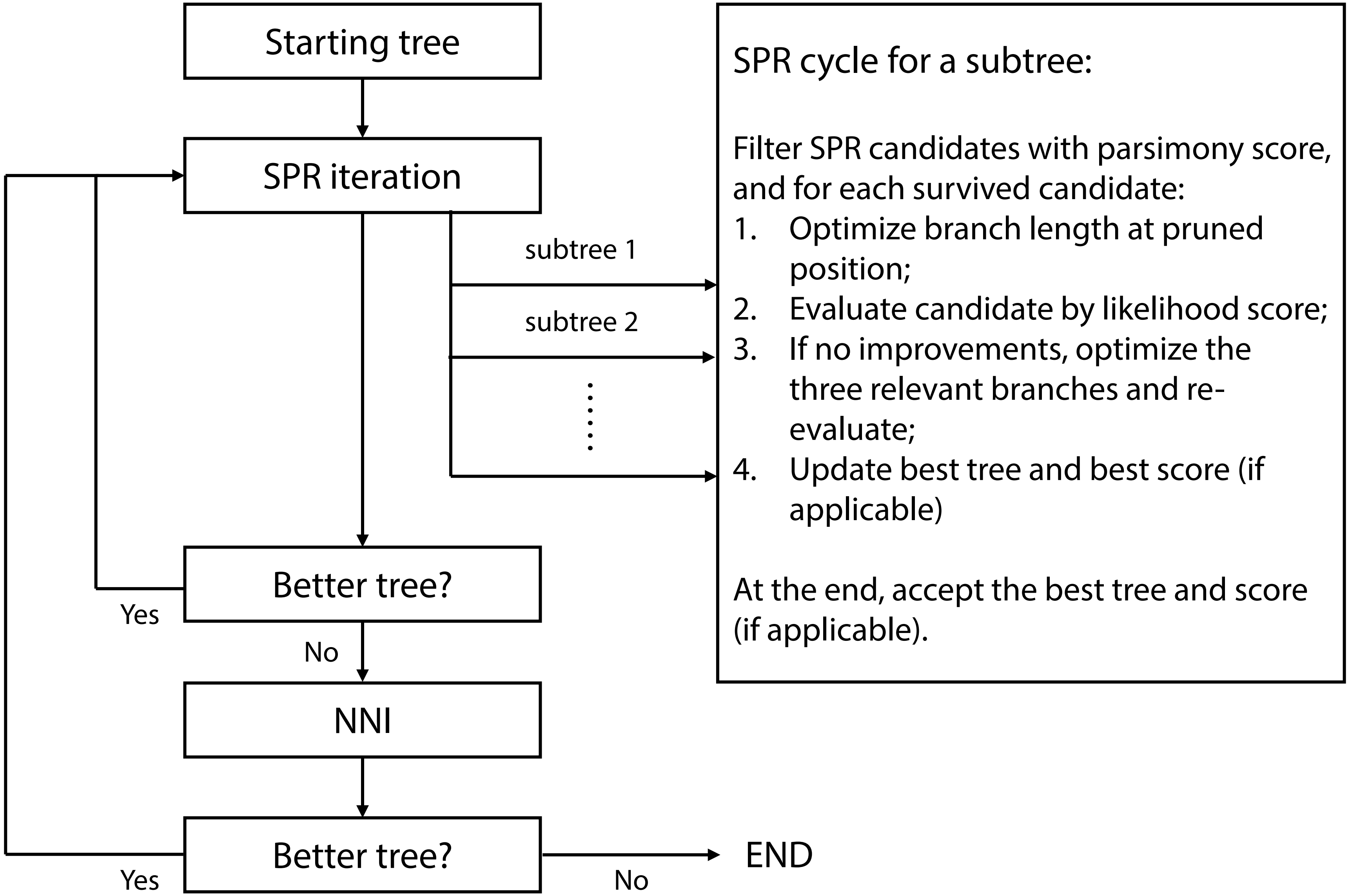


Hordijk and Gascuel (2005) Bioinformatics

PhyML 3

- Use parsimony score instead of distance to filter SPR candidates
- Alternated SPR and NNI searches
  - NNI optimizes all five relevant branches instead of one





SPR cycle for a subtree:

Filter SPR candidates with parsimony score, and for each survived candidate:

1. Optimize branch length at pruned position;
2. Evaluate candidate by likelihood score;
3. If no improvements, optimize the three relevant branches and re-evaluate;
4. Update best tree and best score (if applicable)

At the end, accept the best tree and score (if applicable).

PhyML 3: performance

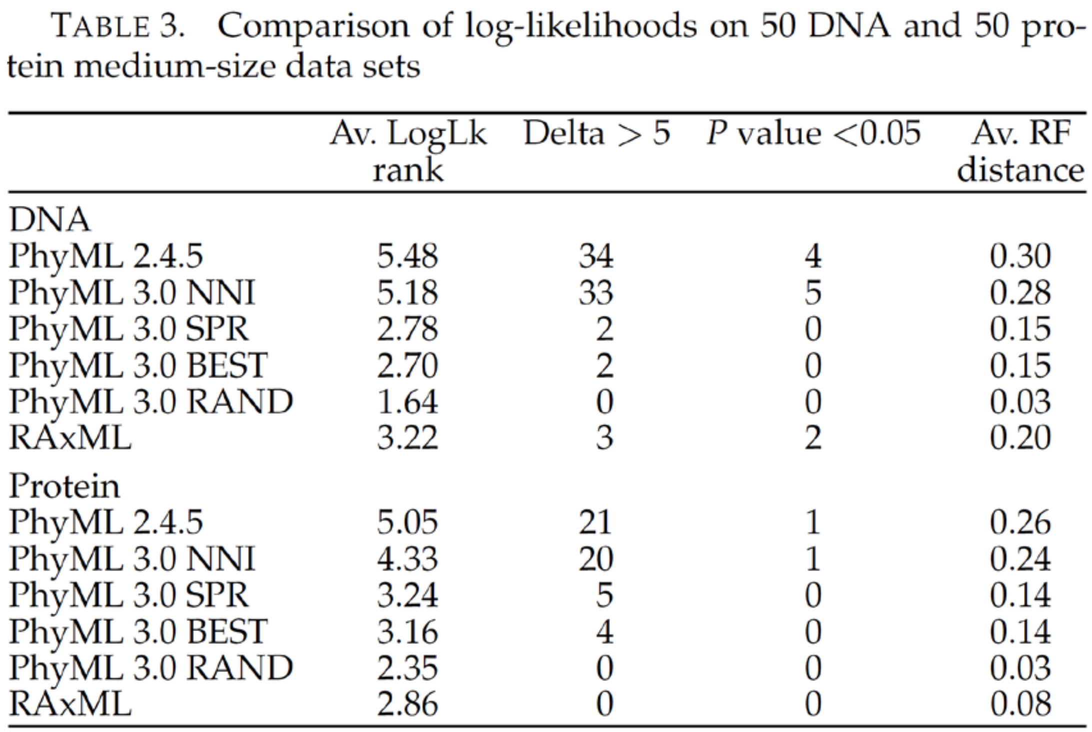


TABLE 3. Comparison of log-likelihoods on 50 DNA and 50 protein medium-size data sets

	Av. LogLk rank	Delta > 5	*P* value <0.05	Av. RF distance
DNA				
PhyML 2.4.5	5.48	34	4	0.30
PhyML 3.0 NNI	5.18	33	5	0.28
PhyML 3.0 SPR	2.78	2	0	0.15
PhyML 3.0 BEST	2.70	2	0	0.15
PhyML 3.0 RAND	1.64	0	0	0.03
RAxML	3.22	3	2	0.20
Protein				
PhyML 2.4.5	5.05	21	1	0.26
PhyML 3.0 NNI	4.33	20	1	0.24
PhyML 3.0 SPR	3.24	5	0	0.14
PhyML 3.0 BEST	3.16	4	0	0.14
PhyML 3.0 RAND	2.35	0	0	0.03
RAxML	2.86	0	0	0.08

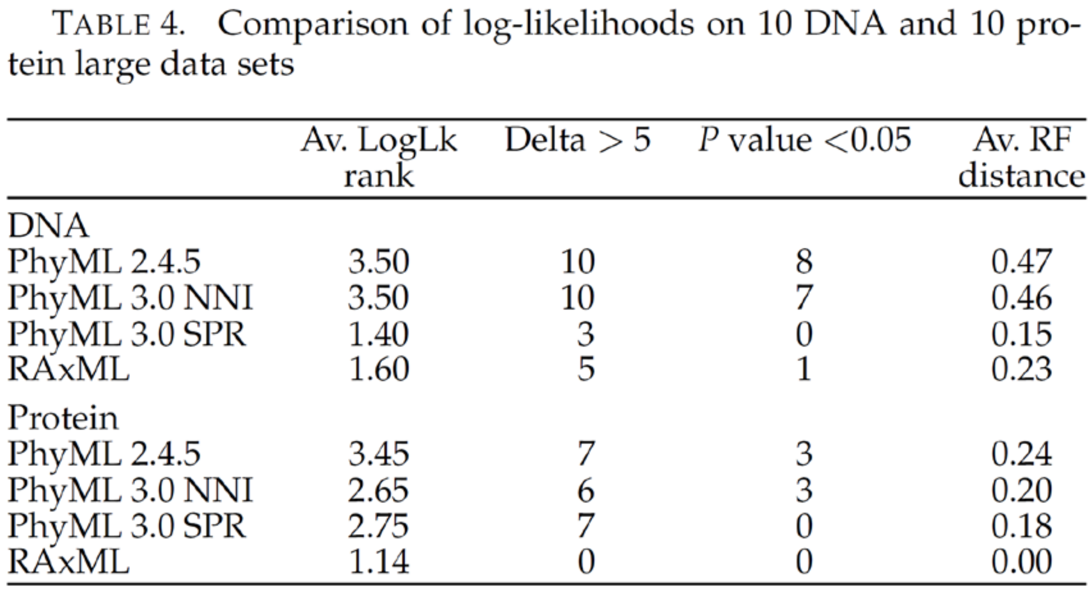
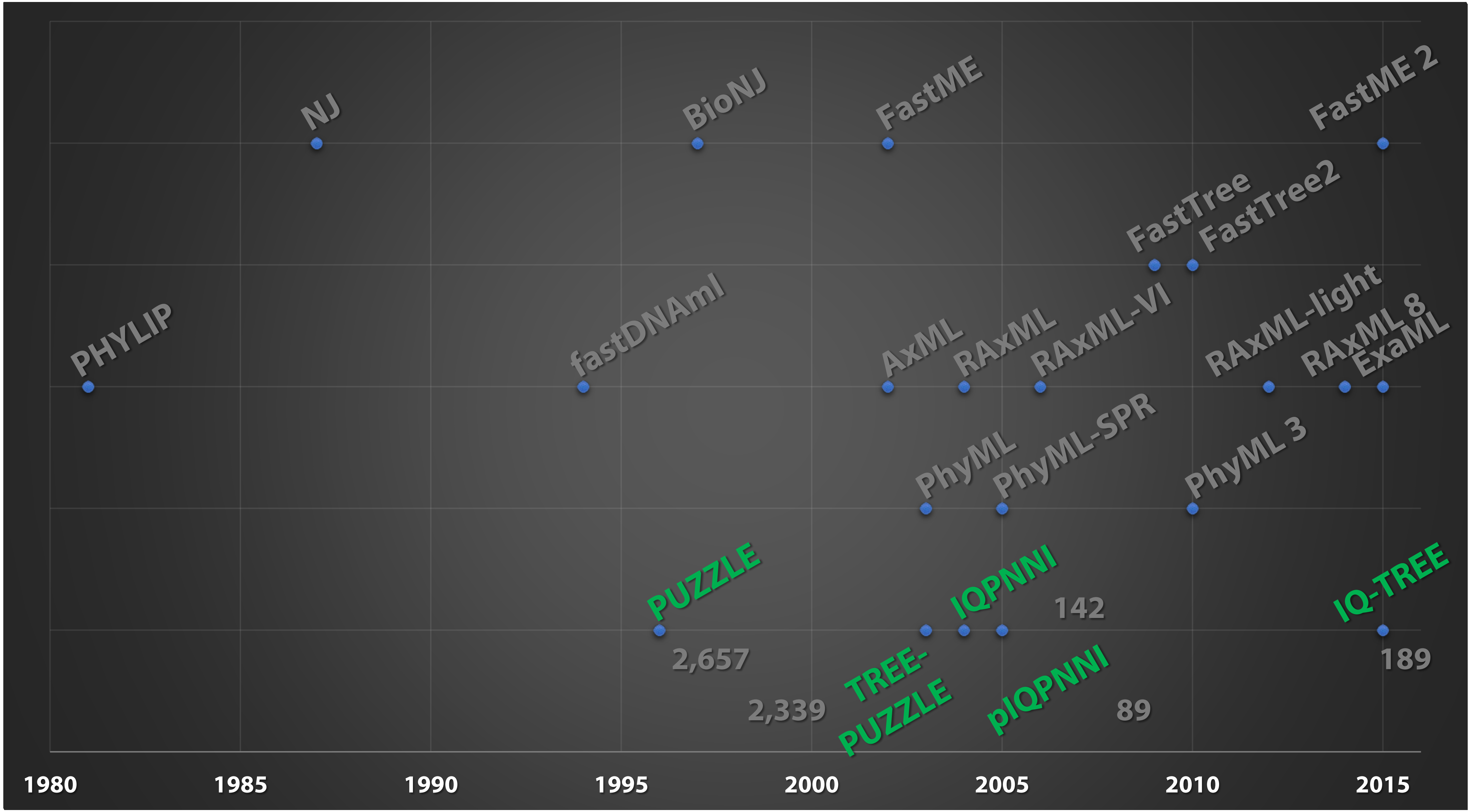


