

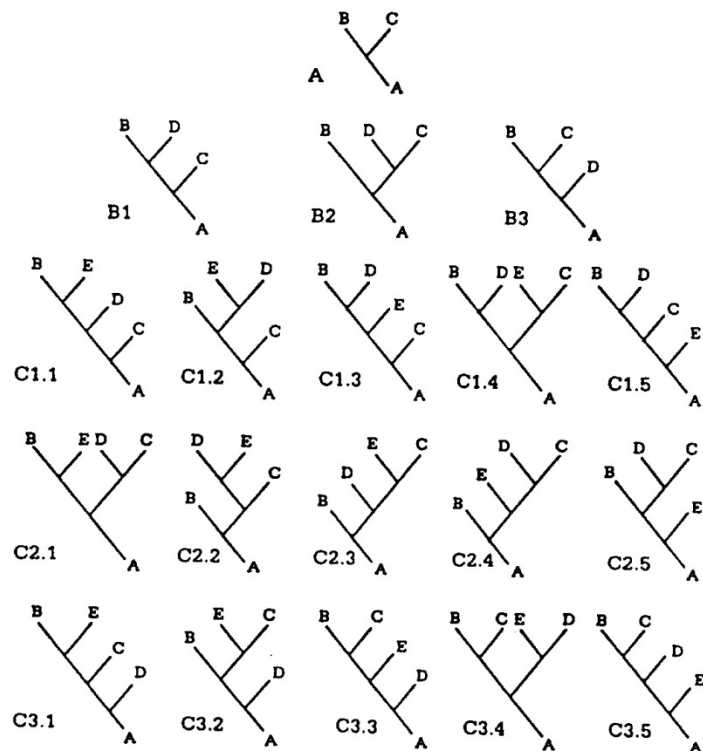
Parsimony analysis techniques for large data sets

Parsimony analysis techniques for large data sets

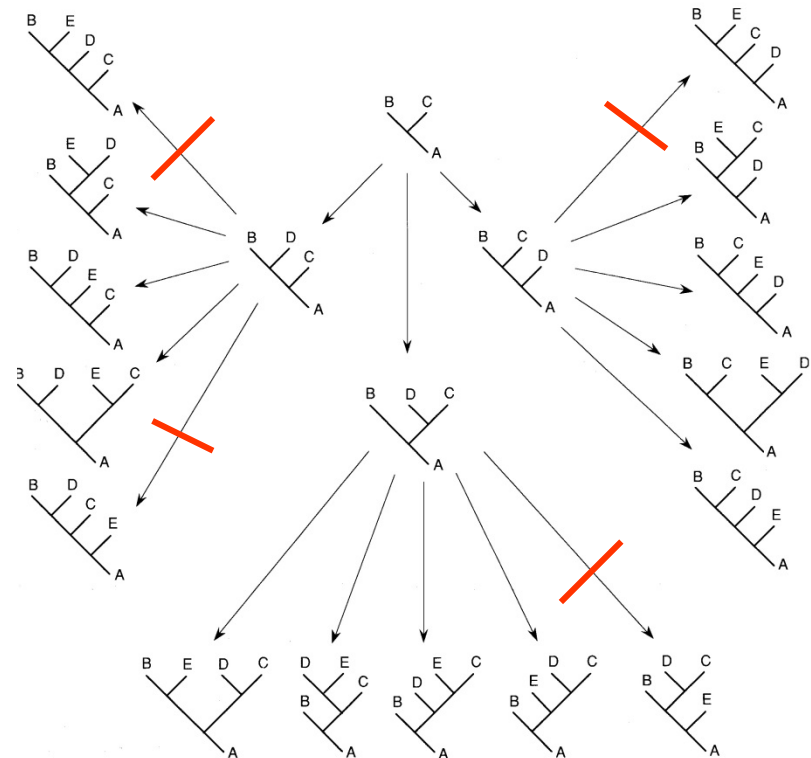
Traditional techniques

Small data sets (up to 25-30 taxa)

exhaustive search



branch-and-bound



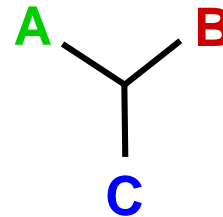
Parsimony analysis techniques for large data sets

Traditional techniques

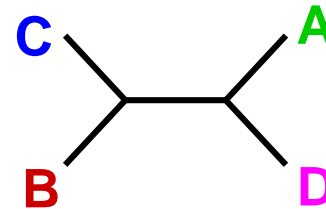
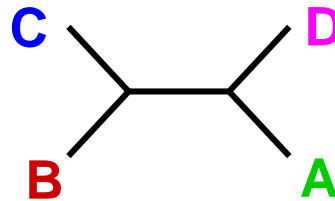
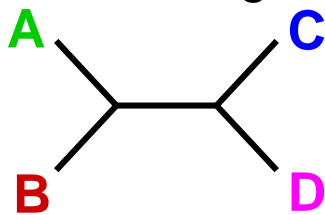
Middle size data sets (up to 200 taxa) – heuristic methods

(1) stepwise addition

First three objects are merged



Then the fourth is randomly selected and gradually added to the three existing branches



The individual trees are judged according to the optimization criteria and one or more of the shortest ones are left until the next round, where the fifth object is added, etc.

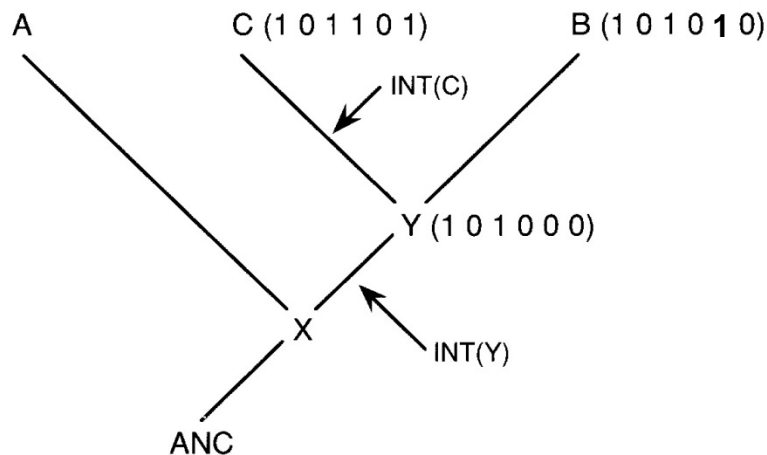
Parsimony analysis techniques for large data sets

Traditional techniques

Middle size data sets (up to 200 taxa) – heuristic methods

(2) Wagner trees

- they are created by gradually adding taxa to the place on the tree that corresponds to the most parsimonious solution



- problem: taxa are placed in the place on the tree that corresponds to the most parsimonious solution with respect to the taxa already on the tree, the most parsimonious location of the taxon on the overall tree might be different

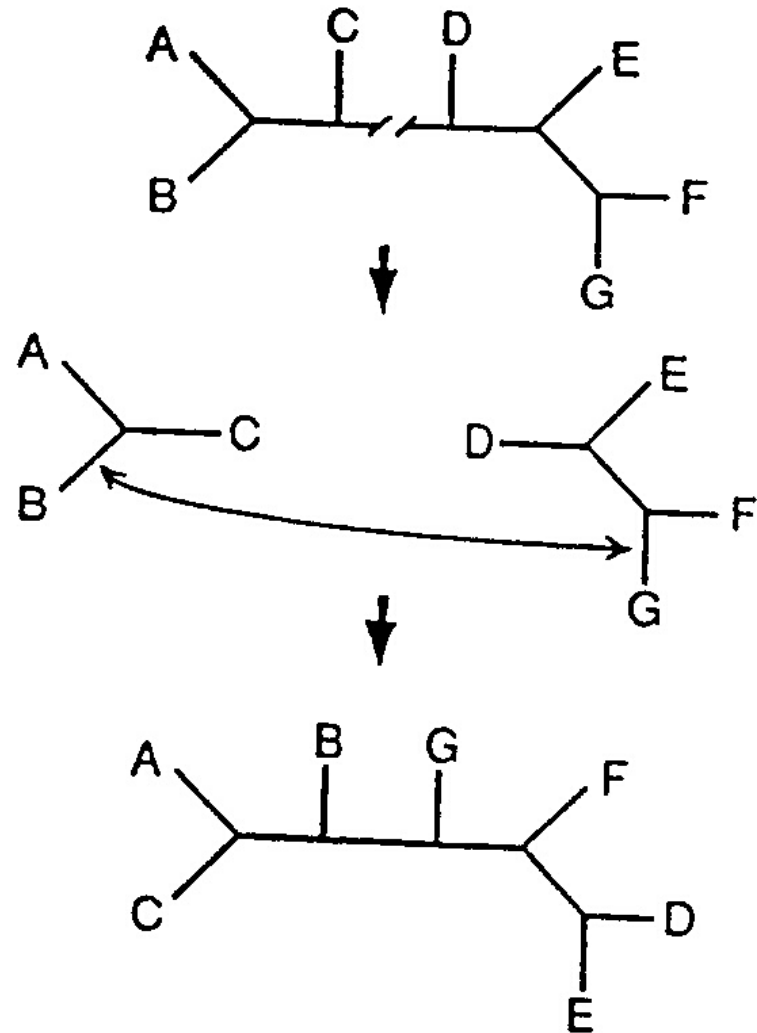
Parsimony analysis techniques for large data sets

Traditional techniques
(3) branch-swapping

tree bisection and reconnection
(TBR)

A separate part of the tree is added to each possible branch of the remaining part of the tree and the shortest resulting tree is searched

The number of such searched options increases with the cube of the number of taxa

