POY, Phylogeny Reconstruction via Optimization of DNA and other Data
version 3.0.11 (May 6 2003)

Ward Wheeler[1], David Gladstein, and Jan De Laet[2]

[1] American Museum of Natural History, Central Park West at 79th Street, 10024 New York, NY USA
[2] Royal Belgian Insititute for Natural Sciences, Vautierstraat 29, B−1000 Brussels, Belgium

---

This is a dump of the command line help of POY, generated using "poy −helpdumpfile poy"
[and converted to pdf format using htmldoc v.1.8.23].
Citation for this document:

```
De Laet, J, and W. Wheeler. 2003.
     POY version 3.0.11 (Wheeler, Gladstein and De Laet, May 6 2003).
     Command line documentation.
```

Files: poy.html, poy.png
Send comments to Jan De Laet

[Contents]

Check AMNH ftp−site for

- availability of new versions
- additional documentation by Dan Janies and Ward Wheeler
- Tk/Tcl graphical shell by Clay Budin (limited functionality).

Thanks to Lone Aagesen, Robert Asher, Cyrille D'Haese, Julian Faivovich, Gonzalo Giribet, Taran Grant, Dan Janies, Diego Pol, Martin Ramirez, Bob Schelly, and Leo Smith for trying out various test versions, to Kurt Pickett for MacosX compilation, and to Steve Thurston for designing the GUI icons. Lauren Aaronson is part of the POY project as a programmer since version 3.0.7.

enter 'poy' for an overview of help commands

Contents

[Top]

Introduction

The structure of POY

Input files and formats

Output and output files

Setting up a parallel run

Lists of commands by functional groups

Command descriptions

References

INTRODUCTION

[Top] [Contents]

POY is a program for phylogenetic analysis of sequence and other data that implements a number of heuristic procedures to search for the tree or trees that have a minimum edit cost for the given data. For classic morphological data, the edit cost of character on a tree is the length of the character on the tree (see Farris 1970 and Fitch 1971). For optimization of unaligned sequence data on a tree, the basic algorithm to determine the edit cost is due to Sankoff (1975). This algorithm, however, is too computationally intensive to be of practical value, and it is known that the problem is NP−complete (Wang &Jiang 1994). The problem of finding, among all possible trees, those trees with a minimal edit cost for classic characters is in itself already NP−complete, so the problem that POY adresses consists of one NP−complete problem nested within another NP−complete problem.

POY implements classic heuristic tree search strategies like branch swapping, treedrifting, treefusing, and ratcheting (see Goloboff 1999 and Nixon 1999) and heuristic procedures to optimize sequences on trees such as Direct Optimization (Wheeler 1996) and Fixed States (Wheeler 1999). These heuristics for calculating minimal edit costs for sequences are tightly integrated with the tree search heuristics.

As a byproduct of sequence optimization, POY can also output tree−dependent multiple alignments of input sequences.

PROGRAM STRUCTURE

[Top] [Contents]

The high level structure of POY is presented in the figure below. A distinction is made between building trees and refining trees. Refinement of trees consists of heuristic procedures to find better trees on the basis of pre−existing trees that contain all terminals. Available procedures in poy are branch swapping using spr and tbr, tree drifting, ratcheting, and tree fusing. Depending on the origin of the trees that are presented as input to these algorithms, a distinction is made between input tree refinement, replicate tree refinement, and final tree refinement. Input tree refinment refines input trees from the −topology and/or −topofile commands that are specified on the comandline. The resulting trees are then set aside for final refinement later on (note that no commands exist yet that operate in the 'input tree refinement' box; such commands will be available in version 3.1). Replicate tree refinement during any replicate is based on the trees that are first constructed from scratch during the build step of that replicate. For each replicate, the trees that result from the replicate refinement are set aside for final refinement after all replicates are done. Final refinement, finally, starts on the basis of the trees that came out of input tree refinement and all replicate refinements.

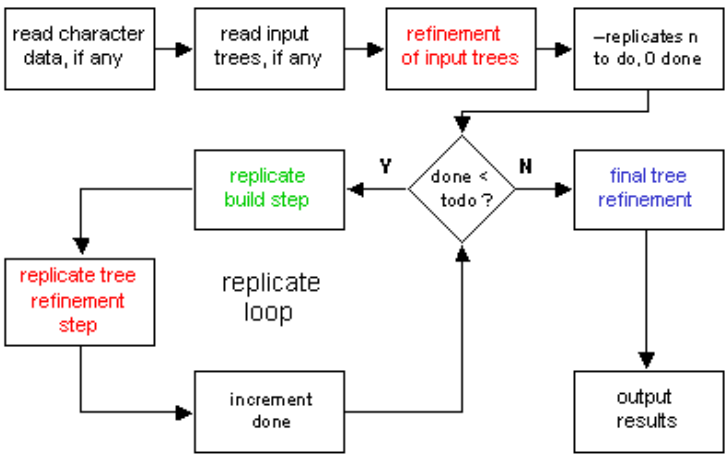

Fig. Structure of POY.