TABLE 4. Comparison of log-likelihoods on 10 DNA and 10 protein large data sets

	Av. LogLk rank	Delta > 5	*P* value <0.05	Av. RF distance
DNA				
PhyML 2.4.5	3.50	10	8	0.47
PhyML 3.0 NNI	3.50	10	7	0.46
PhyML 3.0 SPR	1.40	3	0	0.15
RAxML	1.60	5	1	0.23
Protein				
PhyML 2.4.5	3.45	7	3	0.24
PhyML 3.0 NNI	2.65	6	3	0.20
PhyML 3.0 SPR	2.75	7	0	0.18
RAxML	1.14	0	0	0.00

Guindon et al (2010) Syst. Biol.


NJ
BioNJ
FastME
FastME 2
FastTree
FastTree2
PHYLIP
fastDNAml
AxML
RAxML
RAxML-VI
RAxML-light
RAxML 8
ExaML
PhyML
PhyML-SPR
PhyML 3
PUZZLE
IQPNNI
142
IQ-TREE
2,657
189
TREE-PUZZLE
2,339
pIQPNNI
89
1980
1985
1990
1995
2000
2005
2010
2015

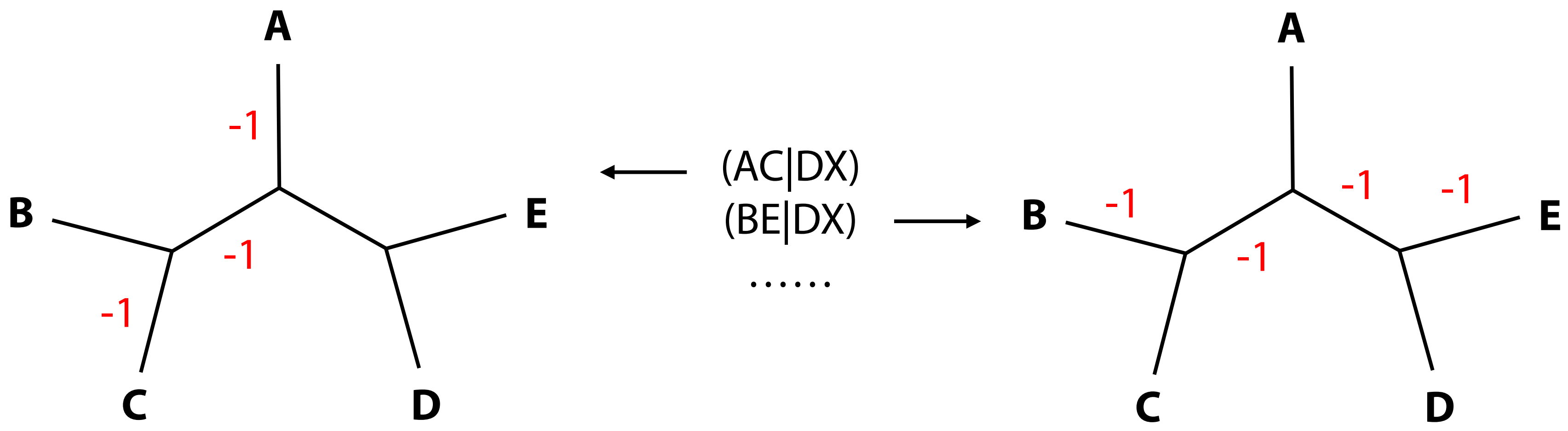
Quartet Puzzling

- Building tree from small pieces
    - 40 sequences

      $1.31x10^{51}$ binary unrooted trees

      931,390 quartets

- Stepwise addition based on quartet trees

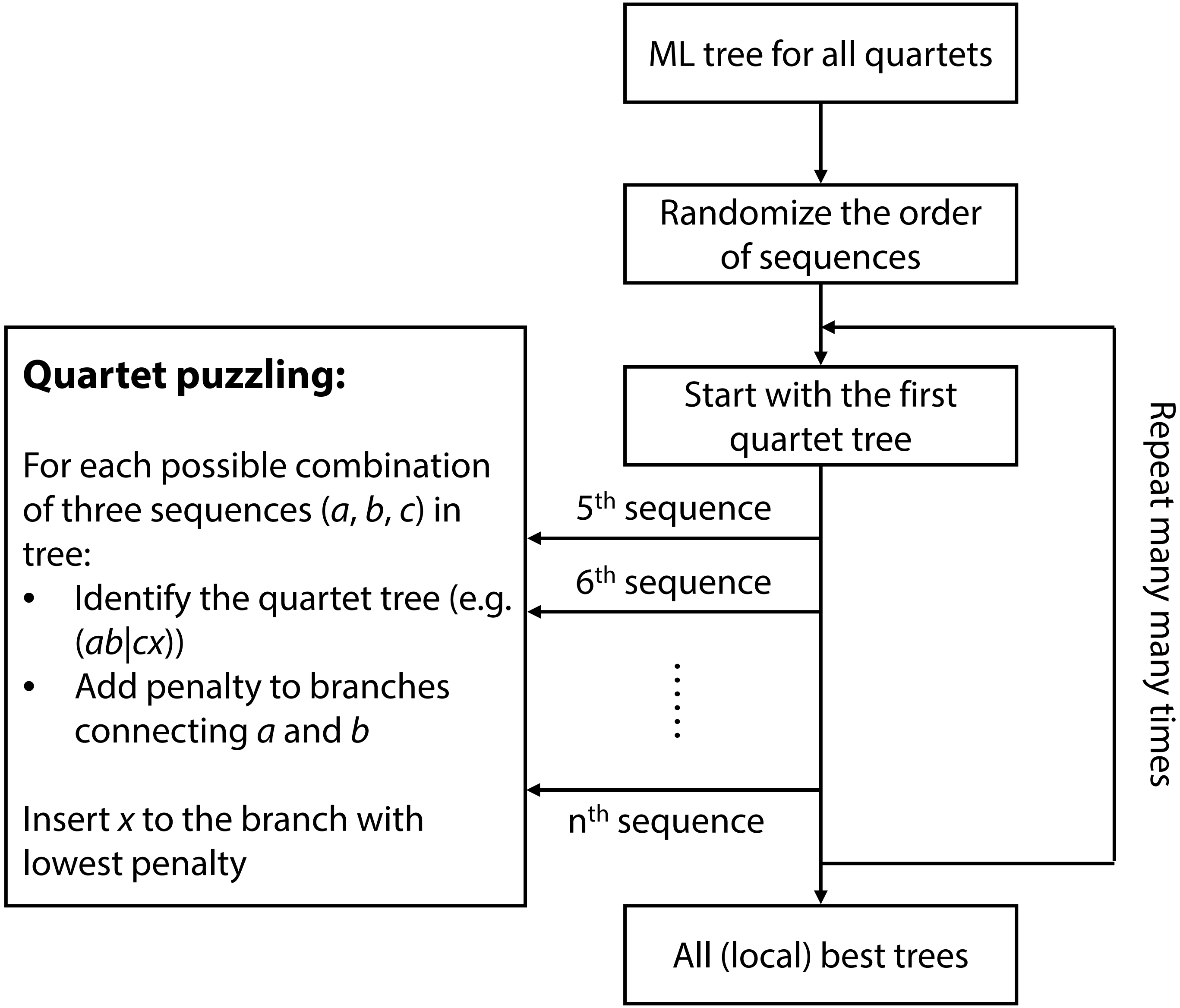


(AC|DX)

(BE|DX)

……

PUZZLE



ML tree for all quartets

Randomize the order of sequences

Start with the first quartet tree

5^{th} sequence

6^{th} sequence

……

n^{th} sequence

All (local) best trees

Repeat many many times

Quartet puzzling:

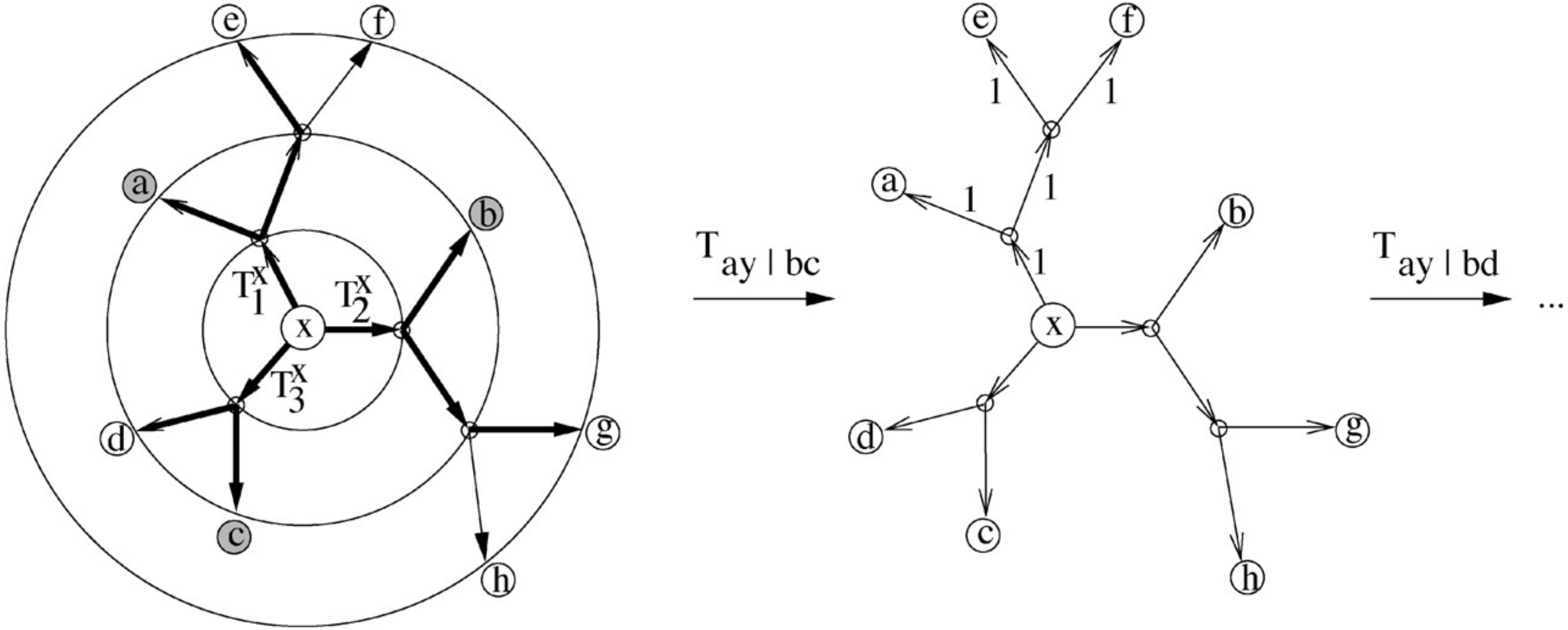
For each possible combination of three sequences (*a*, *b*, *c*) in tree:

- Identify the quartet tree (e.g. (*ab*|*cx*))
- Add penalty to branches connecting *a* and *b*

Insert *x* to the branch with lowest penalty

IQPNNI - Important quartet puzzling + NNI

- Important quartet



Vinh le and Von Haeseler (2004) Mol. Biol. Evol.