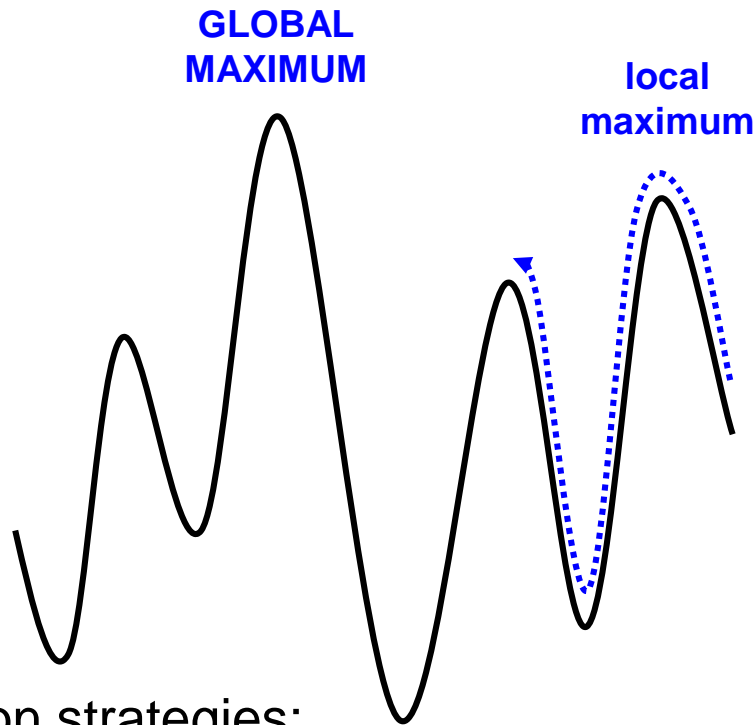


(if the number of taxa increases from 10 to 80, the time required for the search will increase 400x)

Parsimony analysis techniques for large data sets

Traditional techniques

TBR may not always find the shortest tree - the problem of local optima (islands)



Possible solution strategies:

- (1) RAS - random addition sequence - during the repeated formation of Wagner trees, taxa are added in random order (RAS + TBR strategy)
- (2) When swapping, suboptimal trees are also left for the next round (less effective strategy)

Parsimony analysis techniques for large data sets

Strategy used in the program NONA (NONA strategy)

- (1) Reduces the number of (Wagner) trees left from each replication, minimizing time spent on each "island"
- (2) Maximizes the number of initial trees from which the search begins (number of replications)
- (3) The resulting trees are collected from all replications and this file is used for a complete analysis, leaving more trees

Data set of 500 sequences *rbcl* gene (Chase et al. 1993) – Zilla data set:

1 month, PAUP, Mac, TBR from one replication – length of 16225 steps
11.6 months, PAUP, Sun, strategy of leaving more trees in each step (as much as RAM allowed), 8 replications - 8000 trees with a length of 16220 steps (Rice et al. 1997)

NONA program, Sun (left 2 trees from replication, combined results of 20 replications, then TBR with leaving 100 trees) - tree length 16220 steps on average every 78 hours, tree length 16218 steps on average every 150 hours

Parsimony analysis techniques for large data sets

Large data analysis techniques (> 500 taxa)

(over 500 taxa is no longer an effective NONA strategy either)

Parsimony ratchet

Sectorial searches

Tree fusing

Tree drifting

+ their combinations

Parsimony analysis techniques for large data sets

Parsimony ratchet

- (1) An initial tree is generated - a random arrangement of taxa, the creation of a Wagner tree, TBR branch-swapping, leaving a few trees (1-2)
- (2) The tree found in step 1 will be used as a starting point for the iterative strategy
- (3) A randomly selected subset of characters (5-25% of informative characters) is perturbed (distorted); typically the selected characters will double their weight or a jack-knife will be used where these characters will be omitted (their weight will be reduced to 0)
- (4) The tree found in step 1 is swapped with TBR using changed character weights, calculating its length, leaving one or a few trees when swapping - this "handles" the transition from one "island" to another

Parsimony analysis techniques for large data sets

Parsimony ratchet

(5) The character weights are returned to the original state (with the same weights, or different weights if these were at the beginning of the analysis), the tree created in step 4 is used to "swap" (leaving one or a few trees) until for "undisturbed" data does not find the shortest tree

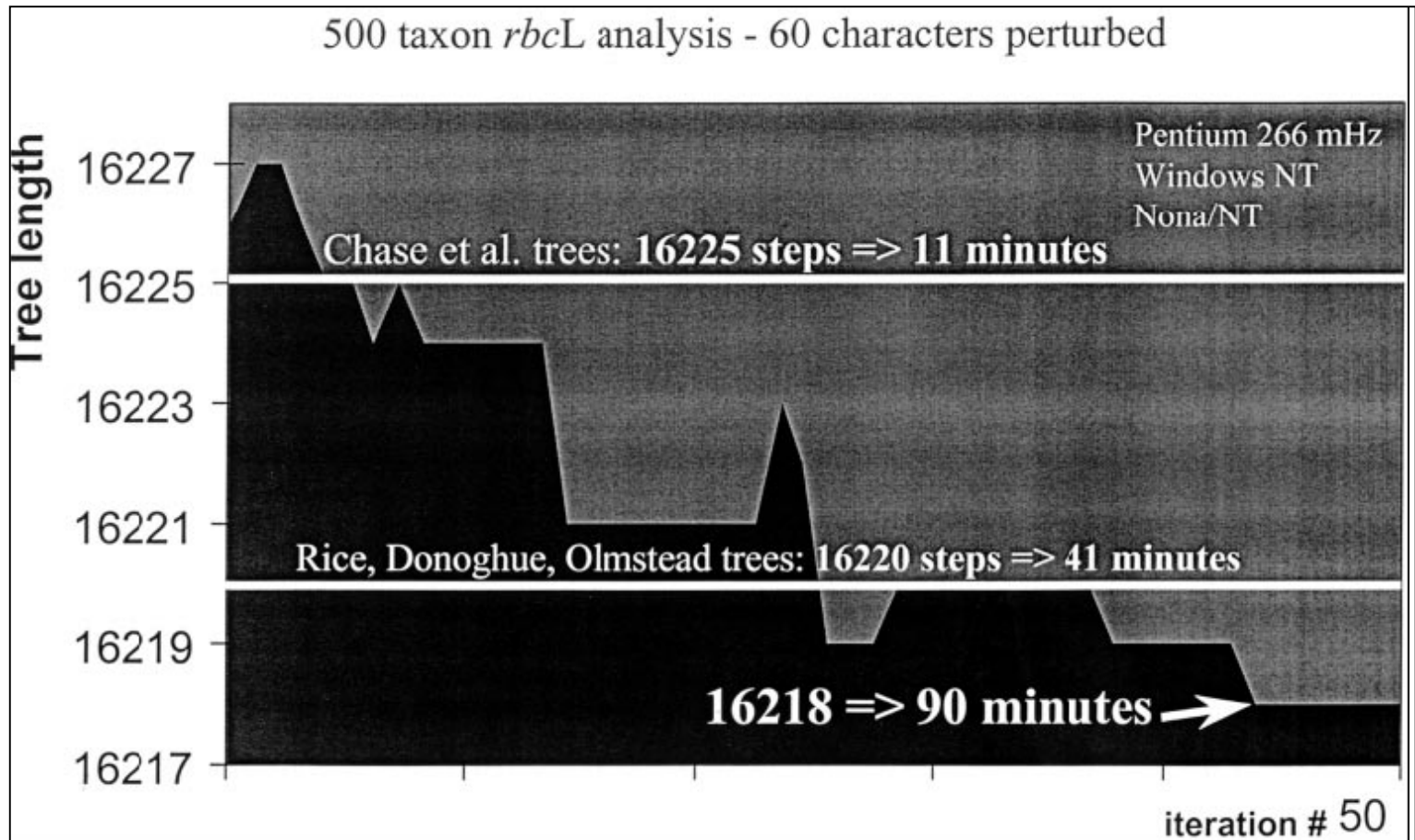
(6) The analysis returns to step 3, where the randomly selected character set is perturbed

The procedure is implemented in several programs - TNT, NONA, PAUPRat (parsimony ratchet application in the PAUP program)

Cladistics **15**, 407–414 (1999)

Kevin C. Nixon

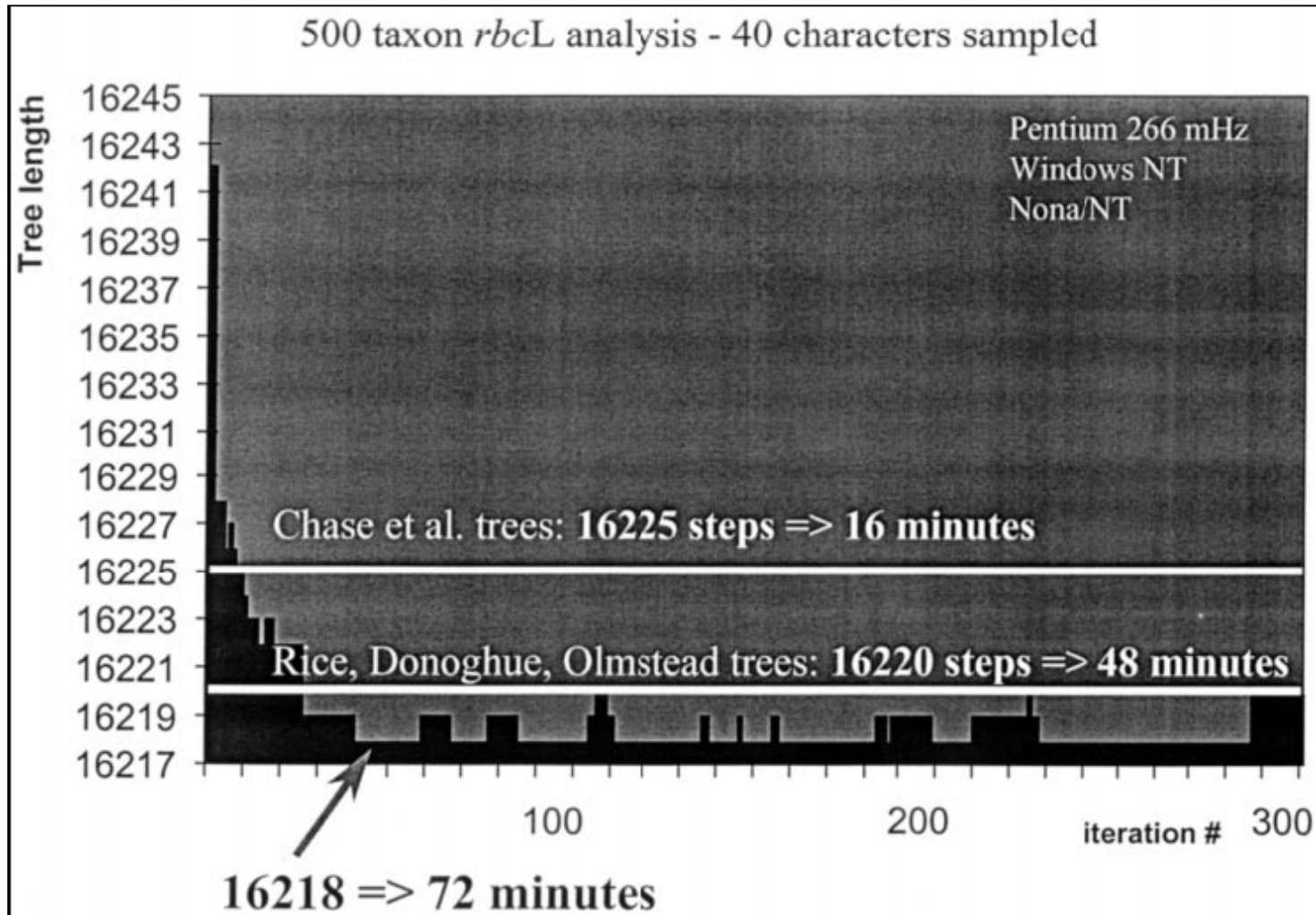
The Parsimony Ratchet, a New Method for Rapid Parsimony Analysis