In the most simple case, the build step of a single replicate consists of building a tree by means of stepwise addition, but several commands exist to make the buildstep more elaborate. One of these (−buildsperreplicate n, n>0) consists of doing several addition sequences within a single replicate (by default, the first addition sequence of the first replicate is 'as is', all the other addition sequences use a pseudorandom sequence of terminals). With more than one addition sequence per replicate, the best trees from each addition sequence (and not just the best trees over all addition sequences) are set aside for replicate refinement after all addition sequences for that replicate are done.

Consider the simple command line

```
poy seqdata1 seqdata2 morphdata
```

POY starts out by reading the character data in files seqdata1, seqdata2, and morphdata. Next it performs 1 replicate (default value −replicates 1), using #n 'as is' addition sequence in the build step (default −oneasis), and refining the resulting trees by adding a round of spr and tbr (POY performs tbr by first doing all spr rearrangnents, followed by all non−spr tbr rearrangments; whenever poy reports about 'tbr', this refers to the non−spr part of what is usually called tbr). Final refinement is skipped because there is only a single replicate. Throughout all this, sequences are optimized using Wheeler's (1996) procedure of direct optimization, the default heuristic for calculating the cost of sequences on a tree.

This default search strategy can be changed by adding commands to overrule the various default settings. E.g., the commandline

```
poy seqdata1 −fixedstates seqdata2 morphdata
    −replicates 10 −nooneasis −buildsperreplicate 3 −spr −notbr
```

uses the same data as before but now it applies fixed state optimizaton (Wheeler 1999) to the sequences of file seqdata2. Seqdata1 is still optimized using the default of direct optimization. Using these settings, 10 replicates are performed instead of 1. In each replicate, the build step consists of 3 pseudorandom addition sequences; the trees that are passed on to replicate refinement (now only spr) are the best trees of each addition sequence (not just those of the best addition sequence). The results of all 10 replicates (i.e., not just the best of all those trees) are finally passed to final refinement, consisting of spr + tbr (default; the spr uses a slightly different heuristic compared to spr in replicate refinement; see −slop and −checkslop).

[a renewed system to associate the various refinement strategies to the three phases of refinement is in the making for 3.1]

INPUT FILES AND INPUT FILE FORMATS

[Top] [Contents]

1. List of terminals to use when parsing character data or tree files:
   see command −terminalsfile fname
   (enter 'poy −ll terminalsfile')
2. Character data (use filename as a command line parameter;
   see command −editnames for restrictions on names of terminals):

   - ♦ numerically coded aligned data (classic morphological dataset)

     - ◊ 'Hennig86 format'
       see command −printccode for a description
   - ♦ IUPAC−code coded unaligned nucleotide sequence data

     - ◊ 'fasta format'
       see command −fastashortname for a description
     - ◊ 'genbank format'
       see command −fastashortname for a description
3. Trees: see command −topofile fname
4. Base substitution and gap costs: see command −molecularmatrix fname

5. List of additional sequences for search−based sequence optimization:
   see command −newstates fname
6. List of commands to use: see command −commandfile fname
7. User−defined base frequencies: see command −basefreq fname
   User−defined instantaneous rates: see command −qmatrix fname

OUTPUT AND OUTPUT FILES

[Top] [Contents]

By default POY sends output to the general output streams stdout (standard output) and stderr (standard error), which by default typically both point to the screen. During a run, warnings, progress information, and other kinds of feedback to users are sent to stderr; results are sen# to stdout.

Default output to stdout when all arguments are datafiles (e.g. 'poy mydata1 mydat2') consists of the best trees and their costs that are found using the default search strategy (building a tree by means of stepwise addition using −oneasis, followed by spr and tbr, and using direct optimization as alignment heuristic); these trees are output two times: once with zero−length branches collapsed, and once with zero−length branches resolved ('binary resolution') as to be able to rediagnose the trees and obtain the same cost.

To modify output, three classes of commands exist:

1. request additional output to stdout or stderr
2. request additional output that is sent directly to a file
3. modify the behavior of any of the above commands

Stdout and stderr can be redirected to files using the redirection facilities of the shell in which poy is run. In unix−linux−gnu shells like Bash, Bourne, and Korn, and in Windows 2000 and Windows XP shells,

```
> fname1
```

redirects stdout to file fname1, and

```
2> fname2
```

redirects stderr to file fname2. For example, using

```
poy testdata >results 2>log
```

the results (best trees and their costs) will be in file 'results'; progress information and other feedback for users in file 'log'. Other shells use different conventions. E.g., in csh and Tcsh,

```
>&fname
```

redirects both stderr and stdout to file 'fname'.

Output to files by default overwrites existing files (but see commands −overwriteprotection and

−overwritedataprotection). If the user has no write permission for these files or the directories that they are in, the program will exit with an error message.