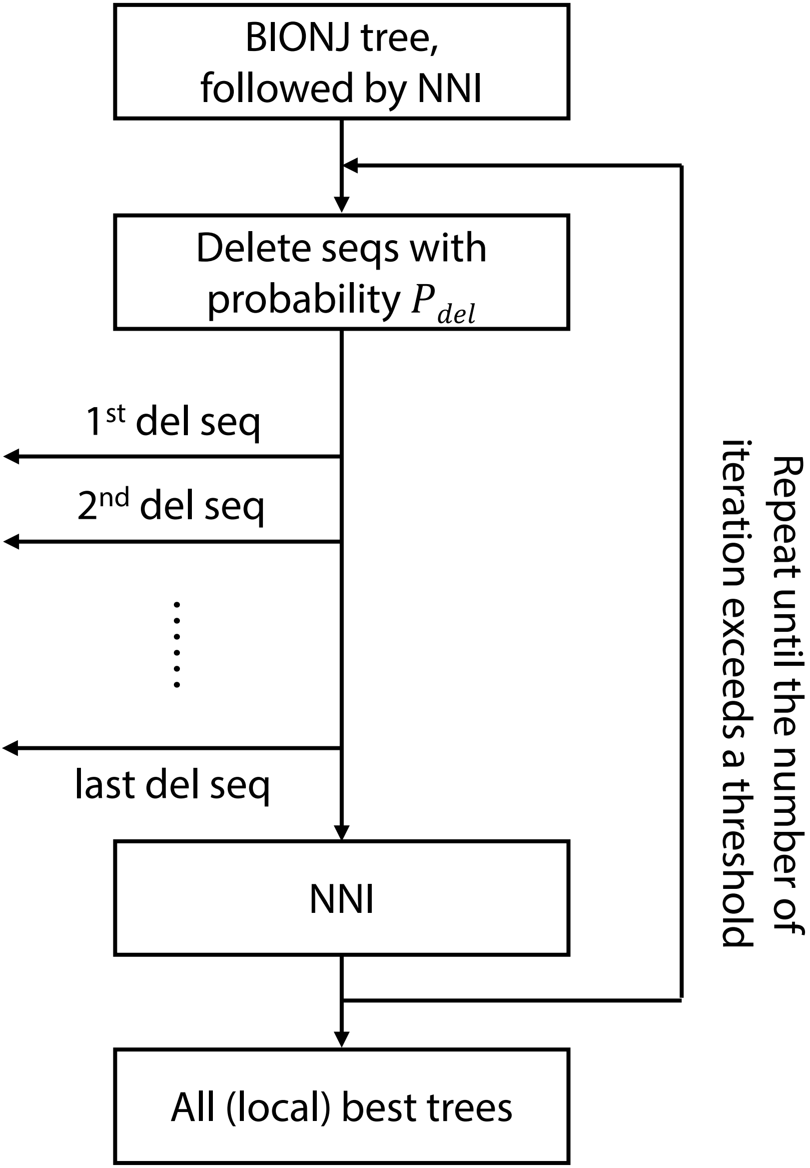
IQPNNI

Important quartet puzzling:

For each important quartet in tree:

- Build quartet tree with NJ (e.g. (*ab*|*cx*))
- Add score of 1 to all branches in the subtree represented by *c*