Cladistics **15**, 407–414 (1999)

Kevin C. Nixon

The Parsimony Ratchet, a New Method for Rapid Parsimony Analysis



Parsimony analysis techniques for large data sets

Parsimony ratchet

Advantages:

The "islands" of the trees are not searched as intensively as in the PAUP program, which does not generate an unnecessarily large number of suboptimal trees with a similar topology as the shortest trees on the island.

Unlike the NONA strategy, it does not start building a new tree in every "replication" but starts with a tree that retains information from already found trees (but they can be from another "island")

Disadvantages:

For smaller data files, it may not always find the same number of trees as a search in the PAUP program, and the resulting consensus tree from the TNT program may have a higher resolution than the tree from the PAUP program.

Parsimony analysis techniques for large data sets

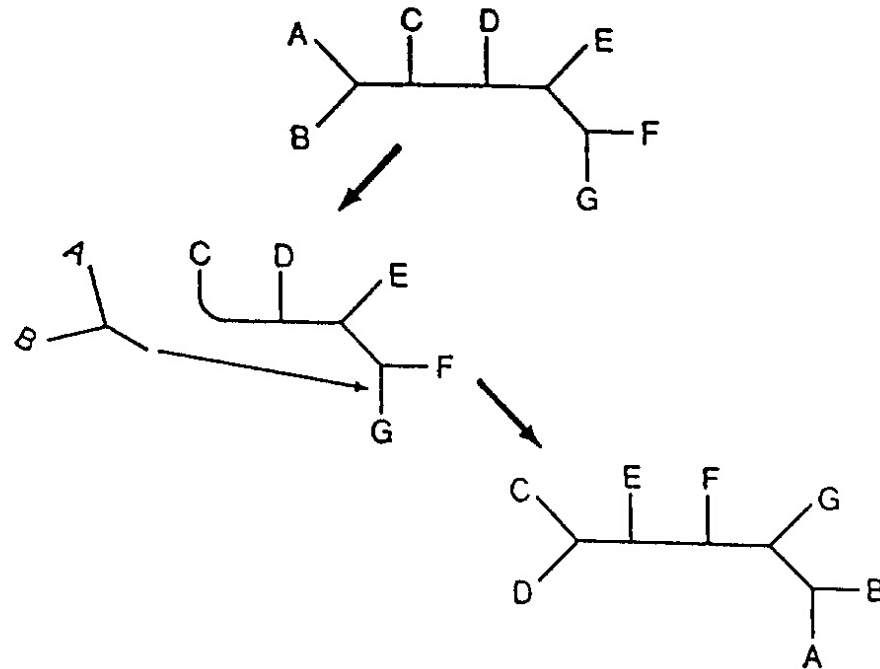
Tree fusing

The basic idea of the method is the exchange of subgroups of taxa between trees (in order to find a shorter tree)

Exchanges include all groups with more than 5 taxa that are present on a strict consensus tree made up of trees between which subgroups are exchanged (groups should not be equally dichotomously divided within)

Order of steps:

- (1) A tree is randomly selected ("target tree")
- (2) One of the remaining trees is randomly selected ("source tree"), **if there is no longer another tree to merge, SPR is swapped and the procedure returns to step 1**
- (3) The result of moving each group of taxa from the source to the destination tree is evaluated and the process returns to step 2



subtree pruning and regrafting (SPR swapping)

Parsimony analysis techniques for large data sets

Tree fusing

If the trees we have are suboptimal, tree fusion will almost always create trees that are closer to the optimum.

Goloboff (1999) states that 10 x RAS + TBR on the Zilla data set takes about 5 minutes and the resulting trees are 16,225-16,230 steps long, a few seconds of tree fusion application reduced the length of the tree for 16,220-16,222 steps

It is likely that at 10 x RAS + TBR each of the sectors will be in the optimal length on at least one of the trees - what we need is to combine these optimal sectors from the individual trees

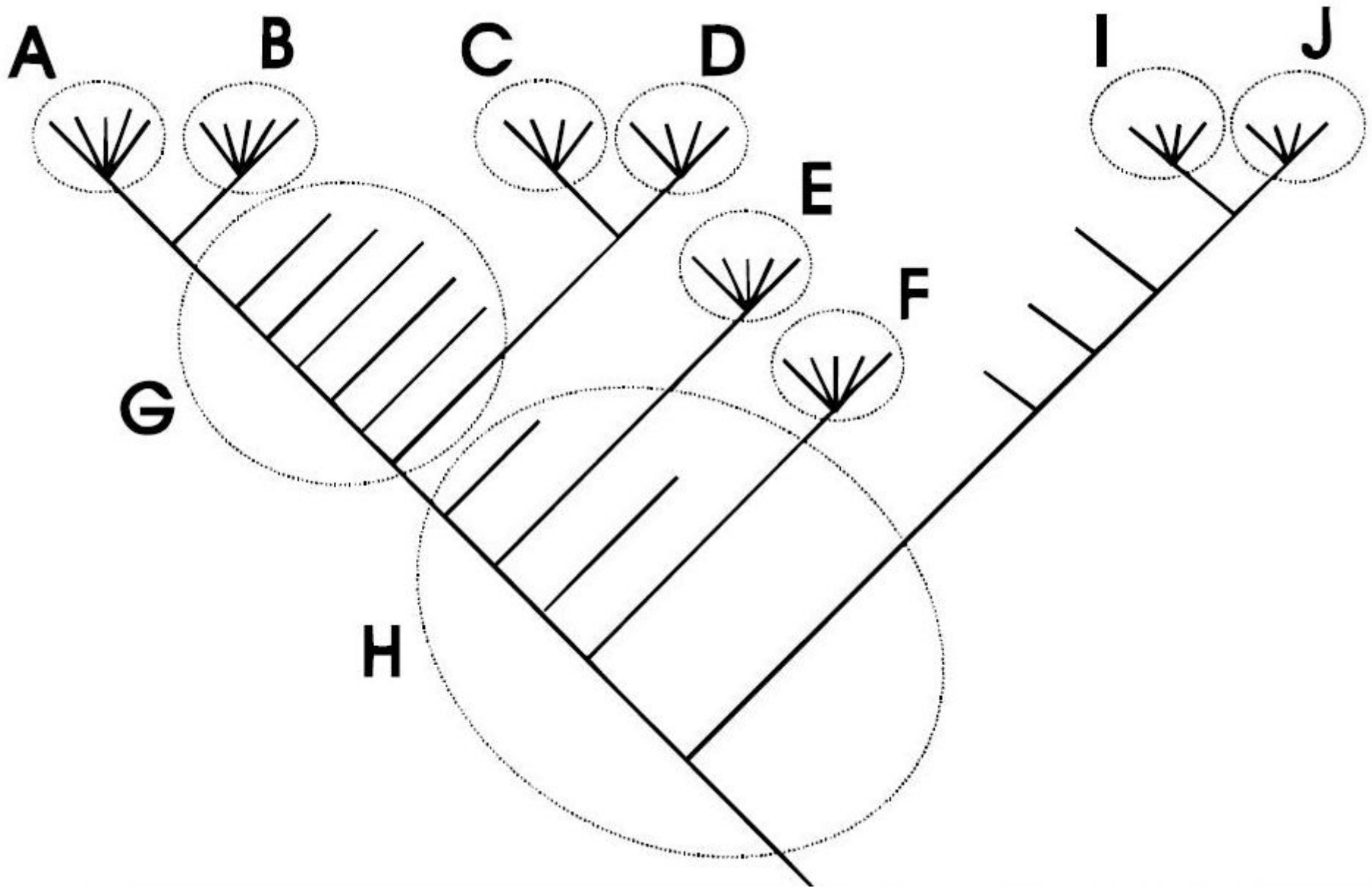
Parsimony analysis techniques for large data sets

Sectorial searches

The basic idea of the method is the selection of sectors from the trees and their separate analysis, if it is possible to achieve a better configuration in this sector, it is returned to the original tree

Sectors are selected either randomly, on the basis of a consensus tree, or in a mixed manner

After a certain sector of the tree (set of taxa) is selected, the data file is partially simplified compared to the original, from several (3-4) character states they become binary and some of the characters cease to be informative (they are identical in the whole data subset). This speeds up the analysis compared to the original data set



Sectors marked on the tree

Parsimony analysis techniques for large data sets

Sectorial searches

RSS – random sectorial searches

Order of steps:

- (1) A sector is randomly selected from the tree to contain approximately 35-55 terminal taxa (individuals)
- (2) About 3 replications will be made with the reduced set, if these replications do not improve the topology, proceed to step 3, if the topology improves, three more replications will be made and proceed to step 3
- (3) The best result from step 2 is selected and placed on the overall tree
- (4) After 5-10 exchanges on the tree, the whole tree will be swapped
- (5) The whole procedure is repeated, for the set Zilla 20-25 repetitions proved to be effective, for the set with 854 taxa 50-60 repetitions

Parsimony analysis techniques for large data sets

Sectorial searches

CSS – consensus sectorial searches

The procedure is similar to a random sector search, differing only in the way the sectors are selected

Sectors here are selected from a consensus tree (created in another analysis), where the polytomies in the consensus tree represent a conflict in the data, the number of taxa (individuals) in the selected sector should be approximately 10

MSS – mixed sectorial searches

Each replication starts with RAS + SPR, as soon as the SPR is completed, a consensus tree is created with the tree from the previous replication (better resolution than with CSS)

XSS – exclusive sectorial searches

The tree is divided into the same non-overlapping sectors, which together cover the whole tree

Parsimony analysis techniques for large data sets

Tree drifting

The basic idea of the method is to repeat TBR alternately with acceptance of only optimal and optimal and suboptimal trees ("drift" phase)

Suboptimal trees are accepted during the "drift" phase with the probability to which they are suboptimal. What is important here is the probability of accepting a suboptimal tree, which is based on the absolute difference in the number of steps and on the scale of the character conflict.