SETUP OF A PARALLEL RUN

[Top] [Contents]

POY's parallel processing functionality is implemented using the PVM (Parallel Virtual Machine) library, so PVM has to be installed before POY can be run in parallel (see http://www.epm.ornl.gov/pvm/pvm_home.html for more information). Parallel processing in POY can be used to simultaneously use multiple processors within any single computer and/or to run POY on clusters of computers (for the latter, see, e.g., http://www.beowulf.org).

POY uses different commands to set up a parallel run in a single node PVM environment (one computer that may have multiple processors) and in an a multiple node environment (a cluster of at least two computers, each of which may have multiple processors). In both cases, parallel processing with the default setup is invoked by adding the command −parallel to the command line (or commandfile).

Single node parallel processing

When the PVM configuration has only one entry, the default setup is to spawn four slave processes, no controllers, and no reserves. Other numbers can be requested using the commands −solospawn n, −controllers n, and −dpmsolospawn n. For example,

```
poy −parallel data −solospawn 2 −replicates 100
```

would be useful on a dual processor machine. In addition to the master task (the copy of POY that is started from the commandline), two additional copies are spawned that the master task will use to perform most of the calculations. The master task itself will mainly do administrative work. As a result, each processor will in the end on average have performed half of the requested replicates (POY sends out the next part of the calculations to perform to whichever slave has indicated that it is free to receive work; the level below this − which s#ave tasks are run at which times on which processor − is left completely to the OS). Note that

```
poy −parallel data −solospawn 1 −replicates 100
```

will not make efficient use of a dual processor machine. There will be two tasks (master plus one slave) but the master will most of the time just be waiting for results from the single slave. Because of overhead in sending messages between master and slave, this will in the end even be less efficient than

```
poy data −replicates 100
```

even if this sequential run uses only one of the two processors. Spawning controllers and reserves mostly makes little sense in a single node environment.

Multiple node parallel processing

With more than one entry in the pvm configuration, the default parallel setup is to spawn as many regular slave tasks as there are additional nodes in the PVM configuration (apart from the master node), no controllers, and no reserve tasks. The regular slave tasks are spawned on nodes other than the master node (the details are left to pvm). The number of regular slave processes can be changed using the commands −jobspernode, −numslaveprocesses, −onan, and −onannum. If −onan is on (off by default), POY will spawn in addition −onannum n regular slave tasks on the master node. Command −jobspernode n (default 1) sets how many slave tasks each non−master node will have on average (the distribution over all slave nodes is left to PVM). If −numslaveprocesses n is specified, the total number of regular slave tasks to be spawned on non−master nodes is set to that value (−numslaveprocesses takes precedence over −jobspernode).

The number of controllers is set using −controllers.

The number of reserve tasks is set using −dpmjobspernode and −dpmnumslaveprocesses, two commands that work as −jobspernode and −numslaveprocesses. If −onan is on, POY will in addition spawn −dpmonannum n reserves on the master node.

Note: the parallel code has been tested on gnu−linux systems only up till now (no Windows, no MacOS).

COMMANDS GROUPED BY FUNCTIONS

[Top] [Contents]

not placed

```
−getnewmatrix
 −nogetnewmatrix
−noncodinggap
−recode
 −norecode
```

Diagnostics

```
−characterweights
 −nocharacterweights
−diagnose
 −nodiagnose
−iafiles
 −noiafiles
−impliedalignment
 −noimpliedalignment
−indices
 −noindices
−pairmatrix
 −nopairmatrix
−printhypanc
 −noprinthypanc
−printlotshypanc
 −noprintlotshypanc
−printqmat
 −noprintqmat
```

-stats
 -nostats

Environment settings

--
-catchslaveoutput
-commandfiledir
-datadir
-enabletmpfiles
 -noenabletmpfiles
-gc
 -nogc
-nopoyini
-overwritedataprotection
-overwriteprotection

Help commands

--help
-about
-cat cmdgroups
-cat commandbrowsing
-cat faulttolerance
-cat helptopics
-cat infiles
-cat introduction
-cat_outfiles
-cat programflow
-cat references
-cat setupparallel
-clist
-cllist
-dlist
-dllist
-help
-helpdumpfile
-helpfile
-helpwidth
-list
-llist

Input files

-basefreq
-commandfile
-molecularmatrix
-newstates
-qmatrix
-terminalsfile
-topofile

Input file modifiers

-deletegapsfrominput
 -nodeletegapsfrominput
-editnames
-fastashortname
 -nofastashortname
-minterminals
-polyaddconverttorange
 -nopolyaddconverttorange
-topopwpickrandom
-topopickrandom

Input trees

-topopwpickrandom
-topopickrandom
-topodiagnoseonly
 -notopodiagnoseonly
-topofile
-topolist
 -notopolist
-topology
-topooutgroup
-toposkipidentical
 -notoposkipidentical

Optimality criterion

-defaultweight
-weight

Optimality criterion, sequence optimization, cost matrix

-change
-extensiongap
-gap
-leading
 -noleading
-molecularmatrix
-trailinggap

Optimality criterion, sequence optimization, ML

-basefreq
-estimatep
 -noestimatep
-estimateparamsfirst
 -noestimateparamsfirst
-estimateq
 -noestimateq
-freqmodel
-gammaalpha
-gammaclasses
-invariantsitesadjust
 -noinvariantsitesadjust