Insert *x* to the branch with highest score



BIONJ tree, followed by NNI

Delete seqs with probability P_{del}

1st del seq

2nd del seq

……

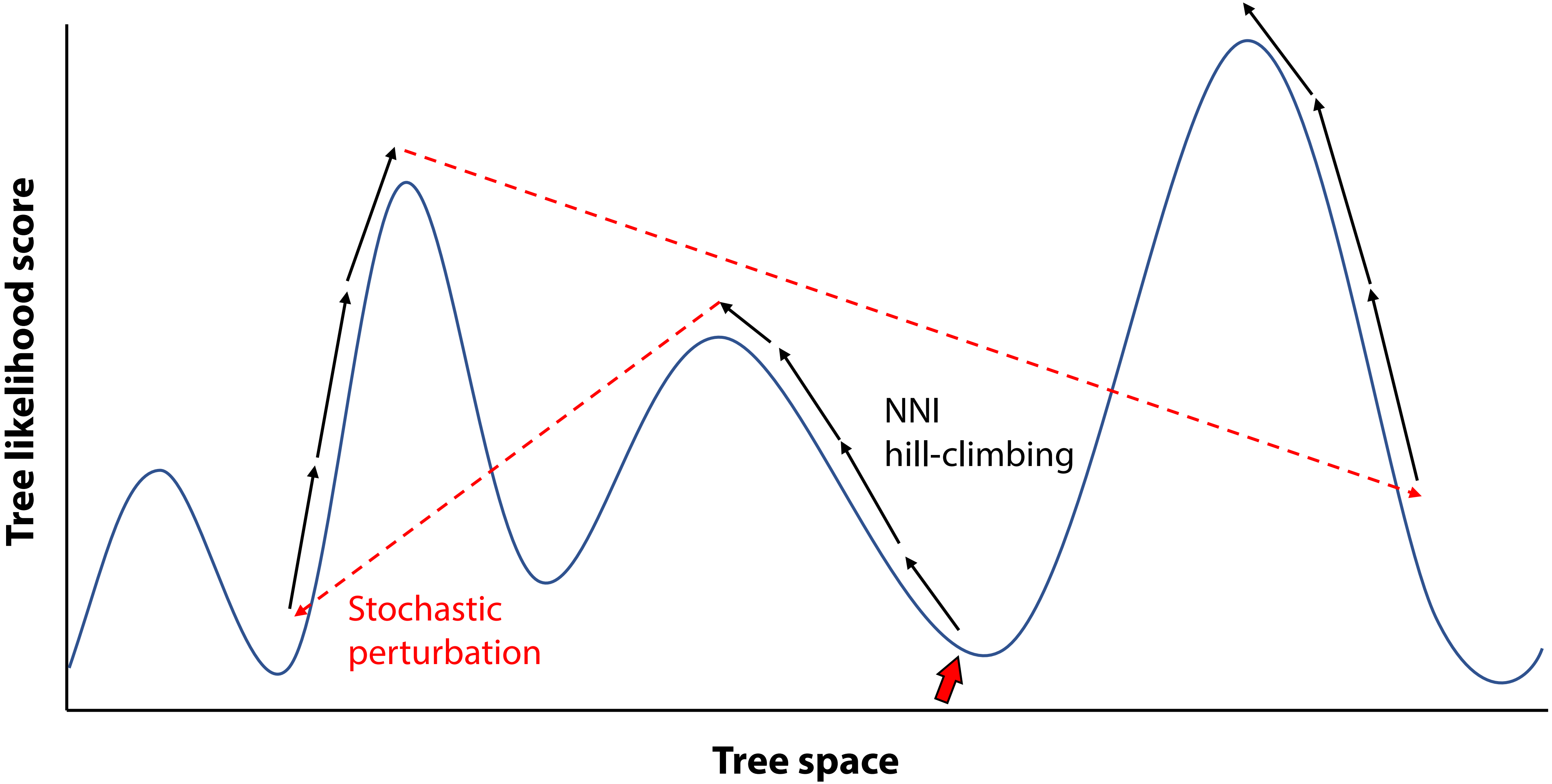
last del seq

NNI

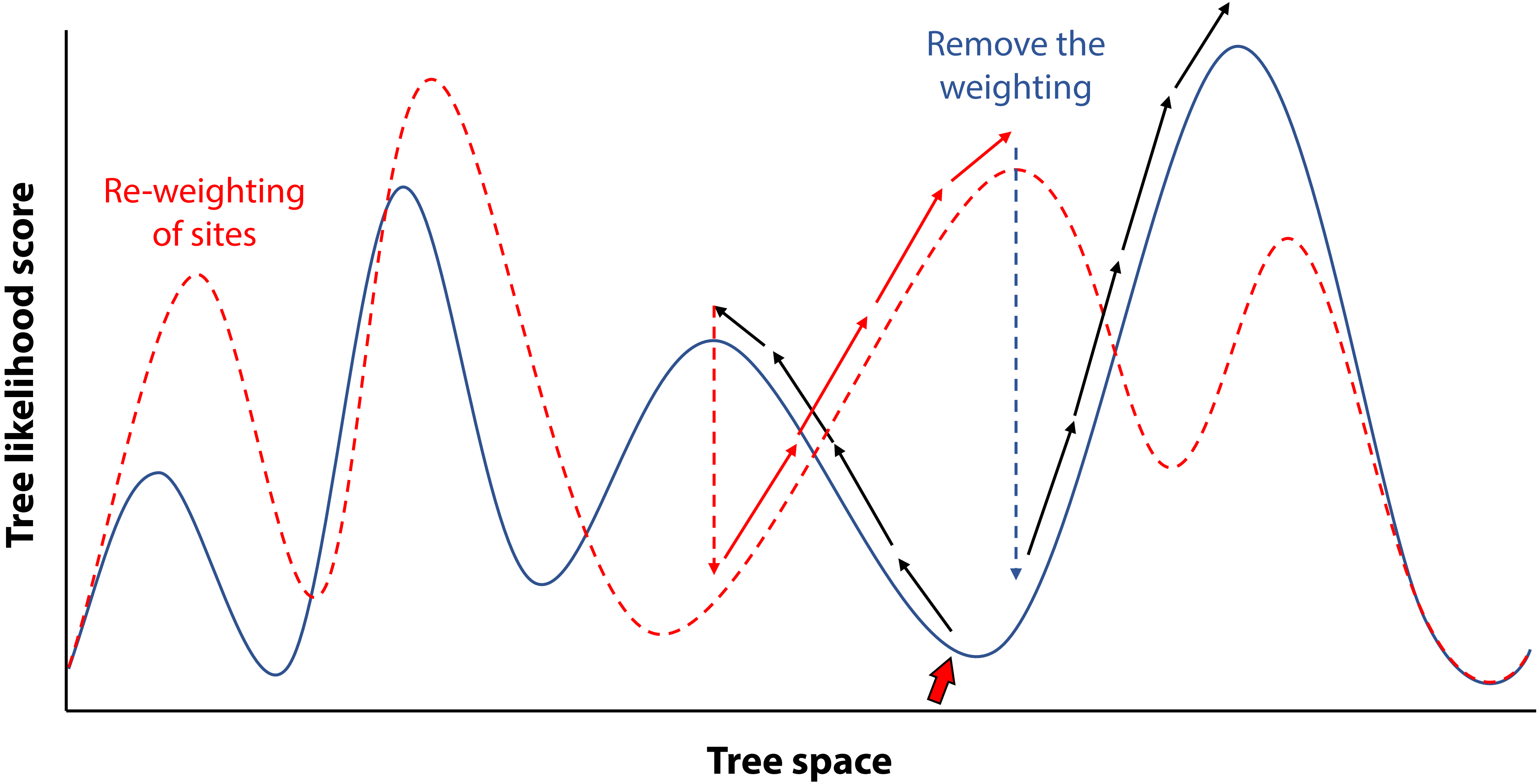
All (local) best trees

Repeat until the number of iteration exceeds a threshold

Escape from local optima

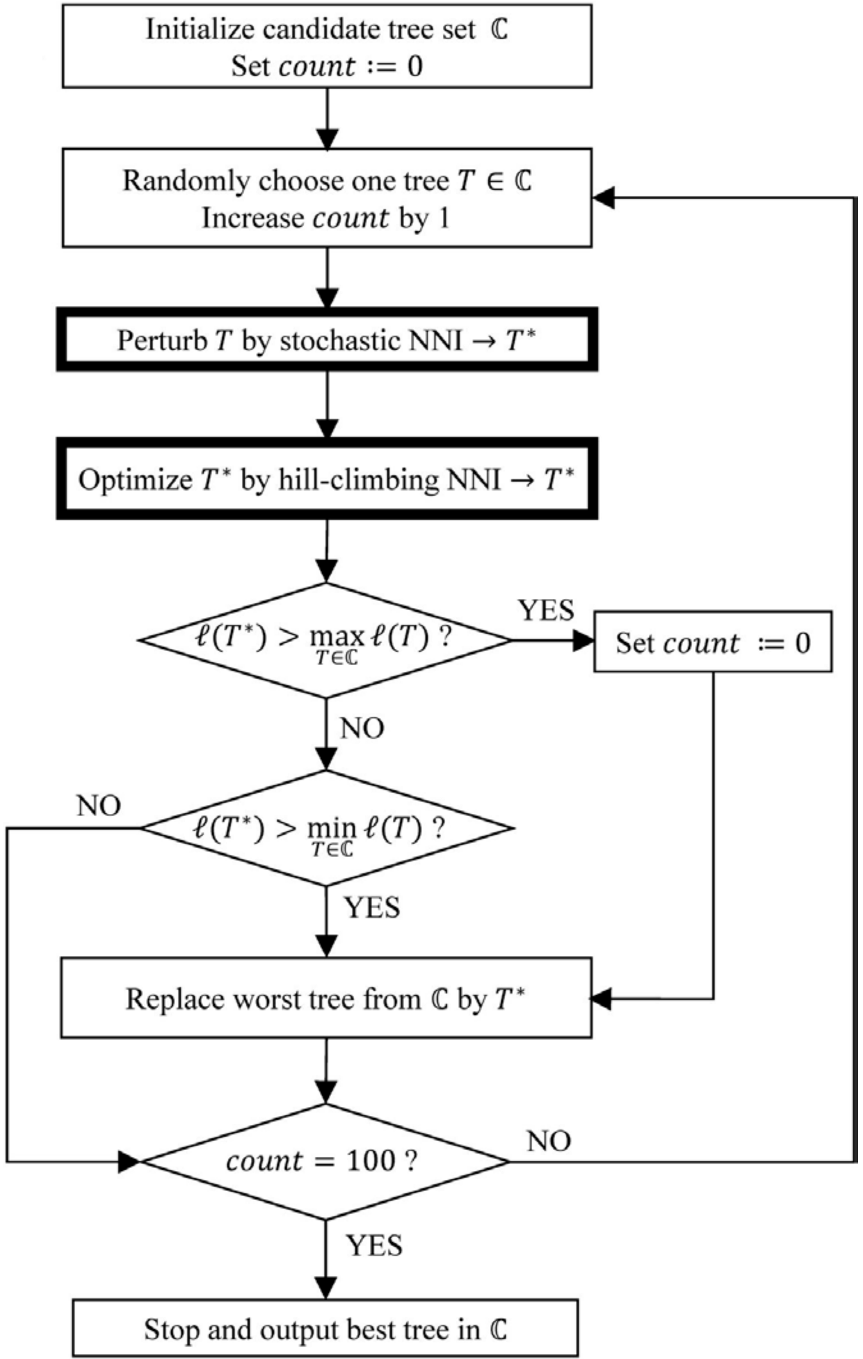


Ratchet


Re-weighting of sites
Remove the weighting
Tree likelihood score
Tree space

IQ-TREE

- A pool of starting trees
- A pool of candidate trees
- NNI- instead of IQP-based perturbation
- Simultaneous NNI modifications
  - Reduced NNI neighborhood



Initialize candidate tree set $\mathbb{C}$
Set $count := 0$

Randomly choose one tree $T \in \mathbb{C}$
Increase $count$ by 1

Perturb T by stochastic NNI $\rightarrow T^*$

Optimize T^* by hill-climbing NNI $\rightarrow T^*$

$\ell(T^*) > \max_{T \in \mathbb{C}} \ell(T)$?

YES

Set $count := 0$

NO

$\ell(T^*) > \min_{T \in \mathbb{C}} \ell(T)$?

NO

YES

Replace worst tree from $\mathbb{C}$ by T^*

$count = 100$?

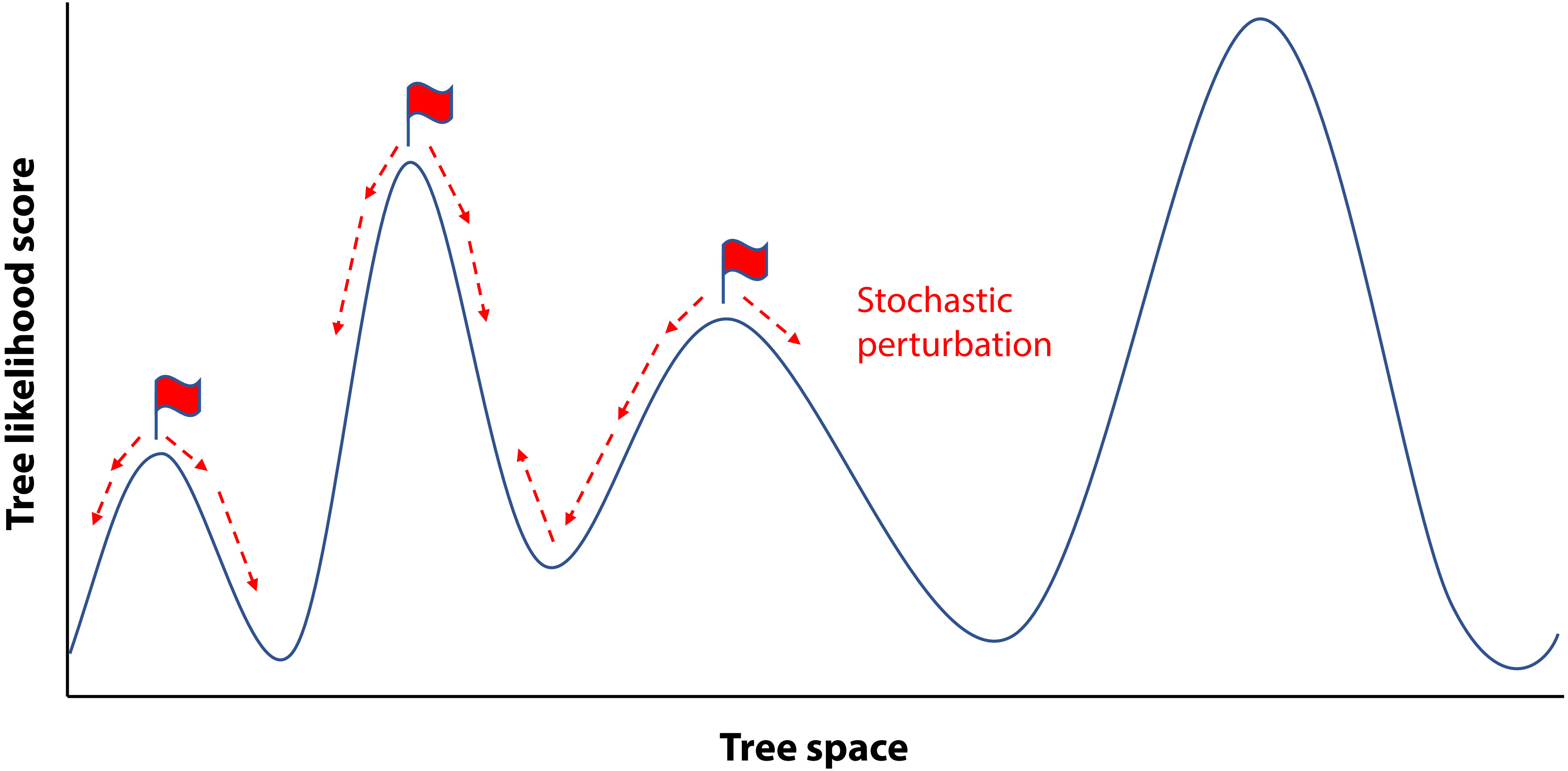
NO

YES

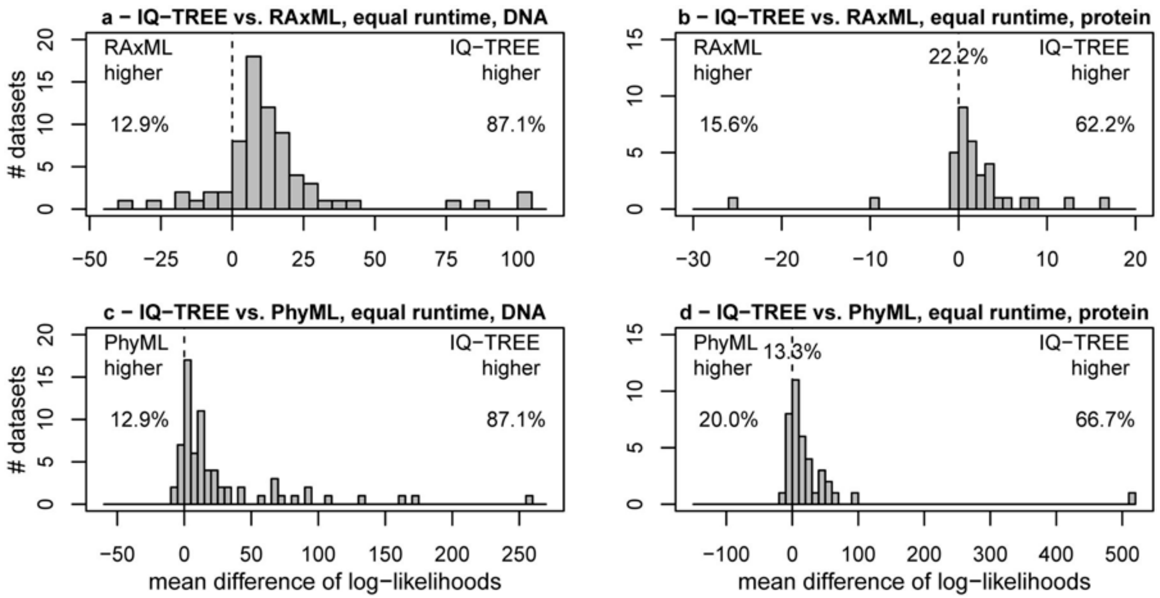
Stop and output best tree in $\mathbb{C}$