Trees that are so long or better are always accepted

After a certain number of accepted changes in the "drift" phase, the analysis returns to accepting only optimal trees

The process is repeated several times

The best results are achieved by a combination of methods: the optimal procedure involves RAS + TBR, then sector search, then "drift" or "ratchet" and the results are then merged

Parsimony analysis techniques for large data sets

In the parsimony analysis, all the most parsimonious trees were always searched for.

A more efficient approach is to find the minimum number of most parsimonious trees that will provide the same consensus tree.

This can be better achieved by finding multiple independent shortest trees most likely from different "islands" rather than TBR by "swapping" trees from the same "island".

As the number of equally parsimonious trees increases, the consensus trees become less and less resolved, over time the resolution becomes stable - at this point it is likely that a tree with the same resolution as from all equally parsimonious trees is obtained.

The whole process can be repeated to gain certainty






Fundación Miguel Lillo
Ministerio de Educación de la Nación
Tucumán - República Argentina

TNT

“Tree analysis using New Technology”

Goloboff, Farris, & Nixon, 2003

	Download last update: Feb. 21, 2022 (except Mac) (note Lin32 and Mac32 are discontinued)
 Windows	Menu interface – standard version Menu interface – no taxon limit Command-driven version Cygwin version (new, including MPI!)
 Mac	Mac 64 Mac 64 – no taxon limit
 Linux	Linux 64 Linux 64 – no taxon limit Linux 64 – with MPI

TNT stands for "Tree analysis using New Technology". It is a program for phylogenetic analysis under parsimony (with very fast tree-searching algorithms; Nixon, 1999, Cladistics 15:407-406; Goloboff, 1999, Cladistics 15:407-428), as well as extensive tree handling and diagnosis capabilities. It is a joint project by Pablo Goloboff, James Farris, and Kevin Nixon.

<http://www.lillo.org.ar/phylogeny/tnt/>

TNT version 1.5, including a full implementation of phylogenetic morphometrics

Pablo A. Goloboff^{ff^{a,*}} and Santiago A. Catalano^{a,b}

^a*Unidad Ejecutora Lillo, Consejo Nacional de Investigaciones Científicas y Técnicas, Miguel Lillo 251, 4000 S.M. de Tucumán, Argentina;*

^b*Facultad de Ciencias Naturales e Instituto Miguel Lillo, Universidad Nacional de Tucumán, Miguel Lillo 205, 4000 S.M. de Tucumán, Argentina*

Accepted 29 February 2016

Abstract

Version 1.5 of the computer program TNT completely integrates landmark data into phylogenetic analysis. Landmark data consist of coordinates (in two or three dimensions) for the terminal taxa; TNT reconstructs shapes for the internal nodes such that the difference between ancestor and descendant shapes for all tree branches sums up to a minimum; this sum is used as tree score. Landmark data can be analysed alone or in combination with standard characters; all the applicable commands and options in TNT can be used transparently after reading a landmark data set. The program continues implementing all the types of analyses in former versions, including discrete and continuous characters (which can now be read at any scale, and automatically rescaled by TNT). Using algorithms described in this paper, searches for landmark data can be made tens to hundreds of times faster than it was possible before (from T to $3T$ times faster, where T is the number of taxa), thus making phylogenetic analysis of landmarks feasible even on standard personal computers.

© The Willi Hennig Society 2016.

Parsimony analysis techniques for large data sets

Format of the data file for the program TNT (identical with the format of the program Hennig86)

```
xread
'Empetrum'
28 14
enkianthus      0000000000001000100000??0000
daboecia        0000000000001000010000??0100
album           0110100101100111110112001100
conradii        0110100101100111110111?1111?
ceratiola       0110100100010112110111101110
nigrum          1101100210010111111112310110
hermaphrod      11011002100100111111112310111
subholarct     11001102100101111111112310111
kardakovii     110010021001011111111124101??
eamesii        1110100210010011111111241011?
atropurpur     11101002100100111111112510111
medium         ?1101002100100111111112210110
rubrum         0101100210010111111111221011?
maclovian      21101012100101111111112210110
;
```

V programe TNT je možné načítať aj nexus formát

Parsimony analysis techniques for large data sets

T. N. T.

Tree Analysis Using New Technology

Version 1.0 © P. Goloboff, J. S. Farris, and K. Nixon

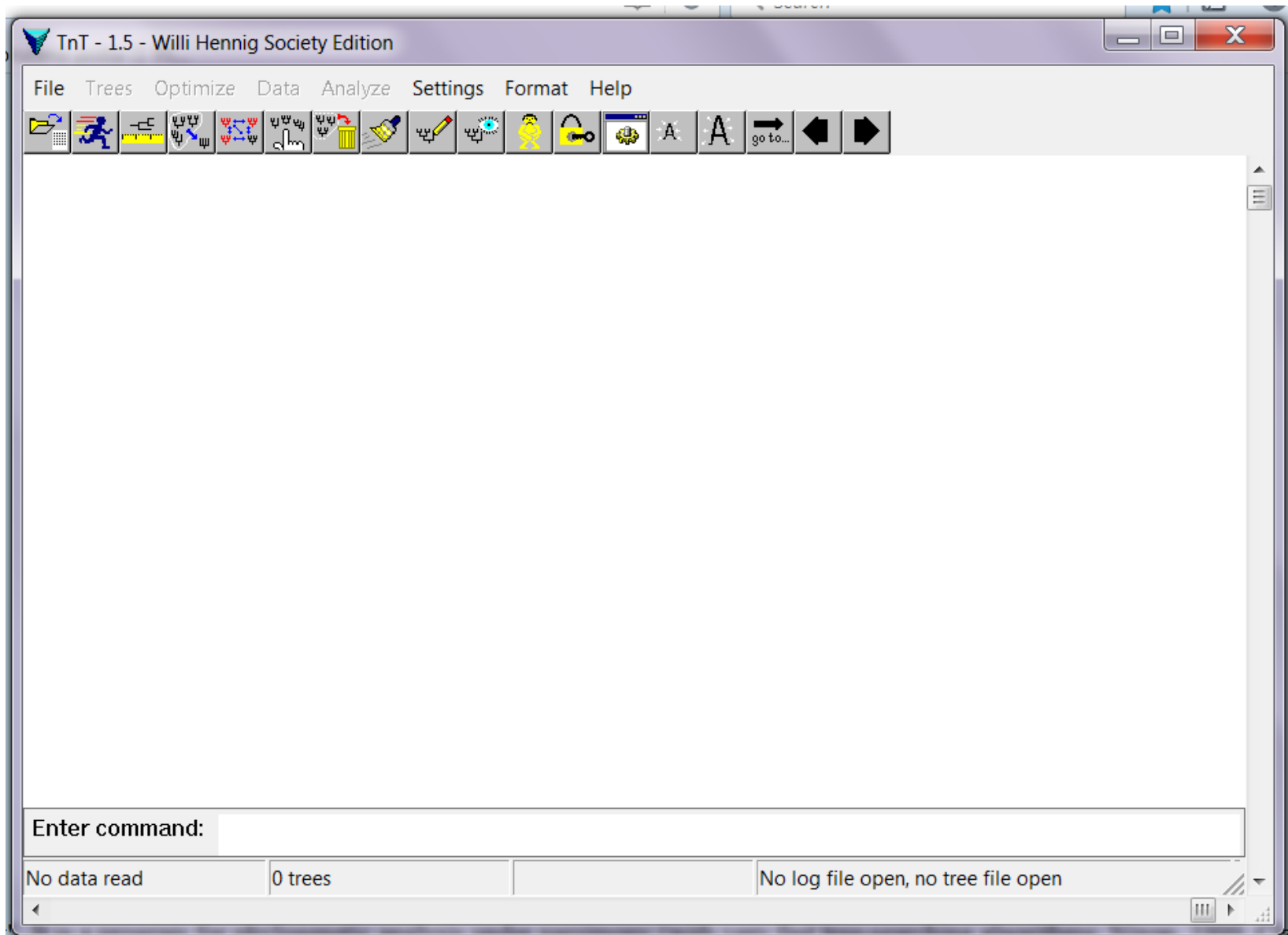
[Introduction-Getting Started](#)
[What TNT doesn't do...](#)
[Printing & Metafiles](#)
[Data Input](#)
[Continuous characters](#)
[Merging Data Files](#)
[Basic Character Settings](#)
[Scopes, sets, and blocks](#)
[Character and block names](#)
[Step-Matrices](#)
[Ancestral states](#)
[Nexus files](#)
[Implied Weighting](#)

[Saving suboptimal trees](#)
[Memory Management](#)
[Tree Collapsing](#)
[Command-Line](#)
[Batch Menus](#)
[Measures of support](#)
[Linux versions](#)
[Consensus estimation](#)
[Search algorithms](#)
[Setting search parameters](#)
[Implementation notes...](#)
[Scripting\(advanced users\)](#)
[Citations](#)