-likelihood
 -nolikelihood
-likelihoodconvergancevalue
-likelihoodestimationsize
-likelihoodesttranseachtime
 -nolikelihoodesttranseachtime
-likelihoodextensiongap
-likelihoodmaxnumiterations
-likelihoodnoncodinggap
-likelihoodroundingmultiplier
-likelihoodstep
-likelihoodstepinterval
-likelihoodtrailinggap
-qmatrix
-submodel
-theta
-totallikelihood
 -nototallikelihood
-trullytotallikelihood
 -notrullytotallikelihood

Optimality criterion, single column data

-goloboff
-kfactor
-multiplier
-polyaddconverttorange
 -nopolyaddconverttorange

Outgroups

-jackoutgroup
-outgroup
-plotoutgroup
-randomizeoutgroup
 -norandomizeoutgroup
-rerootafterbuild
 -norerootafterbuild
-topooutgroup

Output files

-helpdumpfile
-helpfile
-iafiles
 -noiafiles
-jackftree
-jackfpseudotrees
-jackfpseudoconsensustrees
-jackwincladfile
-phastwincladfile
-plotfile
-poybintreefile
-poystrictconsensuscharfile
-poystrictconsensustreefile

-poytreefile
-printhypanc
 -noprinthypanc
-printlotshypanc
 -noprintlotshypanc

Output modifiers

-hypancfile
-hypancname
 -nohypancname
-plotechocommandline
-plotencoding
-plotfile
-plotfrequencies
-plotmajority
-plotoutgroup
-plotstrict
-plottrees
-plotwidth
-verbose
 -noverbose

Output to stdout/stderr

-characterweights
 -nocharacterweights
-diagnose
 -nodiagnose
-impliedalignment
 -noimpliedalignment
-indices
 -noindices
-intermediate
 -nointermediate
-jackboot
 -nojackboot
-jacktree
 -nojacktree
-jackpseudotrees
 -nojackpseudotrees
-jackpseudoconsensustrees
 -nojackpseudoconsensustrees
-pairmatrix
 -nopairmatrix
-printccode
 -noprintccode
-printqmat
 -noprintqmat
-printtree
 -noprinttree
-quote
-repintermediate
 -norepintermediate
-showiterative
 -noshowiterative
-spewbinary

-nospewbinary
-stats
-nostats
-time
-notime
-verbose
-noverbose

Parallel processing

-crashslave
-crashcontroller
-crashreserve
-onversionconflict
-parallel
-noparallel

Parallel processing, dynamic process migration

-dpm
-nodpm
-dpmacceptratio
-dpmautoadjustperiod
-nodpmautoadjustperiod
-dpmautoperiod
-dpmjobspernode
-dpmmaxperiod
-dpmmaxprocessors
-dpmminperiod
-dpmnumslaveprocesses
-dpmonannum
-dpmperiod
-dpmproblemsize
-dpmsolospawn

Parallel processing, granularity of parallellism

-multibuild
-nomultibuild
-multidrift
-nomultidrift
-multirandom
-nomultirandom
-multiratchet
-nomultiratchet

Parallel processing, setup

-controllers
-dpmjobspernode
-dpmnumslaveprocesses
-dpmonannum
-dpmsolospawn
-jobspernode

-numslaveprocesses
-onan
 -noonan
-onannum
-randomizeslaves
 -norandomizeslaves
-solospawn

Sequence optimization, heuristics, direct optimization

-discrepancies
 -nodiscrepancies
-dogaptie
-exact
 -noexact
-prealigned
 -noprealigned#

Sequence optimization, heuristics, direct optimization, iterativepass

-iterativeinitfinal
 -noiterativeinitfinal
-iterativeinitsingle
 -noiterativeinitsingle
-iterativekeepbetter
 -noiterativekeepbetter
-iterativelowmem
 -noiterativelowmem
-iterativepass
 -noiterativepass
-iterativepassfinal
 -noiterativepassfinal
-iterativerandom
 -noiterativerandom
-lowmemthreshold
-maxiterations
-showiterative
 -noshowiterative

Sequence optimization, heuristics, static approximation

-staticapprox
 -nostaticapprox
-staticapproxbuild
 -nostaticapproxbuild

Sequence optimization, heuristics, fixed states

-fixedstates
 -nofixedstates

Sequence optimization, heuristics, search-based optimization

-compressstates
 -nocompressstates
-newstates
-printhypanc
 -noprinthypanc
-printlotshypanc
 -noprintlotshypanc

Support

-bremer
 -nobremer
-bremerspr
 -nobremerspr
-jackboot
 -nojackboot
-jackfrequencies
-jackoutgroup
-jackftree
-jackfpseudotrees
-jackfpseudoconsensustrees
-jacktree
 -nojacktree
-jackpseudotrees
 -nojackpseudotrees
-jackpseudoconsensustrees
 -nojackpseudoconsensustrees
-jackwincladfile