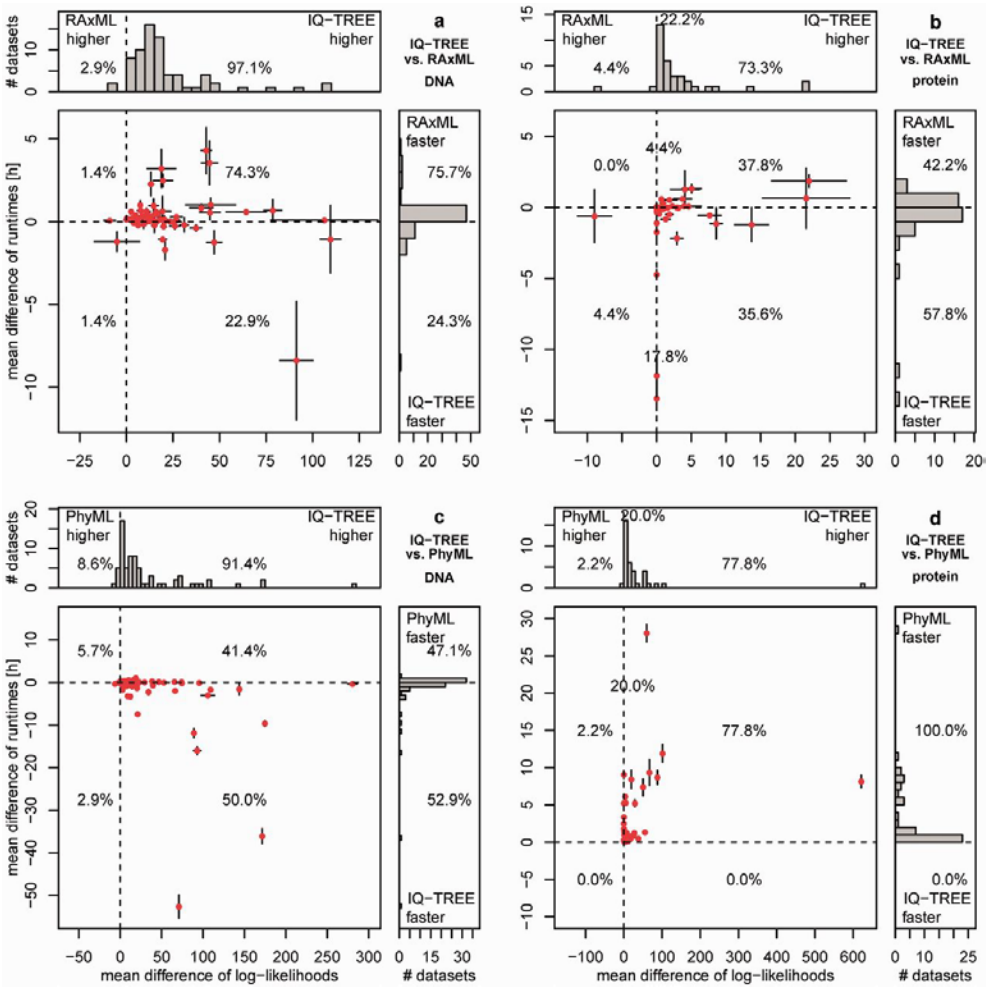
Nguyen et al (2015) Mol. Biol. Evol.

Escape from local optima: IQ-TREE



IQ-TREE: performance





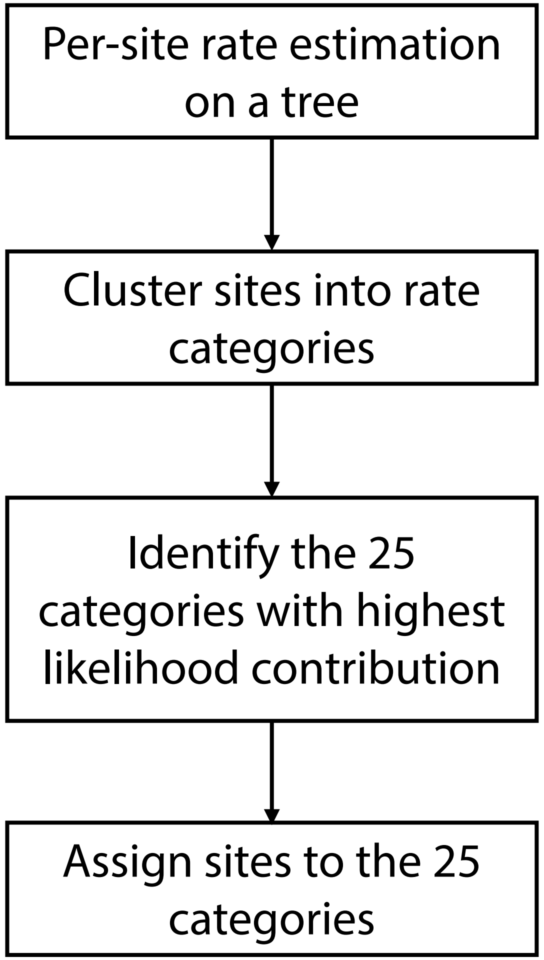
Nguyen et al (2015) Mol. Biol. Evol.

Other techniques for fast phylogenetics

- GAMMA vs CAT

GAMMA vs CAT

- GAMMA
  - model rate heterogeneity among sites using the gamma distribution
  - each site has certain probability belonging to each rate category
- CAT
  - assign sites into fixed number of rate categories
  - each site belongs to a specific rate category



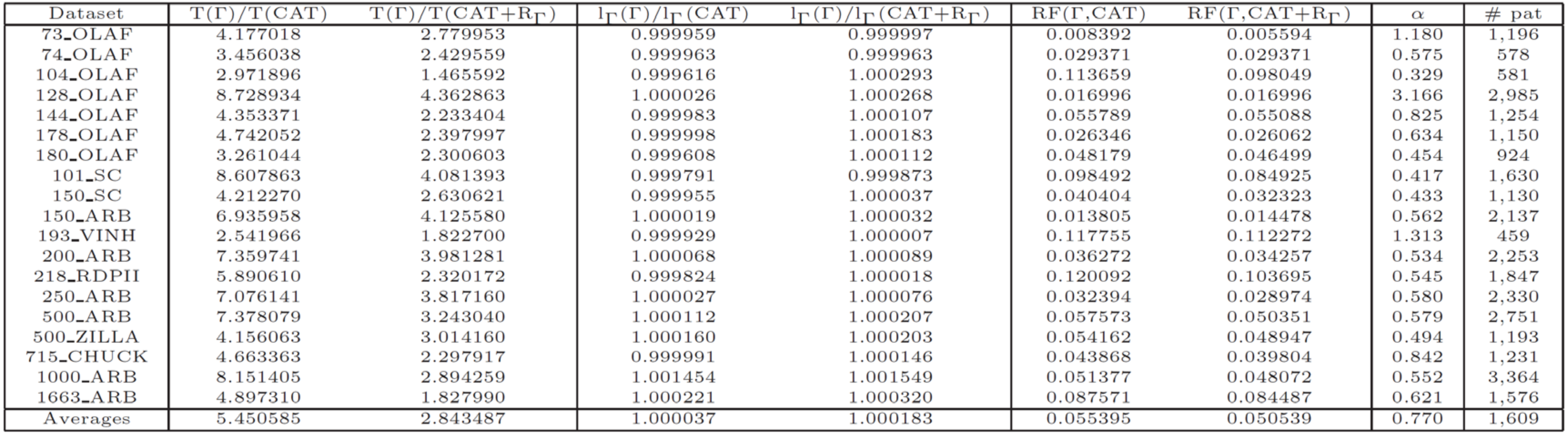
Per-site rate estimation on a tree

Cluster sites into rate categories

Identify the 25 categories with highest likelihood contribution

Assign sites to the 25 categories

GAMMA vs CAT



Dataset	$T(\Gamma)/T(CAT)$	$T(\Gamma)/T(CAT+R_\Gamma)$	$l_\Gamma(\Gamma)/l_\Gamma(CAT)$	$l_\Gamma(\Gamma)/l_\Gamma(CAT+R_\Gamma)$	$RF(\Gamma,CAT)$	$RF(\Gamma,CAT+R_\Gamma)$	α	# pat
73_OLAF	4.177018	2.779953	0.999959	0.999997	0.008392	0.005594	1.180	1,196
74_OLAF	3.456038	2.429559	0.999963	0.999963	0.029371	0.029371	0.575	578
104_OLAF	2.971896	1.465592	0.999616	1.000293	0.113659	0.098049	0.329	581
128_OLAF	8.728934	4.362863	1.000026	1.000268	0.016996	0.016996	3.166	2,985
144_OLAF	4.353371	2.233404	0.999983	1.000107	0.055789	0.055088	0.825	1,254
178_OLAF	4.742052	2.397997	0.999998	1.000183	0.026346	0.026062	0.634	1,150
180_OLAF	3.261044	2.300603	0.999608	1.000112	0.048179	0.046499	0.454	924
101_SC	8.607863	4.081393	0.999791	0.999873	0.098492	0.084925	0.417	1,630
150_SC	4.212270	2.630621	0.999955	1.000037	0.040404	0.032323	0.433	1,130
150_ARB	6.935958	4.125580	1.000019	1.000032	0.013805	0.014478	0.562	2,137
193_VINH	2.541966	1.822700	0.999929	1.000007	0.117755	0.112272	1.313	459
200_ARB	7.359741	3.981281	1.000068	1.000089	0.036272	0.034257	0.534	2,253
218_RDPII	5.890610	2.320172	0.999824	1.000018	0.120092	0.103695	0.545	1,847
250_ARB	7.076141	3.817160	1.000027	1.000076	0.032394	0.028974	0.580	2,330
500_ARB	7.378079	3.243040	1.000112	1.000207	0.057573	0.050351	0.579	2,751
500_ZILLA	4.156063	3.014160	1.000160	1.000203	0.054162	0.048947	0.494	1,193
715_CHUCK	4.663363	2.297917	0.999991	1.000146	0.043868	0.039804	0.842	1,231
1000_ARB	8.151405	2.894259	1.001454	1.001549	0.051377	0.048072	0.552	3,364
1663_ARB	4.897310	1.827990	1.000221	1.000320	0.087571	0.084487	0.621	1,576
Averages	5.450585	2.843487	1.000037	1.000183	0.055395	0.050539	0.770	1,609