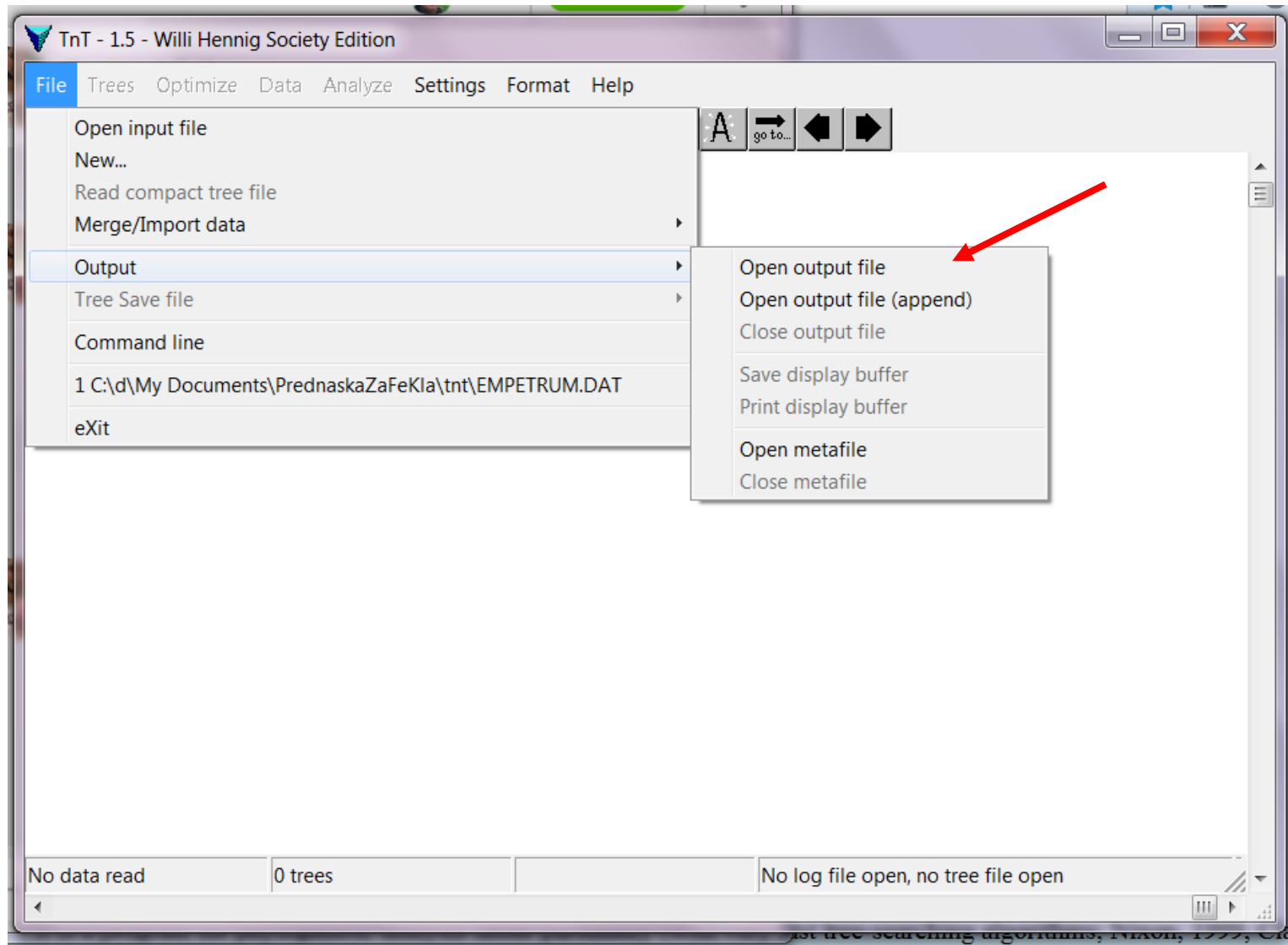
Introduction - Getting Started ([back to index](#))

TNT is a program for phylogenetic analysis under parsimony. It provides fast tree-searching algorithms, as well as extensive capabilities for tree diagnosis, consensus, and manipulation. This documentation explains some important things that are not immediately apparent (or cannot be done easily) from the menus themselves. For details on how to use commands, users must refer to the on-line help of TNT (accessed typing *help* <enter> at the command prompt or the command dialog opened with [File/CommandLine](#)). This file, [tnt.htm](#), provides a general description of the program; if the file is copied to your Windows directory,

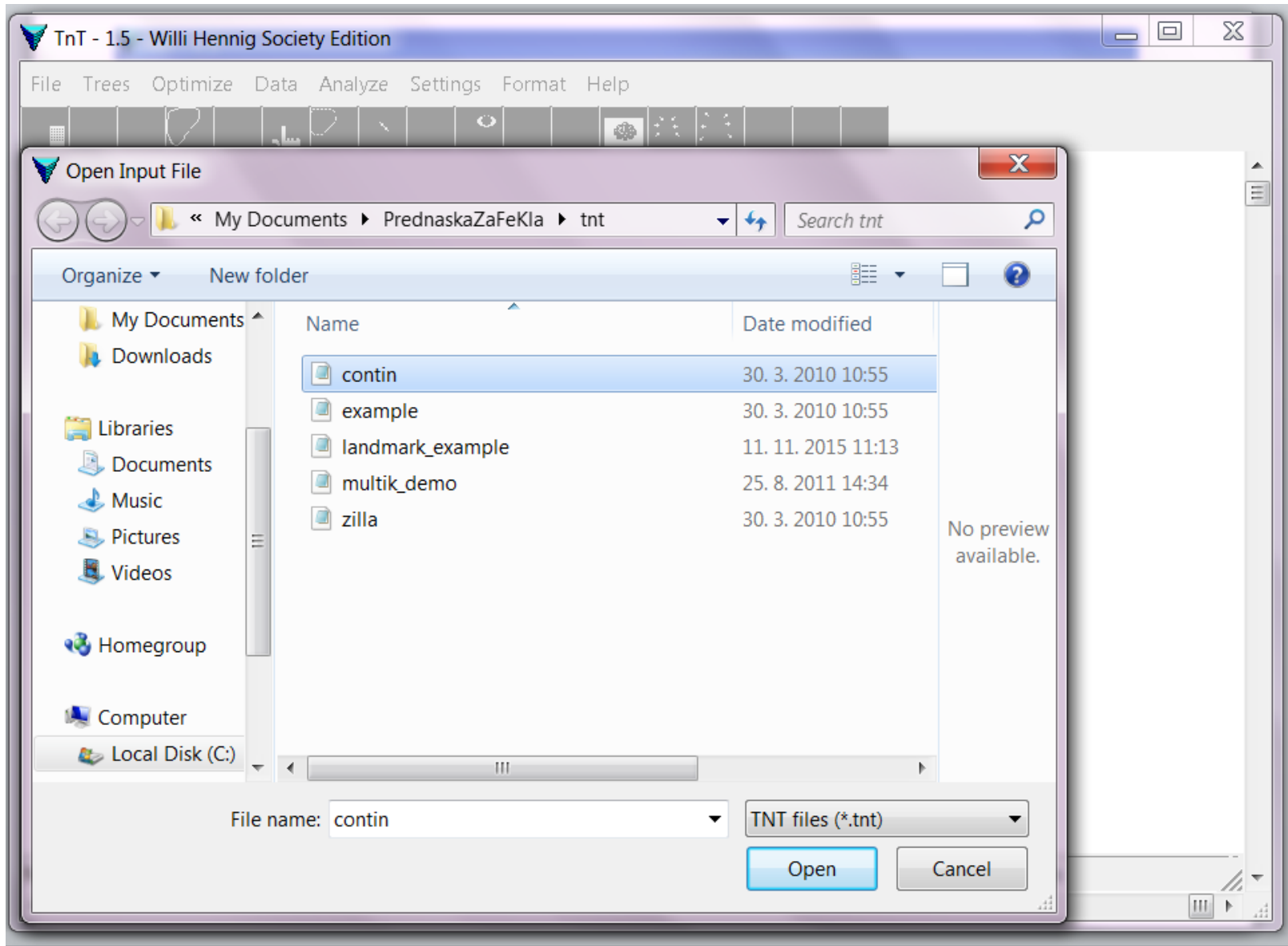
"Help" for the program is in the form of an html file and is installed in the program directory when unpacking the file "ziptnt.exe", copied to the directory c: \ WINDOWS \, then works as Help in the program itself