Tree buffer

-buildmaxtrees
-fitchtrees
 -nofitchtrees
-holdmaxtrees
-maxtrees
-sprmaxtrees
-tbrmaxtrees

Tree search heuristics, general

-cutswap
 -nocutswap
-incremental
 -noincremental
-jackstart
 -nojackstart
-randomizeoutgroup
 -norandomizeoutgroup
-replicates
-rerootafterbuild
 -norerootafterbuild
-slop
-upincremental
 -noupincremental

Tree search heuristics, build step of a replicate

-approxbuild
 -noapproxbuild
-approxquickspr
 -noapproxquickspr
-approxquicktbr
 -noapproxquicktbr
-buildmaxtrees
-buildmaxtrees
-buildslop
-buildsperreplicate
-buildspr
 -nobuildspr
-buildtbr
 -nobuildtbr
-oneasis
 -nooneasis
-seed
-staticapproxbuild
 -nostaticapproxbuild

Tree search heuristics, constrained searches

-agree
-constrain
 -noconstrain
-disagree
-dropconstraints
 -nodropconstraints

Tree search heuristics, refinement, general

-checkslop
-minstop
-quick
 -noquick
-staticapprox
 -nostaticapprox
-stopat

Tree search heuristics, refinement, drifting

-approxdrift
 -noapproxdrift
-driftequallaccept
-driftlengthbase
#a href="#drsp">-driftspr
 -nodriftspr
-drifttbr
 -nodrifttbr
-numdriftchanges
-numdriftspr

-numdrifttbr

Tree search heuristics, refinement, fusing

-fuseafterreplicates
 -nofuseafterreplicates
-fuselimit
-fusemaxtrees
-fusemingroup
-fusewhilereplicates
 -nofusewhilereplicates
-fusingrounds
-treefuse
 -notreefuse
-treefusespr
 -notreefusespr
-treefusetbr
 -notreefusetbr

Tree search heuristics, refinement, ratcheting

-checkfrequency
-ratchetoverpercent
-ratchetpercent
-ratchetseverity
-ratchetslop
-ratchetspr
-ratchettbr
-ratchettrees

Tree search heuristics, refinement, spr

-spr
 -nospr
-sprmaxtrees

Tree search heuristics, refinement, tbr

-tbr
 -notbr
-tbrmaxtrees

Deprecated commands

-build
 -nobuild
-dpmmaxprocessors
-maxprocessors
-random

COMMAND DESCRIPTIONS

[Top] [Contents]

The descriptions for the commands that relate to maximum likelihood and iterative pass are from or based on Janies &Wheeler 2002.
Other descriptions from or based on original descriptions from Janies and Wheeler are as indicated below.

Commands require arguments as indicated on the first line of each description. Defaults, if they exist, are indicated between square brackets on the first line (these defaults are active when the command is not entered). Commands that require no arguments are toggles and come in −xyz / −noxyz pairs (with the exception of the −agree/−disagree pair, −nopoyini, −help, −−help, and the −catxyz commands). All commands start with a hyphen. Character data files (see Input and input files) cannot start with a hyphen and must be entered directly on the command line (i.e., not as an argument to a command).

−−
Short for −commandfile

−−help
Print (stderr, but see −helpfile) an overview of commands for getting more information.
Same as −help.

−about
Print (stderr) general information about the program

−agree
When performing a search, only keep trees that agree with the groups that are specified using the −constrain command. This is the default way in which constraints are used. See −disagree for an alternative.

see −poystrictconsensuscharfile −bremer

−approxbuild [noapproxbuild]
Skip a correction for a shortcut in cost calculations during the build step of replicates. In general it is not so efficient to spend too much time building starting trees for swapping, so −approxbuild may mostly be the best choice.

−approxdrift [approxdrift]
Skip a correction for a shortcut in cost calculations during tree rearrangements while drifting.

−approxquickspr [approxquickspr]
Skip a correction for a shortcut in cost calculations while doing spr in input tree and replicate refinement.

−approxquicktbr [approxquicktbr]
Skip a correction for a shortcut in cost calculations while doing tbr in input tree and replicate refinment.

−basefreq filename
Read base and indel frequencies from a file, for use under −likelihood. The order is A C G T GAP and the file must either contain just these five numbers on the first line, or start like that, followed by a semicolon (if the semicolon is present, the file can have additional lines that can be used for comments). If −basefreq is not specified, POY will estimate these frequencies according to the setting of −estimatep.

−bremer [nobremer]
Estimate Bremer support values based on a TBR search. The command requires a constraint file (see −constrain, −poystrictconsensuscharfile) and estimates support for the groups that are in this file.

Compared to calculating Bremer support by consensing ever more inclusive sets of suboptimal trees, this approach is faster but it may overestimate group support.