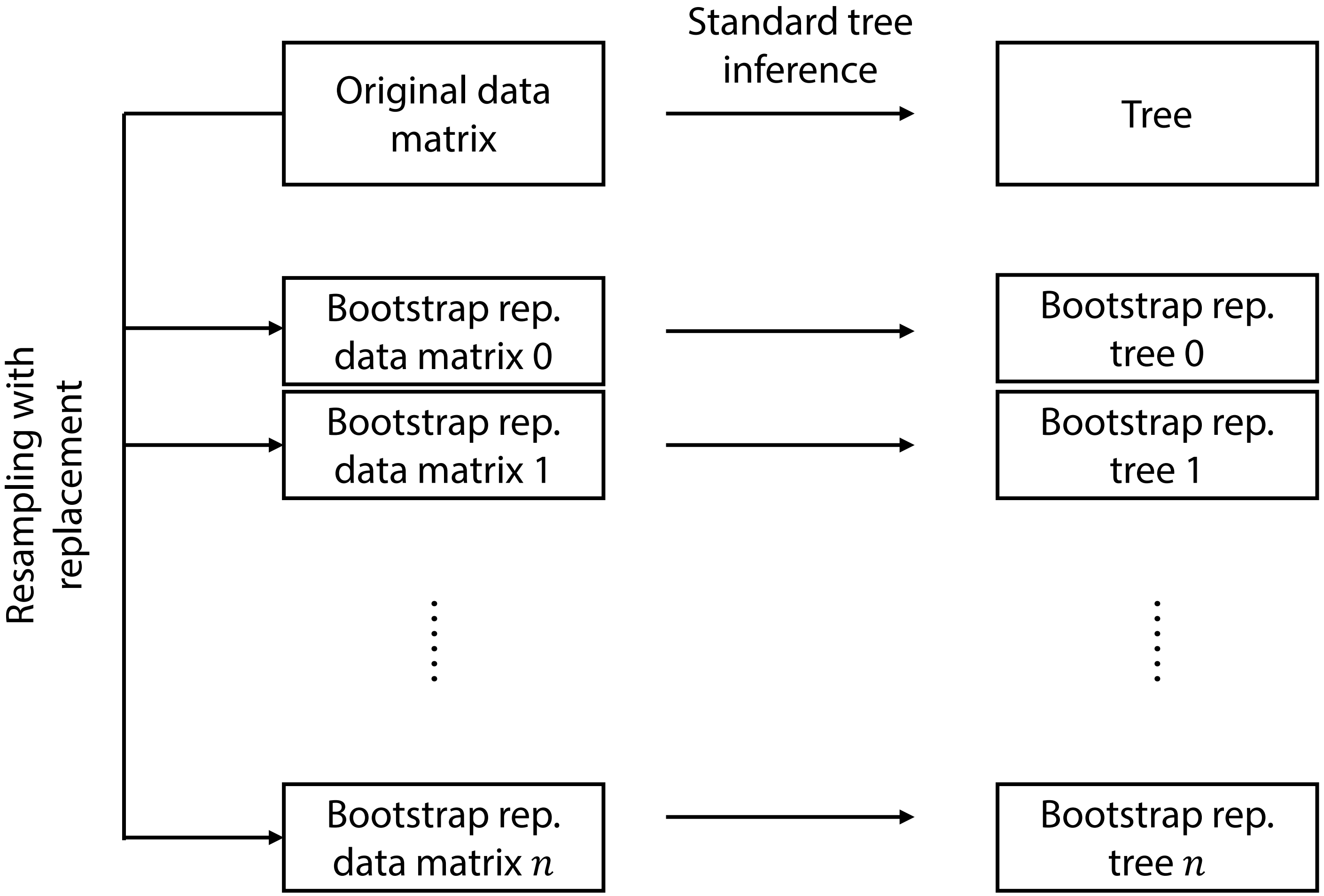
Stamatakis (2006) Proceedings 20th IEEE International Parallel & Distributed Processing Symposium

Other techniques for fast phylogenetics

- GAMMA vs CAT

- Fast approaches for node support

Standard Bootstrap



Standard tree inference

Original data matrix

Tree

Resampling with replacement

Bootstrap rep. data matrix 0

Bootstrap rep. tree 0

Bootstrap rep. data matrix 1

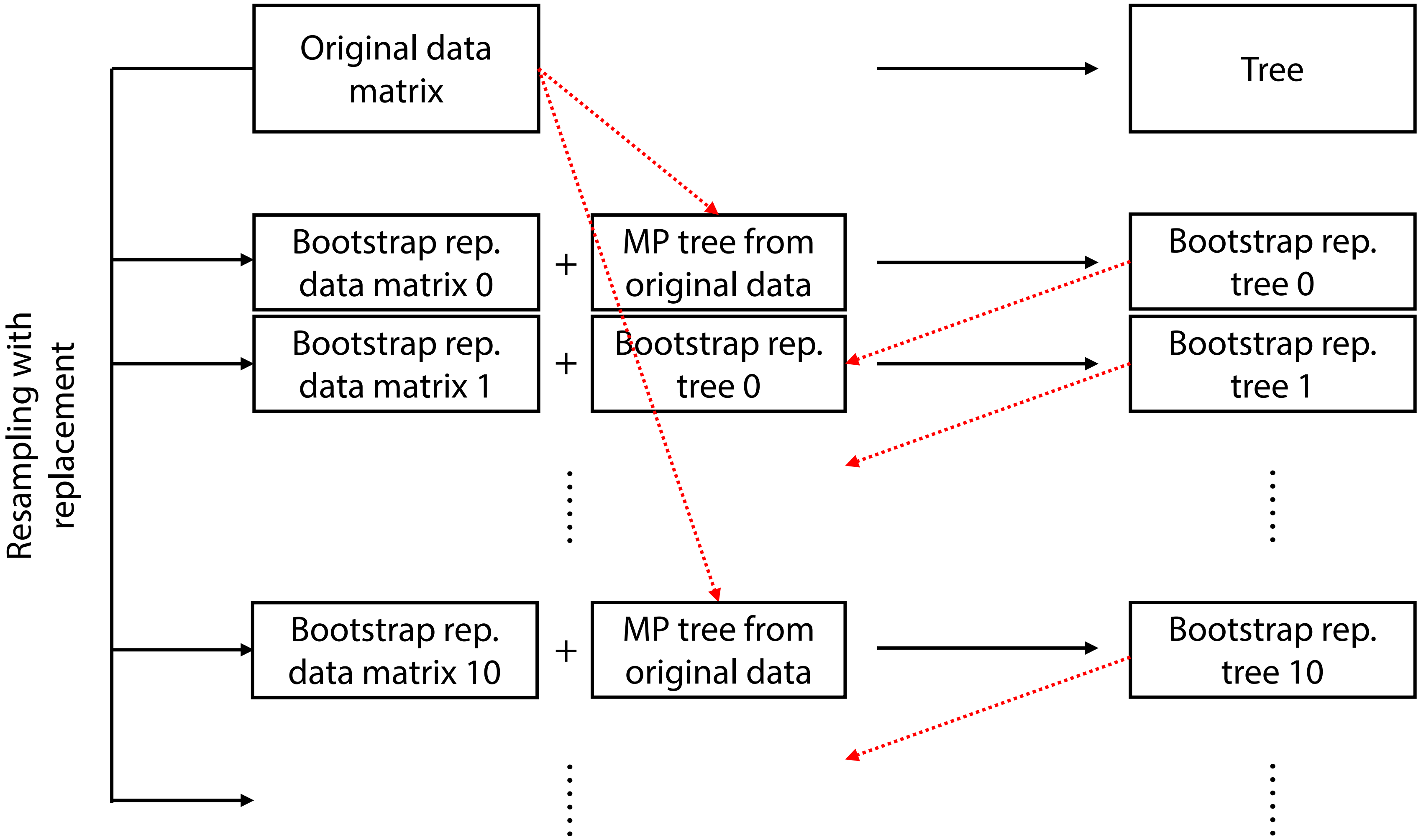
Bootstrap rep. tree 1

……

Bootstrap rep. data matrix n

Bootstrap rep. tree n

Rapid bootstrap (RAxML)



Resampling with replacement

Original data matrix

Tree

Bootstrap rep. data matrix 0 + MP tree from original data

Bootstrap rep. tree 0

Bootstrap rep. data matrix 1 + Bootstrap rep. tree 0

Bootstrap rep. tree 1

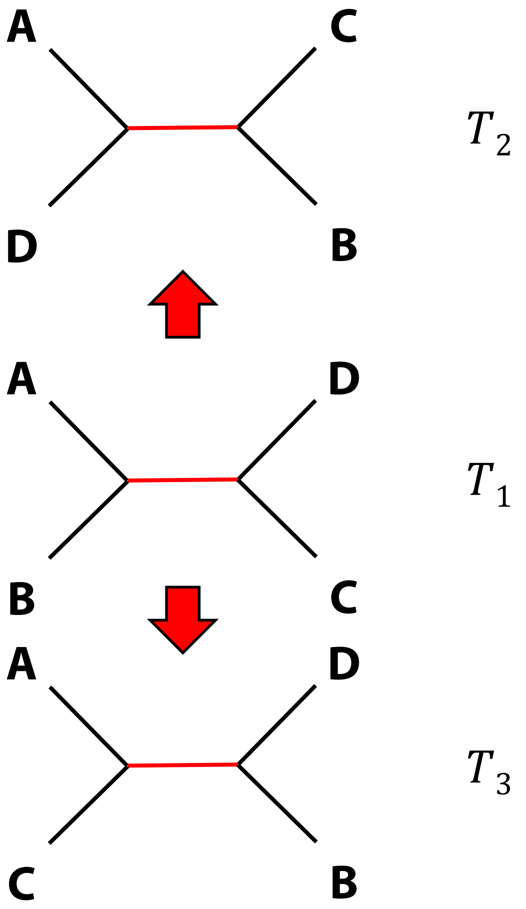
Bootstrap rep. data matrix 10 + MP tree from original data

Bootstrap rep. tree 10

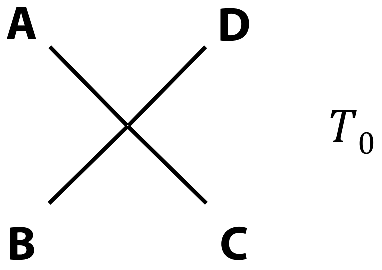
Additional shortcuts:

- LSR radius randomly chosen between 5 and 15;
- 2 iterations of LSR;
- More aggressive subtree skipping;
- Thorough optimization for best 5 instead of 20;