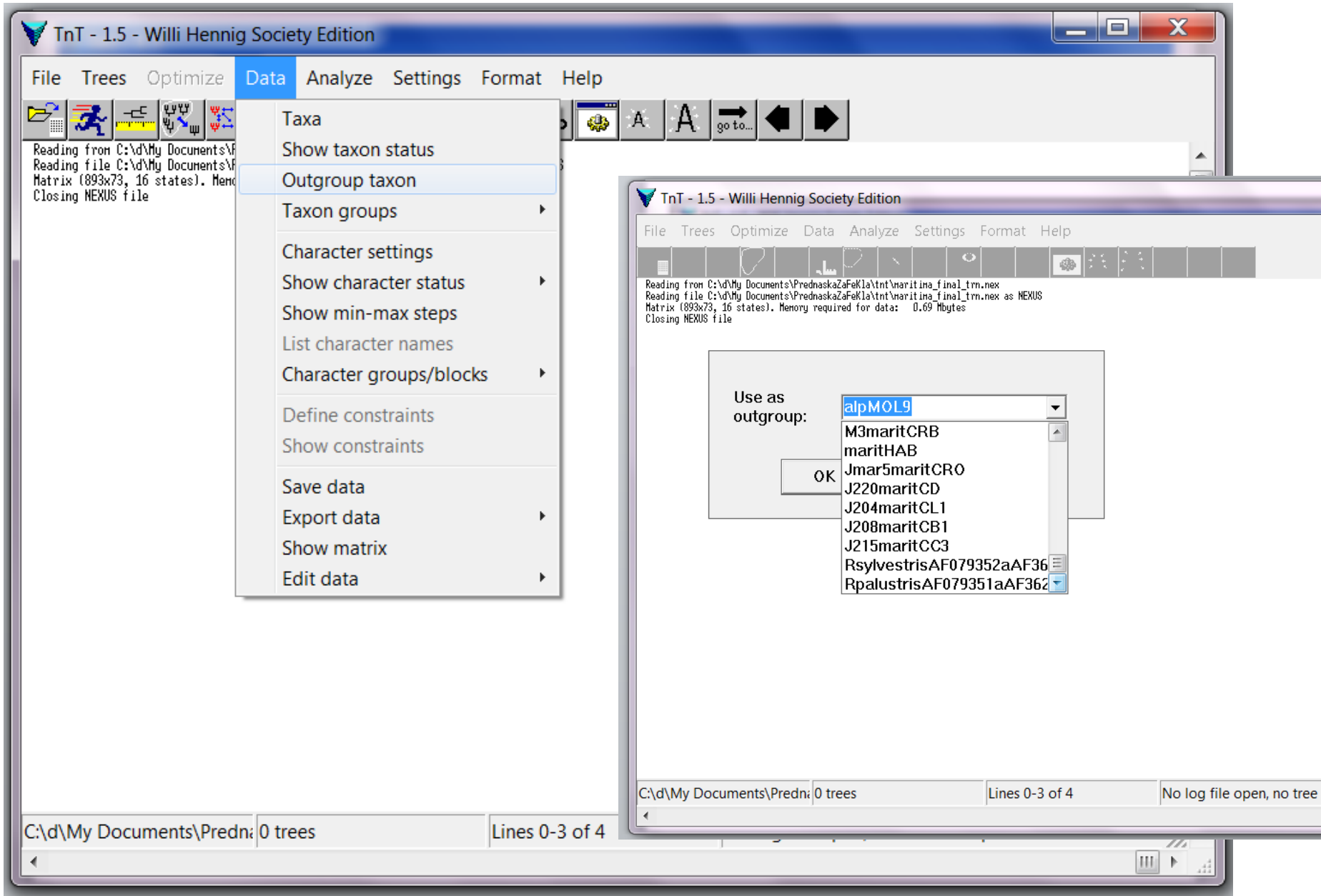
Introductory page of the program TNT



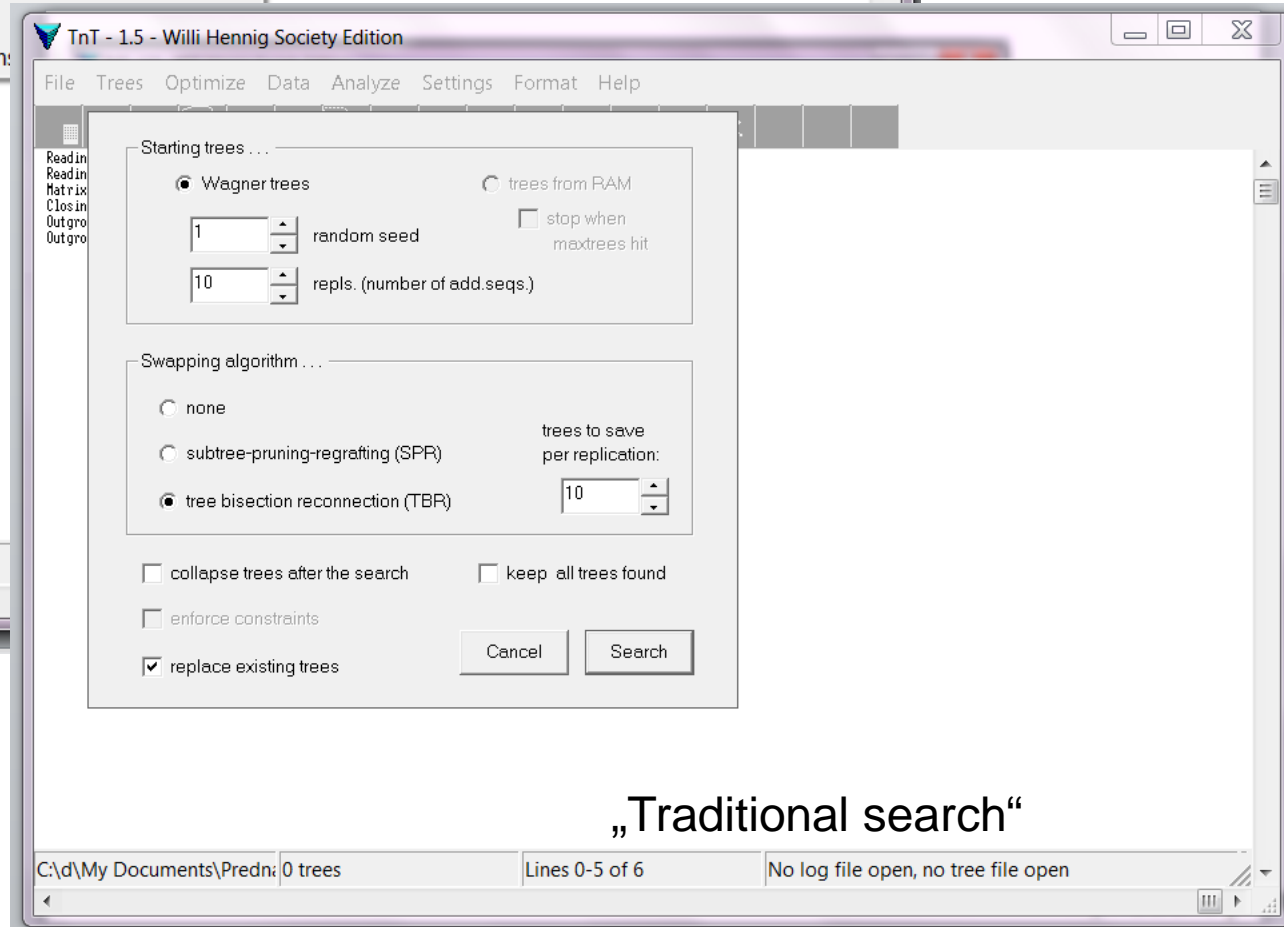
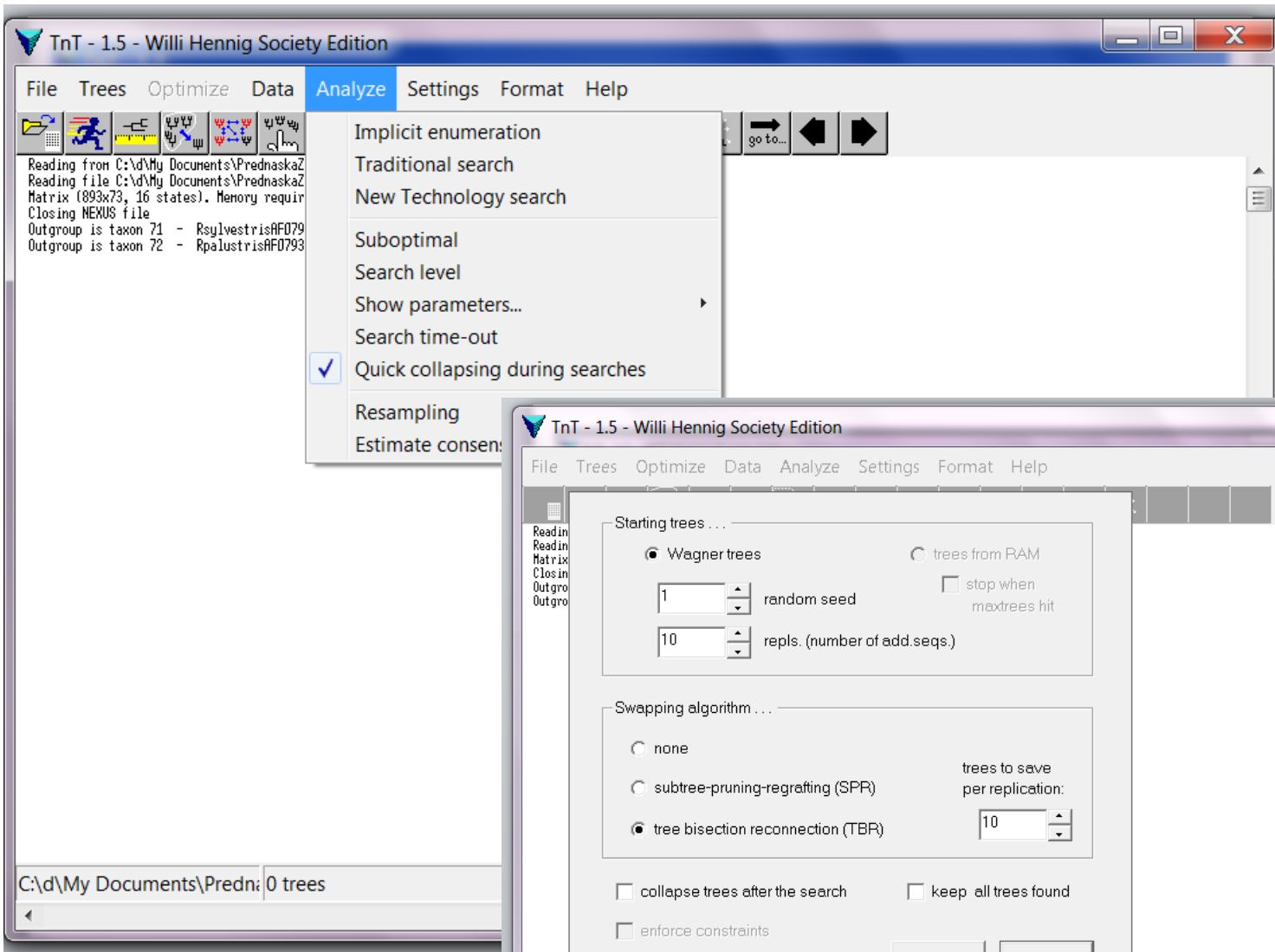
Opening of the .log file for sequence of steps and the results of the analysis