When the TBR search results in trees that are shorter than any tree that satisfies all constraints, POY will report negative support values for some or all groups of the constraint file. If the constraint file was derived from the strict consensus of the best trees found thus far, such negative support values indicate inadequacy of the tree search strategy that was employed thus far.

see also −agree −disagree

−bremerspr [nobremerspr]
As −bremer, but using SPR instead of TBR.

−build
Deprecated; use −notopodiagnoseonly instead.

−buildmaxtrees n [the value of −maxtrees n]
Set maximum number of trees held during the build step of a replicate.

see −maxtrees −fitchtrees

−buildslop n [the value of −slop n]
Set the slop value that is used in the build step of a replicate.

−buildsperreplicate n [1]
Do n addition sequences in the build step of any single replicate of −replicates n. If −oneasis is on (default), then the first addition sequence of the first replicate is as found in the first datafile (or concatenated −terminalsfiles). All other addition sequences are pseudorandom (cf −seed). Under nooneasis, all addition sequences are pseudorandom./nThe best tree(s) of each addition sequence are

subsequently passed to replicate refimnement.

−buildspr [nobuildspr]
Append a single tree SPR search after each addition sequence during the build step of a replicate. Mostly makes sense only in combination with −buildsperreplicates n, n > 1

−buildtbr [nobuildtbr]
Append a single tree TBR search after each addirion sequence during the build step of a replicate. Mostly makes sense only in combination with −buildsperreplicate n, n > 1.

−cat_cmdgroups
Print (stderr, but see −helpfile) a list of commands grouped according to function

−cat_commandbrowsing
Print (stderr, but see −helpfile) a list of commands to browse command documentations.

−cat_helptopics #
Print (stderr, but see −helpfile) a list of specific help topics.

−cat_infiles
Print (stderr, but see −helpfile) information about acepted formats for inputfiles.

−cat_introduction
Print (stderr, but see −helpfile) information about the kind of analyses that poy does?

−cat_outfiles
Print (stderr, but see −helpfile) an overview of output and output files

−cat_programflow
Print (stderr, but see −helpfile) information about the structure of the program; in combination with −helpfileEFER also produces a flowchart (png format)

−cat_references
Print (stderr, but see −helpfile) a list of literature references that are used in command line help.

−cat_setupparallel
Print (stderr, but see −helpfile) information about setting up a parallel run.

−catchslaveoutput [nocatchslaveoutput]
Under −parallel, redirect all stderr/stout output from slave tasks to stderr/stdout of the master task.

−change n [1]
Set base substitution cost. Use −moleculermatrix to specify different costs for different substitutions.

−characterweights [nocharacterweights]
Output (stdout) character information (length, minimum length, maximum length, weight, ci, ri) for all characters and trees in the analysis Characters are single columns for xread formatted data, sequences or fragments for other character data (see input file section); they are numbered in the order as encountered when parsing the commandline.

−checkfrequency n [0]
Set frequency of message checking in parallel ratcheting (−multiratchet only) (Janies &Wheeler 2002).

−checkslop n [the value of −slop n]
Slop value for tbr branch swapping in final tree refinement. Setting −checkslop values high compared to the other slop values increases the relative effort to find shorter trees to final refinement. This is often an efficient search strategy.

−clist xyz
Print (stderr) a list of all commands that have documentation that contains commands that contain the string xyz.

see −help −cat_commandbrowsing −cat_helptopics

−cllist xyz
Print (stderr) a list of all commands that have documentation that contains commands that contain the string xyz, and include the documentation of these commands in this list.

see −help −cat_commandbrowsing −cat_helptopics

−commandfile fname
In the command line, substitute the content of file fname for itself. E.g., with file PLOTSTRICT containing

```
'−printtree −plottrees off −plotmajority off −plotstrict on −plotfile stdout'
```

the command line

```
poy mydata −commandfile PLOTSTRICT −replicates 5
```

amounts to

```
poy mydata −printtree −plottrees off −plotmajority off
     −plotstrict on −plotfile stdout −replicates 5
```

A single run can have multiple −commandfile commands, and the command files themselves may have embedded −commandfile commands. Nesting is unlimited as long#as there is no circular recursion.

As other input files, command files must be plain text files. There are no further restrictions on format (there can be multiple lines with multiple commands each).

"−−" (double dash) can be used as shorthand for −commandfile, so

```
poy mydata −− PLOTSTRICT −replicates 5
poy mydata −commandfile PLOTSTRICT −replicates 5
```

are the same (note the space between "−−" and "PLOTSTRICT").

POY checks if there is a file in the current directory that is called "poy.ini". If that file exists, it is automatically treated as a commandfile, and its contents come at the beginning of the command line (automatic expansion of poy.ini can be turned off with −nopoyini). As an example, with file poy.ini containing

```
−parallel
```

commandlines

```
poy mydata −− PLOTSTRICT −replicates 5
poy mydata −− PLOTSTRICT −replicates 5 −− poy.ini −nopoyini
```

will expand to

```
poy mydata −parallel −printtree −plottrees off −plotmajority off
     −plotstrict on −plotfile stdout −replicates 5
```

Command line

```
poy mydata −− PLOTSTRICT −replicates 5 −nopoyini
```

will expand to

```
poy mydata −printtree −plottrees off −plotmajority off
     −plotstrict on −plotfile stdout −replicates 5
```

To keep track of commands that are actually in use, both the original and the fully expanded command line are output to stderr at the start of a run.