Local branch support

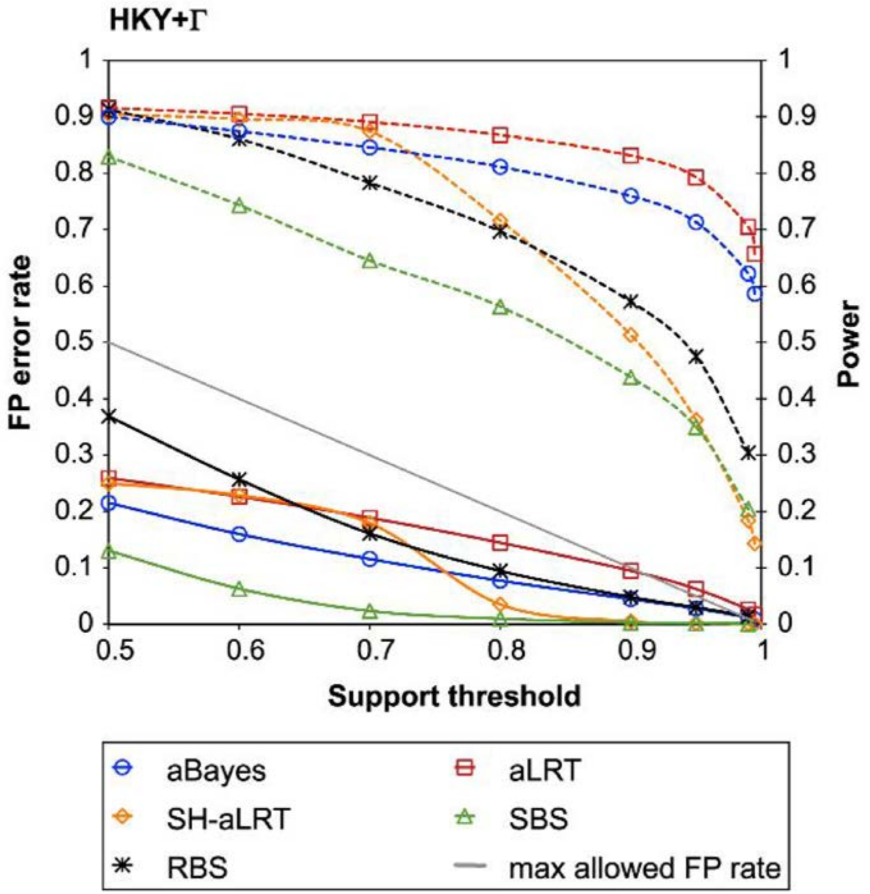


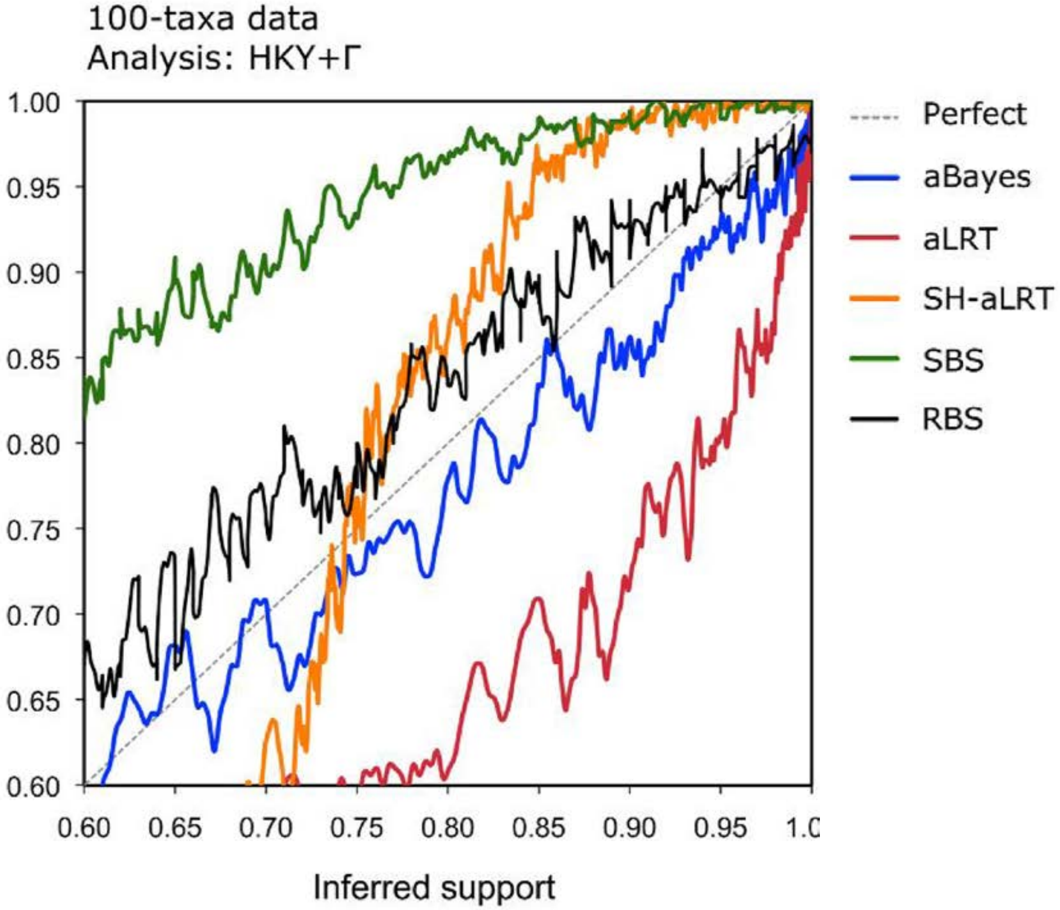
1. Approximate likelihood-ratio test (aLRT):
    - $aLRT \leftarrow 2(l_1 - \max(l_2, l_3)) < 2(l_1 - l_0)$



2. SH-aLRT:
    - aLRT with RELL bootstrap re-sampling
    - Support value = $count(aLRT > aLRT^*)$
3. Approximate Bayes (aBayes):
    - $\Pr(T_1|D) = \Pr(D|T_1)\Pr(T_1)/\sum_{i=1}^{3}\Pr(D|T_i)\Pr(Ti)$

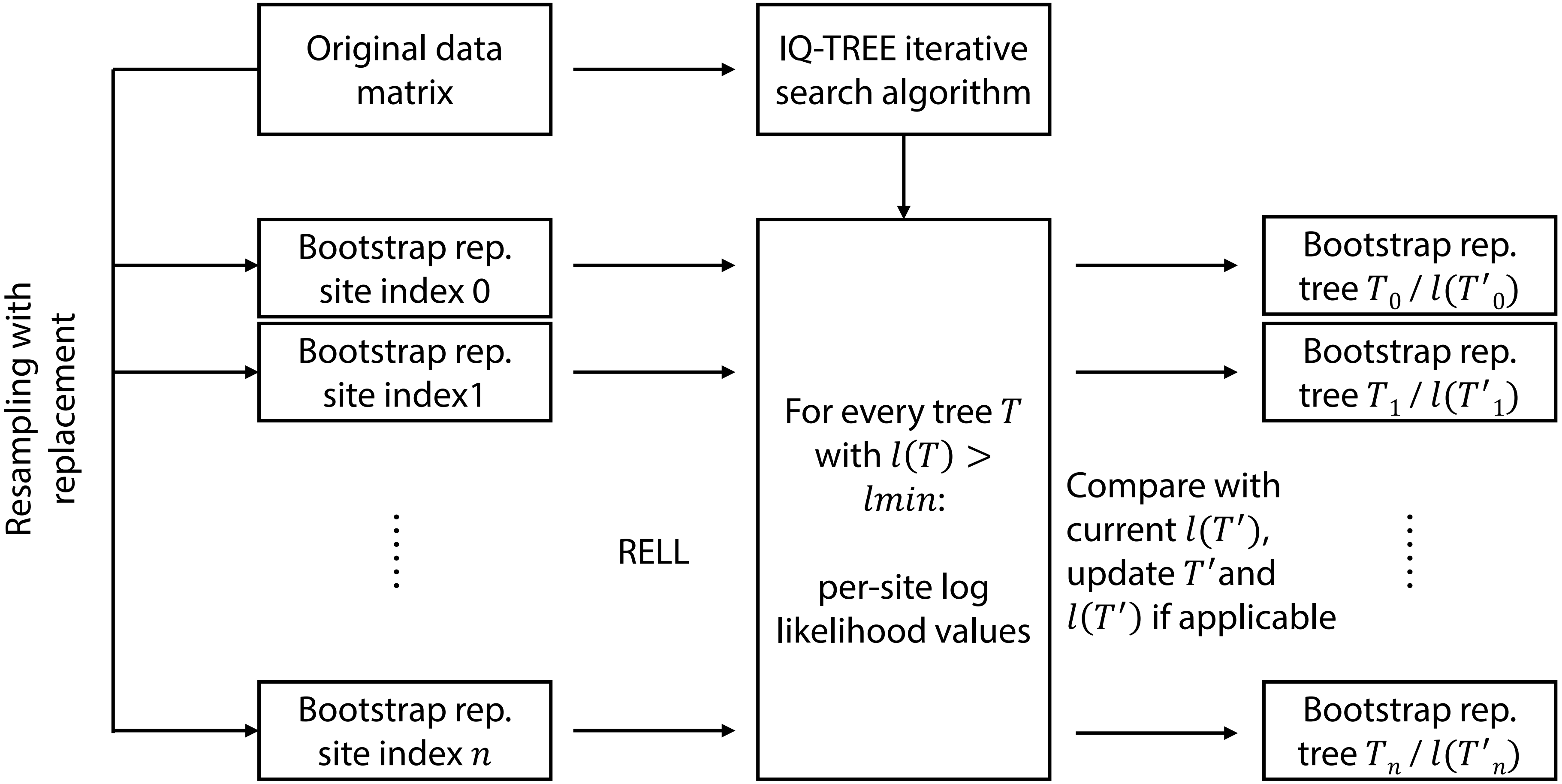
Local branch support: performance





Anisimova et al (2011) Syst. Biol.

Ultra-fast bootstrap (IQ-TREE)



Resampling with replacement

Original data matrix

IQ-TREE iterative search algorithm

Bootstrap rep. site index 0

Bootstrap rep. site index1

……

Bootstrap rep. site index n

RELL

For every tree T with $l(T) > lmin$:

per-site log likelihood values

Compare with current $l(T')$, update T' and $l(T')$ if applicable

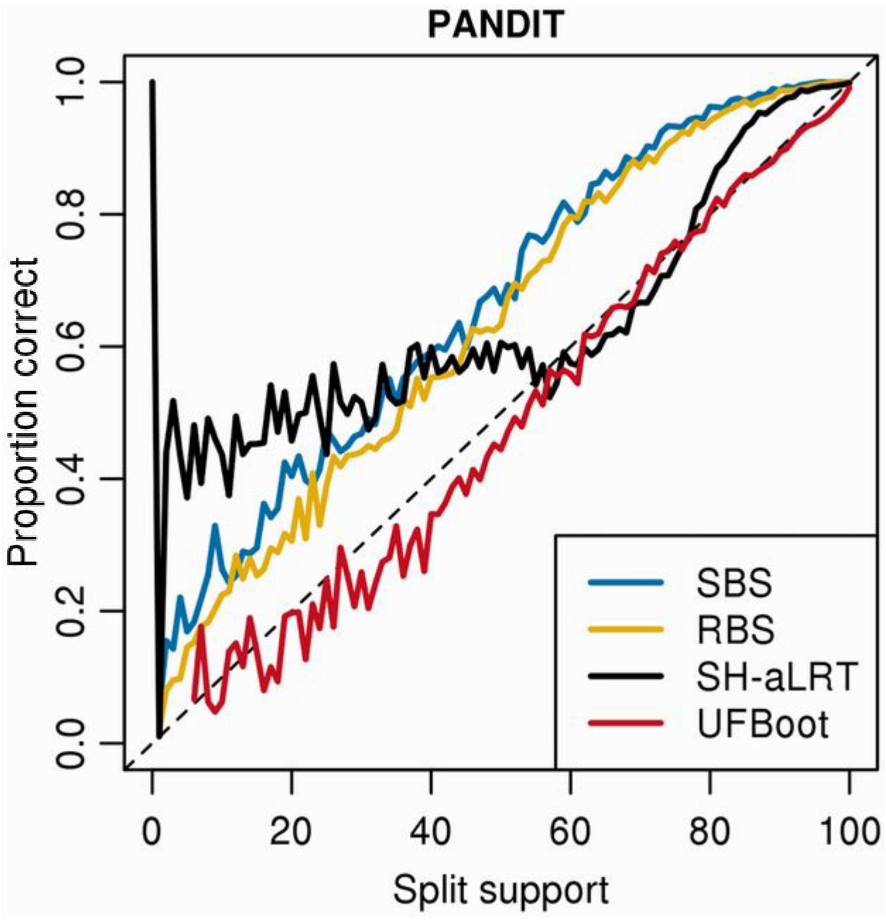
Bootstrap rep. tree T_0 / $l(T'_0)$

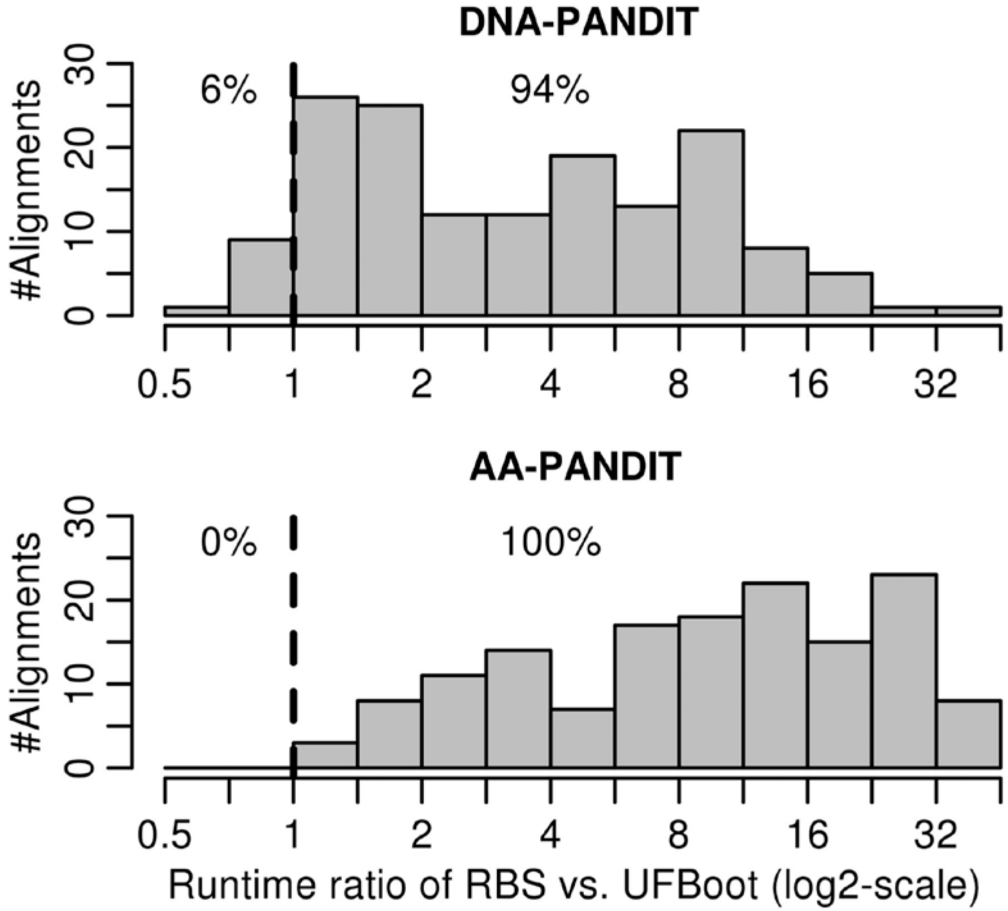
Bootstrap rep. tree T_1 / $l(T'_1)$

……

Bootstrap rep. tree T_n / $l(T'_n)$

UFBS: performance





Minh et al (2013) Mol. Biol. Evol.

Other techniques for fast phylogenetics

- GAMMA vs CAT
- Fast approaches for node support
- Parallelization

Parallelization

- Multi-threading and MPI
- Parallel tree searches:
  - MPI: RAxML, PhyML, TREE-PUZZLE/pIQPNNI/IQ-TREE
- Likelihood calculation
  - RAxML/IQ-TREE

Other techniques for fast phylogenetics

- GAMMA vs CAT

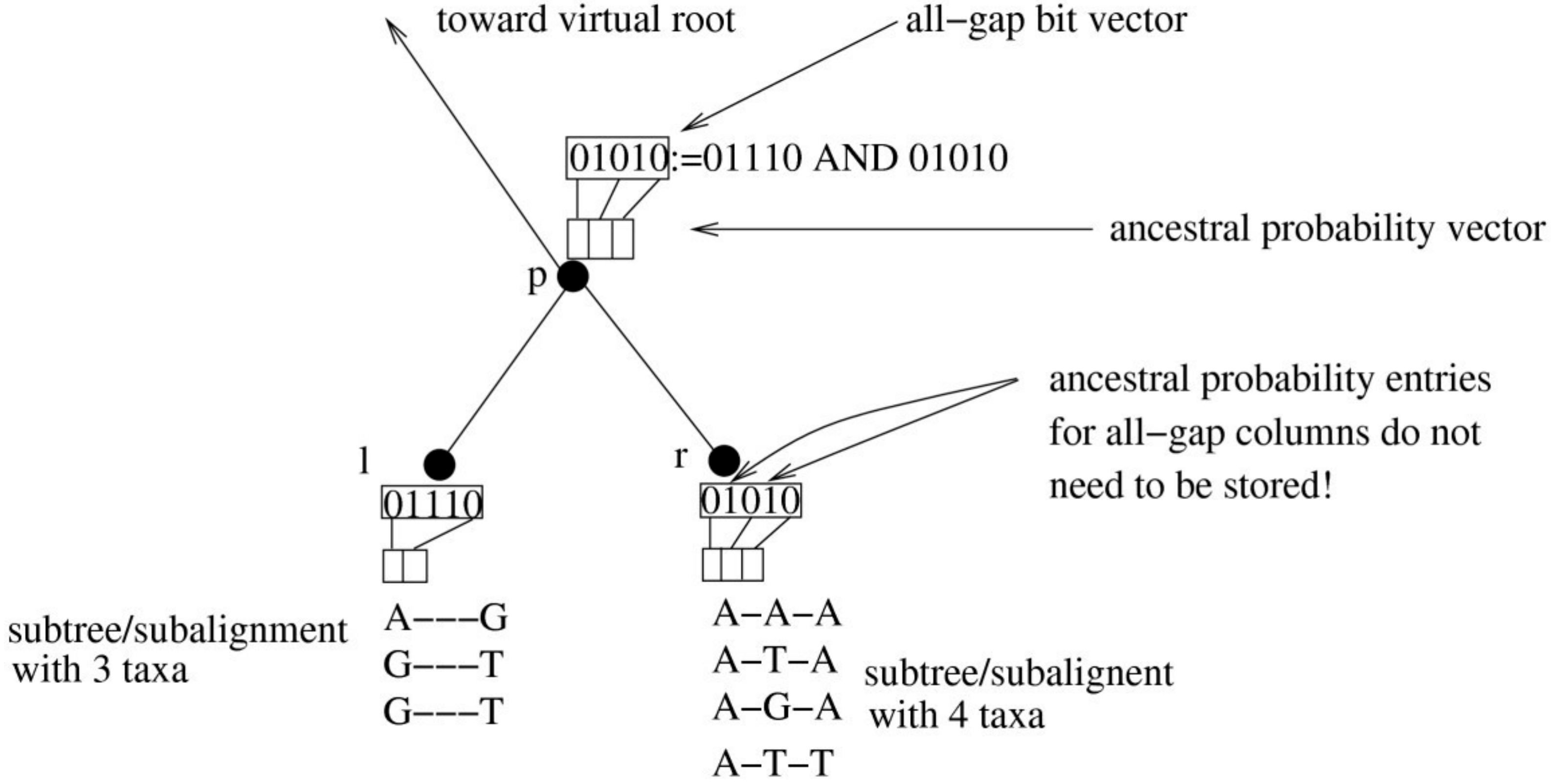
- Fast approaches for node support

- Parallelization

- Memory saving

Memory saving

- Subtree equality vector (SEV) for gappy data set



Izquierdo-Carrasco et al (2011) BMC Bioinformatics

Recommendations:

- Multiple searches using distinct starting trees

- More thorough searches by tuning key parameters
  - FastTree: more thorough NNI (“-mlacc”, “-slownni”) / no. of ME SPR (“-spr”)
  - RAxML: Lazy SPR radius (“-c”) / old hill-climbing strategy (“-f o”)
  - IQ-TREE: length of search (“-nstop”) / perturbation strength (“-pers”)

- More than one phylogenetic approaches

NJ

Saitou, N. and M. Nei (1987). "The neighbor-joining method: a new method for reconstructing phylogenetic trees." Mol Biol Evol **4**(4): 406-425.

Gascuel, O. and M. Steel (2006). "Neighbor-joining revealed." Mol Biol Evol **23**(11): 1997-2000.

Mihaescu, R., et al. (2009). "Why Neighbor-Joining Works." Algorithmica **54**(1): 1-24.

BIONJ

Gascuel, O. (1997). "BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data." Mol Biol Evol **14**(7): 685-695.

FastME

Desper, R. and O. Gascuel (2002). "Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle." J Comput Biol **9**(5): 687-705.

Lefort, V., et al. (2015). "FastME 2.0: A Comprehensive, Accurate, and Fast Distance-Based Phylogeny Inference Program." Mol Biol Evol **32**(10): 2798-2800.

FastTree

Price, M. N., et al. (2009). "FastTree: computing large minimum evolution trees with profiles instead of a distance matrix." Mol Biol Evol **26**(7): 1641-1650.

Price, M. N., et al. (2010). "FastTree 2--approximately maximum-likelihood trees for large alignments." PLoS One **5**(3): e9490.

RAxML/ExaML

Stamatakis, A. P., et al. (2002). "AxML: a fast program for sequential and parallel phylogenetic tree calculations based on the maximum likelihood method." Proc IEEE Comput Soc Bioinform Conf **1**: 21-28.

Stamatakis, A., et al. (2005). "RAxML-III: a fast program for maximum likelihood-based inference of large phylogenetic trees." Bioinformatics **21**(4): 456-463.

Stamatakis, A. (2006). "RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models." Bioinformatics **22**(21): 2688-2690.

Stamatakis, A. (2014). "RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies." Bioinformatics **30**(9): 1312-1313.

Stamatakis, A., et al. (2012). "RAxML-Light: a tool for computing terabyte phylogenies." Bioinformatics **28**(15): 2064-2066.

Kozlov, A. M., et al. (2015). "ExaML version 3: a tool for phylogenomic analyses on supercomputers." Bioinformatics **31**(15): 2577-2579.

PhyML

Guindon, S. and O. Gascuel (2003). "A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood." Syst Biol **52**(5): 696-704.

Hordijk, W. and O. Gascuel (2005). "Improving the efficiency of SPR moves in phylogenetic tree search methods based on maximum likelihood." Bioinformatics **21**(24): 4338-4347.

Guindon, S., et al. (2010). "New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0." Syst Biol **59**(3): 307-321.

TREE-PUZZLE/IQPNNI/IQ-TREE

Schmidt, H. A., et al. (2002). "TREE-PUZZLE: maximum likelihood phylogenetic analysis using quartets and parallel computing." Bioinformatics **18**(3): 502-504.

Vinh le, S. and A. Von Haeseler (2004). "IQPNNI: moving fast through tree space and stopping in time." Mol Biol Evol **21**(8): 1565-1571.

Minh, B. Q., et al. (2005). "pIQPNNI: parallel reconstruction of large maximum likelihood phylogenies." Bioinformatics **21**(19): 3794-3796.

Nguyen, L. T., et al. (2015). "IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies." Mol Biol Evol **32**(1): 268-274.

CAT/FreeRates model

Stamatakis, A. (2006). Phylogenetic models of rate heterogeneity: a high performance computing perspective. Proceedings 20th IEEE International Parallel & Distributed Processing Symposium.

Meyer, S. and A. von Haeseler (2003). "Identifying site-specific substitution rates." Mol Biol Evol **20**(2): 182-189.

Fast approaches for support values

Stamatakis, A., et al. (2008). "A rapid bootstrap algorithm for the RAxML Web servers." Syst Biol **57**(5): 758-771.

Minh, B. Q., et al. (2013). "Ultrafast approximation for phylogenetic bootstrap." Mol Biol Evol **30**(5): 1188-1195.

Anisimova, M. and O. Gascuel (2006). "Approximate likelihood-ratio test for branches: A fast, accurate, and powerful alternative." Syst Biol **55**(4): 539-552.

Guindon, S., et al. (2010). "New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0." Syst Biol **59**(3): 307-321.

Anisimova, M., et al. (2011). "Survey of branch support methods demonstrates accuracy, power, and robustness of fast likelihood-based approximation schemes." Syst Biol **60**(5): 685-699.

Thanks!