Opening the data file – in the format of the program TNT or in the nexus format



Selection of the outgroup

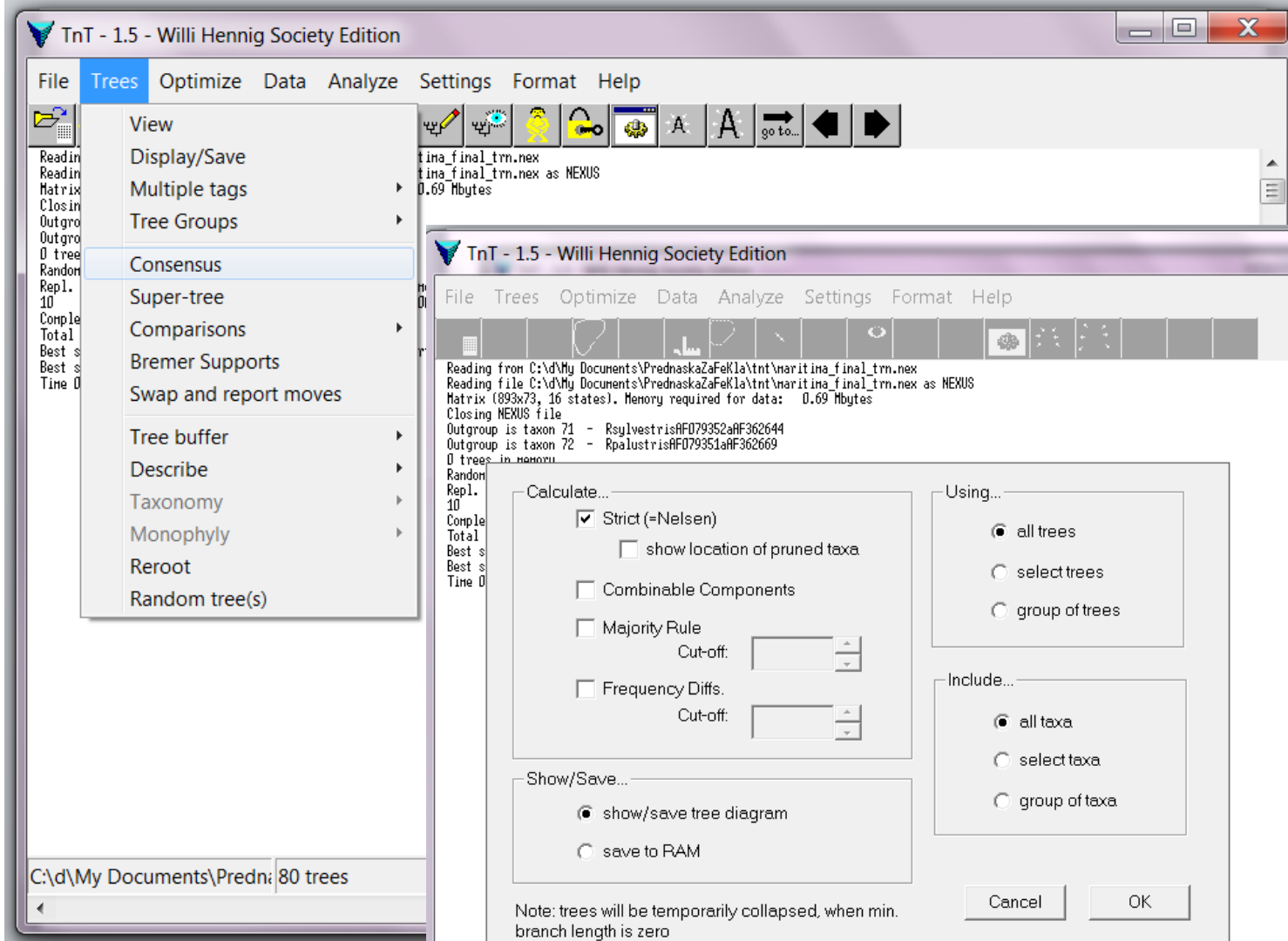


„Traditional search“



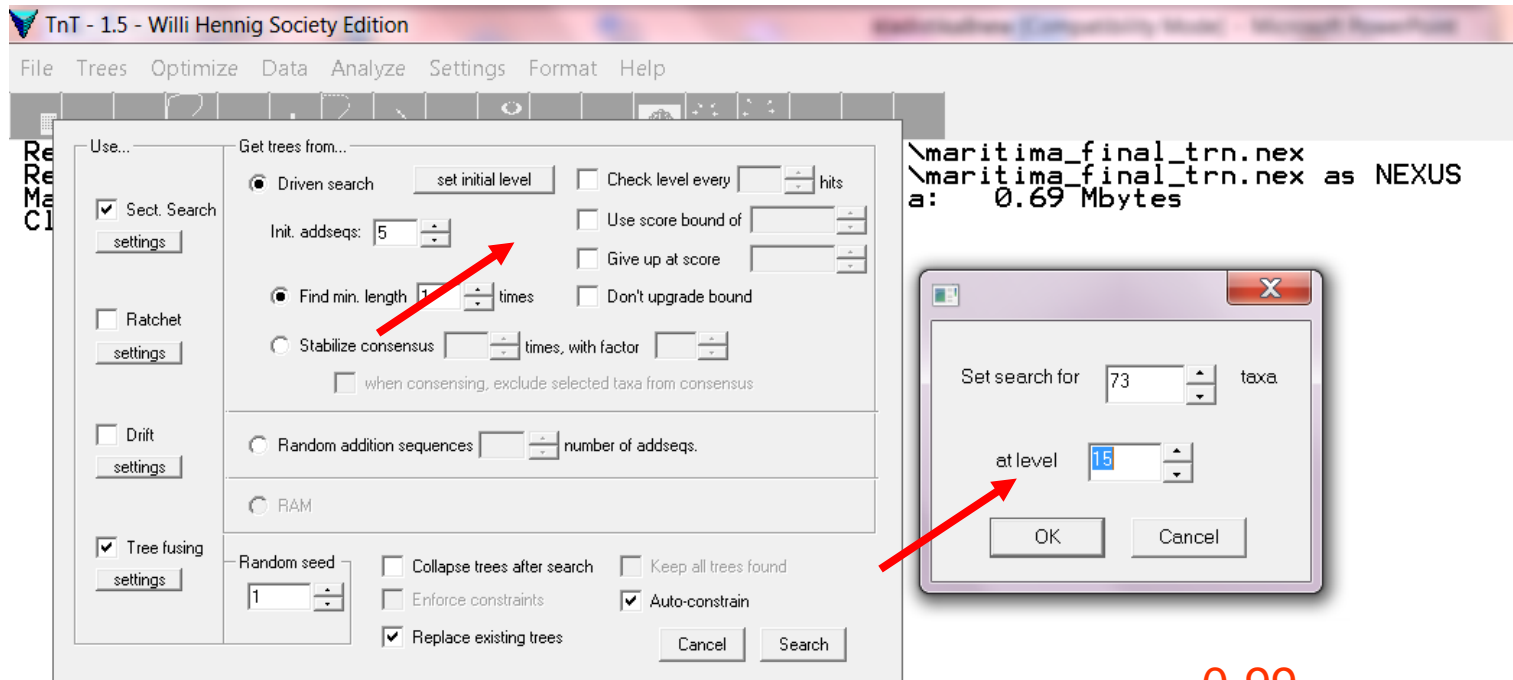
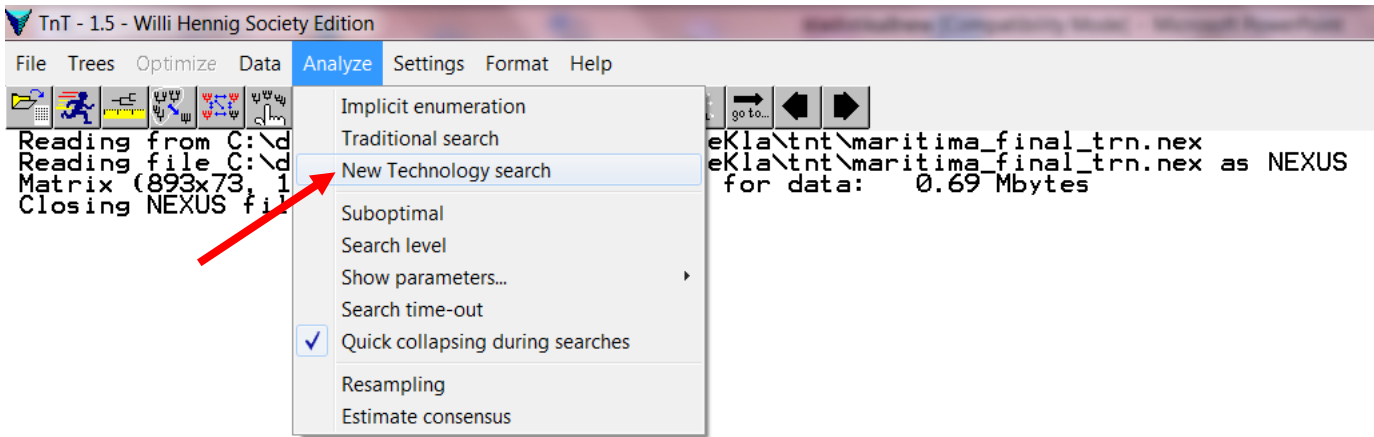
```
Reading from C:\My Documents\PrednaskaZaFeKla\tnt\maritina_final_trn.nex
Reading file C:\My Documents\PrednaskaZaFeKla\tnt\maritina_final_trn.nex as NEXUS
Matrix (893x73, 16 states). Memory required for data: 0.69 Mbytes
Closing NEXUS file
Outgroup is taxon 71 - RsylvestrisAF079352aAF362644
Outgroup is taxon 72 - RpalustrisAF079351aAF362669
0 trees in memory
Random seed is 1
Repl. Algor.   Tree       Score      Best Score  Time      Rearrang.
10   TBR       79 of 80   -----    353       0:00:00   6,873,547
Completed 10 random addition sequences.
Total rearrangements examined: 6,873,547.
Best score hit 10 times out of 10 (some replications overflowed).
Best score (TBR): 353. 80 trees retained.
Time 0.14 secs.
```

Result of the traditional search



Strict consensus tree

Strict consensus tree



Techniques for large data sets (> 500 taxa)

Setting the intensity of the search



- Use...
- Sect
setting
- Rato
setting
- Drift
setting
- Tree
setting

Ratchet parameters ...

Stop perturbation phase when ...

20 substitutions made, or 99 % swapping completed

Perturbation phase ...

4 Up-weighting prob.

4 Down-weighting prob.

Alternate equal weights

Number of iterations ...