Besides expanding command files, poy also expands command abbreviations, provided these are unambiguous. As an example,

```
poy −randomizeout −poly
```

is turned automatically into

```
poy −randomizeoutgroups −polyaddconverttorange
```

after which poy will tell you that there are no data or tree files, and exit.

If there is no unique match, you get a list of possibilities + a warning that there is an ambiguous abbreviation. As an example,

```
poy −rando
```

will list

```
−randomizeoutgroup −randomizeslaves
```

output the ambiguous abbreviation and exit.

Command expansion does NOT apply to arguments of commands.

see also −commandfiledir, −−

−commandfiledir dirname [current directory]
Set the directory where POY looks for commandfilenames (see −commandfile fname). There can be multiple −commandfiledir commands, each next one determining the directory for command files in the rest of the command line. The command line is scanned from left to right.

The default commandfile poy.ini is expanded at the start of the command line and is therefore not affected by −commandfiledir. So it must always be in the current directory.

As an example if directory /home/me/myshorthands contains a file PWC with contents

```
"−phastwincladfile"
```

and a file IA that contains

```
"−impliedalignment"
```

and the current directory contains a file IA_TO_FILE with contents

```
"− commandfiledir /home/me/myshorthands/ −− IA −− PWC"
```

then

```
poy mydata −− IA_TO_FILE myimpliedalignment
```

will expand to

```
poy mydata −impliedalignment −phastwincladfile myimpliedalignment
```

which, on the basis of the best trees that are found using the default tree search strategy, will calculate an implied alignment and output it to file myimpliedalignment.

When dealing with a commandfile directory and a commandfile, the program does not check, fttb, if the directory is a valid directory or if the file is a valid file in that directory. It only checks if the literal concatenation of the two is a valid file. This is why "/home/me/myshorthands" needs the "/" at the end. Without it, POY would notify you that file "/home/me/myshorthandsmycommand" d#es not exist and then exit. It follows that

```
poy mydata −commandfiledir IA_T −commandfile O_FILE myimpliedalignment
```

though hardly advisable, would work equally well. To keep track of what is going on, both the original and the the fully expanded command line are printed to stderr at the start of a run.

A related command is −datadir, which is used to set the directory of input files other than command files.

see also −− −datadir

−compressstates [nocompressstates]
During search−based optimization (see −newstates), keep track of the states used during the build stage and remove unused states from the state set (as initialized by −newstates filename) during swapping. The state set is returned to its original size for a final round of swapping if −checkslop n is specified. (Janies &Wheeler 2002)

−constrain [noconstrain]
Toggle to interpret all following xread−formatted character data files (see −printccode) as files that contain constraints for tree searches. Default usage of constraints (−agree) is to to conform to the group inclusion characters as specified in the files that contain constraints. These datafiles must have binary characters only, and these are interpreted as group inclusion charcters.

see also −disagree −dropconstraints −poystrictconsensuscharfile −jackcharfile

−controllers n [0]
Number of subclusters in parallel search. With ns the number of slave nodes, these ns nodes are divided into n groups of ns div n; the remaining ns mod n slaves are assigned to the first ns mod n subclusters.

E.g., in a cluster that consists of 60 slave nodes plus one master node

```
poy −parallel controllers 3 −multirandom −jobspernode 1 −replicates 30
```

sets up three subclusters (of each 20 nodes), each of which is controlled by a controller task. The controller tasks are spawned on three random machines in the cluster. As they do not perform a lot of calculations themselves, they contribute little to the overall workload of these nodes.

Using this setup, POY then performs the 30 randomized searches that are requested as follows: initially each of the subclusters is requested to do one complete randomized search. The controllers do this by parallelizing this task in turn over the 20 slave nodes that they command. Poy then waits until it gets results back from any of the three controllers, and as soon as it receives such results it sends a request for the next search to that same controller from which it has gotten results. This repeats until the 30

searches have been sent out.

While waiting for results, POY performs various tasks such as filtering out duplicate trees that were found in different subclusters, or notifying the two other subclusters of a new current best length in case it receives trees of a new shortest length from the third subcluster. If it has nothing else to do, it can sometimes be caught whistling.

Setting up a cluster of subclusters serves to balance two contradictory considerations, as to achieve optimal overall efficiency of the available computing resources: reducing overhead of parallelism versus using as many nodes as are available.

Using a variety of empirical datasets, Janies &Wheeler (2001) found good speedups (little parallel overhead) for various parallel algorithms in POY for cluster sizes of up to 16−32 nodes. Beyond that, overhead sometimes increases dramatically.