0 Total number

0 Auto-constrained iterations

Cancel OK

a_final_trn.nex
a_final_trn.nex as NEXUS
9 Mbytes

Setting the parameters of the method

Constraints is OFF
Random seed is 1 „Driven search“ – results of the search at the intensity level 15
0 trees in memory

Repl. Algor.	Tree	Score	Best Score	Time	Rearrangings.
5 FUSE	5	-----	-----	0:00:00	1,304,941

Completed search.
Total rearrangements examined: 1,304,941.
No target score defined. Best score hit 1 times.
Best score: 353. 8 trees retained.
0.28 secs.

Constraints is OFF
Random seed is 1 „Driven search“ - results of the search at the intensity level 99
0 trees in memory

Repl. Algor.	Tree	Score	Best Score	Time	Rearrangings.
13 FUSE	13	-----	353	0:00:01	51,132,385

Completed search.
Total rearrangements examined: 51,132,385.
No target score defined. Best score hit 1 times.
Best score: 353. 19 trees retained.
2.08 secs.

Random addition sequences, 1000 replicates – results of the search

Space for 40000 trees in memory
Constraints is OFF
Random seed is 1
0 trees in memory

Repl. Algor.	Tree	Score	Best Score	Time	Rearrangings.
1000 FUSE	1000	-----	353	0:03:14	4,825,467,220

Completed search.
Total rearrangements examined: 4,825,467,220.
No target score defined. Best score hit 1 times.
Best score: 353. 168 trees retained.
194.92 secs.

For help on command "xxxx" enter "help xxxx" or "xxx?"
 For help on ALL commands, enter "help*"
 For help with scripting language, enter "help+."
 Enter "help [topic]" for possible help topics.

COMMANDS:

agroup	alltrees	ancstates	apo	bbreak	beep	best
blength	blocks	break	bsupport	ccode	cdir	change
chkmoves	chomo	ckeeep	cls	clbuffer	cnames	collapse
comcomp	condense	constrain	costs	cscores	cstree	dmerge
drift	edit	echo	export	fit	fillsank	force
freqdifs	help	hold	hybrid	ienum	incltax	info
log	keep	length	lmark	lmbox	lmrealign	lquote
majority	matchtax	map	minmax	mixtrees	mono	mrp
mult	mxram	mxproc	naked	nelsen	nstates	outgroup
procedure	pause	pcrprune	pfijo	piwe	pruncom	prunmajor
prunnelsen	pruntax	qcollapse	qnelsen	quote	quit	randtrees
ratchet	rcompl	rdir	rebuild	recons	report	reroot
resample	resols	rfreqs	riddup	rseed	run	save
screen	scores	sectsch	shortread	shpcomp	silent	slft
smatrix	sort	sprdiff	subopt	svtxt	tables	taxcode
taxlabels	taxonomy	taxname	tchoose	tcomp	tequal	tfuse
tgroup	timeout	tnodes	tplot	tread	tsave	tshrink
tsize	ttags	txtsize	tzert	view	vversion	warn
watch	xcomp	xgroup	xinact	xmult	xperm	xpiwe
xread	xwipe	unique	unshared	usminmax	window	zzz

Enter command: help

No data read

0 trees

Lines 0-26 of 27

No log file open, no



Depicts possible commands

chkmoves	chomo	ckeeep	cls	clbuffer	cnames	collapse
comcomp	condense	constrain	costs	cscores	cstree	dmerge
drift	edit	echo	export	fit	fillsank	force
frequdifs	help	hold	hybrid	ienum	incltax	info
log	keep	length	lmark	lmbbox	lmrealign	lquote
majority	matchtax	map	minmax	mixtrees	mono	mrp
mult	mxram	mxproc	naked	nelsen	nstates	outgroup
procedure	pause	pcrprune	pfijo	piwe	pruncom	prunmajor
prunnelsen	pruntax	qcollapse	qnelsen	quote	quit	randtrees
ratchet	rcompl	rdir	rebuild	recons	report	reroot
resample	resols	rfreqs	riddup	rseed	run	save
screen	scores	sectsch	shortread	shpcomp	silent	slfwt
smatrix	sort	sprdiff	subopt	svtxt	tables	taxcode
taxlabels	taxonomy	taxname	tchoose	tcomp	tequal	tfuse
tgroup	timeout	tnodes	tplot	tread	tsave	tshrink
tsize	ttags	txtsize	tzert	view	vversion	warn
watch	xcomp	xgroup	xinact	xmult	xperm	xpiwe
xread	xwipe	unique	unshared	usminmax	window	zzz

RATCHET

Ratchet, from trees in memory. Options are:

- iter N number of iterations
- [no]equal periodic rounds with original weights [not]
- numsubs N number of replacements (i.e. accepted tree rearrangements) to do in perturbation phase
- upfactor N probability of upweighting a character
- downfact N same, for downweighting
- [no]autoconst N number of auto-constrained cycles
- [no]giveup N percentage of full swap to complete during perturbation
- findscore N if score N or better found, stop
- [no]fuse NxR every N iterations, do R rounds of fusing to the N trees
- [no]dumpfuse if fusing fails to produce a better tree, [don't] dump all the suboptimal trees
- [no]tradrat [don't] run the original ratchet (i.e. noequal; during perturbation swap to completion and don't accept equally good rearrangements).

Options are set with "ratchet:[options];" or "ratchet=[options];"
 (first case changes settings only, second case runs as well). With
 "ratchet:;" current settings are displayed

Enter command: help ratchet

No data read

0 trees

Lines 8-48 of 49

No log file c

Depicts syntax of possible commands

Zimbra: Junk x http://www.lillo.../phylogeny/tnt/ x +

www.lillo.org.ar/phylogeny/tnt/

Zimbra: Inbox Most Visited Getting Started Doručená pošta (1 95...

Linux/Mac.- You have to open the data file yourself, and then run the script

```
proc filename.tnt ; aquickie ; [enter]
```

The script will provide you with a few choices, then run the analysis automatically

TNT

The program GB2TNT (Goloboff & Catalano, 2012, Cladistics 28: 503–510) processes data from the GenBank files, so that it is easy to diagnose results and color differences

The main documentation is in the file *tnt.htm*, and recent additions or bugfixes are in the file *BugFixes.htm*, both of these are contained within all the packages (except the character-mode Windows version). On-line help on all TNT commands can be obtained with the *help* command. A general description of TNT version 1.0 for Windows is in a [QuickTutorial](#) (a PowerPoint presentation; this is probably the easiest way to get started). Some example files for scripting come in the self-extracting files *zipdruns.zip* or *tnt_scripts.zip*, and additional scripts are [here](#) (with full [documentation](#) of the scripting language in a pdf file). Instructions to view large trees in Windows and process the taxonomy contained within taxon names (as in Goloboff et al., 2009, Cladistics 25:211-230) can be found [here](#)

data (including all the taxonomic information from the current version is 0.75) can be downloaded [here](#).

Opening QuickTutorial.zip

You have chosen to open:

QuickTutorial.zip
which is: WinZip File (892 KB)
from: http://www.lillo.org.ar

What should Firefox do with this file?

Open with WinZip (default)

Save File

Do this automatically for files like this from now on.

OK Cancel

Warning/Disclaimer

Keep in mind that freezes/crashes might occasionally occur, and save results frequently. The program is intended only for strictly academic use, and is provided "as is," with no express or implied warranties. None of the authors of TNT is responsible in any way for any problems the program causes to your computer, your data, your career, or your life. A copy of the license agreement can be found [here](#)

Bug Reports

All output produced by the program is saved to an internal text-buffer. The text-buffer is displayed in this window. The size of the buffer can be changed with *Settings / Memory*. The text-buffer can be saved to file at any time.

Help calls the explorer and displays the contents of the file *tnt.htm*. For this, the file *tnt.htm* must be copied to your windows directory. This provides detailed descriptions of many commands and options.

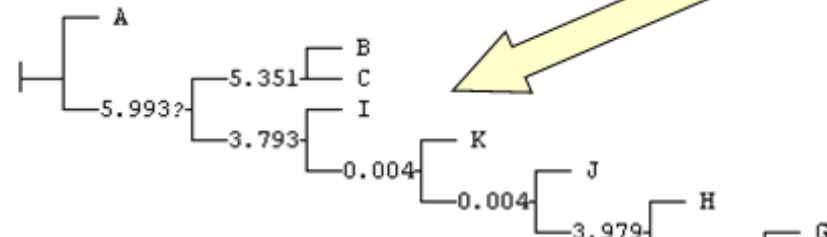
For proper viewing of trees, the font *Tred* should be installed in the system (go to Control Panel / Fonts / Install New Font)



Start swapping from 375 trees (score 47.932-53.932) ...
Repl. Algor. Tree Score Best Score
--- TBR 385 of 386 ----- 47.932
Completed TBR branch-swapping.
Total rearrangements examined: 356,005
Note: some trees of different length may become
Best score (TBR): 47.932-53.932. 386 trees found.
5.64 secs.

Note: for consensus calculation, trees will be temporarily collapsed (when min. branch length = 0)

Bremer supports (from 386 trees, cut 0)



Enter command:

C:\tnt\now\contin.tnt 386 trees Lines 45-65 of 70 No log file open, no tree file open

Parsimony analysis techniques for large data sets

Literature:

- Nixon, K. C. (1999): The Parsimony Ratchet, a new method for rapid parsimony analysis. – *Cladistics* 15: 407-414.
- Goloboff, P. A. (1999): Analyzing large data sets in reasonable times: solutions for composite optima. – *Cladistics* 15: 415-428.
- Goloboff, P. A. (2002): Techniques for analyzing large data sets. – In: DeSalle, R. et al., *Methods and tools in biosciences and medicine. Techniques in molecular systematics and evolution*. Birkhauser Verlag, Basel, pp. 70-79.
- Chase, M. W. et al. (1993): Phylogenetics of seed plants: an analysis of nucleic sequences from the plastid gene *rbcL*. – *Annals of Missouri Botanical Garden* 80: 528-580. [miesto uverejnenia dátového súboru „Zilla“]
- Rice, K. A., Donoghue, M. J. & Olmstead, R. G. (1997): Analyzing large data sets: *rbcL* 500 revisited. – *Systematic Biology* 46: 554-563. [pokus o novú analýzu dátového súboru „Zilla“]