In this example, any single of the 30 searches that have been requested would be performed most efficiently on a single node (then there would be no parallel overhead within the search). Howeve#, as there are 60 nodes and only 30 searches were requested, half of the nodes would sit there idle until the other half would have completed their search. However efficiently they manage to do that, overall efficiency in the cluster would never exceed 50 %.

By breaking up a large cluster into subclusters of 16 to 32 nodes (benchmarking is advised), POY manages to keep its parallel algorithms in the region where good speedups can be expected, while at the same time it manages to keep everyone busy. This trade off should result in optimal overall efficiency.

Note that controllers are only spawned when −multirandom is on (otherwise they would be spawned but not used). If more controllers are requested than regular slave tasks, the number of controllers is adjusted to the number of regular slave tasks. A single controller is never spawned.

see also −parallel −dpm

−crashcontroller n
Under −parallel and with controllers n, n > 1, make controller jobs exit in the midst of calculations. The lower n, the more frequent controllers will crash. Just for fun.

−crashreserve n
Under −parallel and −dpm, make reserve jobs exit in the midst of calculations. The lower n, the more frequent reserve jobs will crash. Just for fun.

−crashslave n
Under −parallel, make regular slave jobs exit in the midst of calculations. The lower n, the more frequent slaves will crash. Just for fun.

−cutswap [cutswap]
Use a shortcut for calculating the length of a tree after swapping a branch while doing ree refinement (cf. Goloboff 1996a) for regular characters.

see −discrepancies

−datadir dirname [current drectory]
Set the directory where POY looks for input files other than command files. These can be files with actual character data or files with other information (see e.g. −molecularmatrix). To set the directory for command files, use −commandfiledir.

As an example, running POY from /home/me/workdir and with data files its1a, its1b, and its2a, its 2b, and its2c in directory /home/me/gentians/Lisianthius/its

```
poy −datadir /home/me/gentians/Lisianthius its1a its1b its2a its2b
     its2c −replicates 5 −datadir ./ −molecularmatrix 111
```

will expand to

```
poy −datdir /home/me/gentians/Lisianthius /home/me/gentians/Lisianthius/its1a
   /home/me/gentians/Lisianthius/its1b /home/me/gentians/Lisianthius/its2a
   /home/me/gentians/Lisianthius/its2b /home/me/gentians/Lisianthius/its2c
   −replicates 5 −datadir ./ −molecularmatrix ./111
```

This reduces typing effort or useless/confusing duplication of data files into the working directory. To keep track of everything, the original and fully expanded command line is output to stderr at the start of a run.

see also −commandfiledir

−defaultweight n [1]
Set the weight (see −weight) of all following character data files to n. This can be used with an argument of −1 to search for the worst possible tree for a dataset. This maximum tree length is necessary for rescaled ILD (Wheeler &Hayashi 1997) and meta ILD calculations (Wheeler et al., *in prep*). (Janies &Wheeler 2002)

−deletegapsfrominput n [20]
See −fastashortname

−diagnose [nodiagnose]
At the end of a run, output (stdout) branch lengths, c#aracter changes, and inner node state assignments for all trees.

−disagree
Use with in combination with −constrain to find shortest trees subject to the constraint that no groups that are specified in the constraints may be present. This overrules the default treatment of constraints (see −agree). One application is for Bremer support tests (employed in Frost et al. 2001) on a group by group basis. This is an alternative to the −bremer command that produces values for all groups simultaneously.

−discrepancies [discrepancies]

Print (stderr) difference between cladogram length as found from shortcuts and a complete down pass. (Janies &Wheeler 2002).

−dlist xyz
Print (stderr) a list of all commands that have documentation that contains the string xyz.

see −help −cat_commandbrowsing −cat_helptopics

−dllist xyz
Print (stderr) a list of all commands that have documentation that contains the string xyz, and include this documentation.

see −help −cat_commandbrowsing −cat_helptopics

−dogaptie "shorter", "longer", or "nopreference" [shorter]
When doing a pairwise alignment between two descendant sequences of a node on a tree (and calculating an intermediate sequence) during direct optimization or optimization alignment, it is possible that at certain cells in the dynamic programing table a tie exists between inserting a gap in the first or the second sequence; depending on how the tie is resolved, a different intermediate sequence may be calculated, which ultimately may affect the heuristically calculated cost of the tree (and assessment of zero−length branches). With '−dogaptie shorter' (default), the gap will in such cases always be inserted in the shorter sequence; with '−dogaptie longer' in the longer; with '−dogaptie nopreference' poy arbitrarily resolves the tie in a way that depends on the order of the descendant sequences. In versions older than 3.0.8. this command was not available and ties were resolved using 'nopreference'. This command was introduced to make the direct optimization cost calculation of a tree independent of the order of siblings in a tree: using either 'shorter' or 'longer', the calculated cost will not depend on sibling order (e.g. inputtrees (a (b (c d))) and (a ((c d) b)) are guaranteed to give the same cost; note that there is still dependence on the root: e.g. ((c d) (a b)) might give a different cost; also note that this cost might be different for 'shorter' versus 'longer').
When rediagnosing trees from a previous run (see −topofile and −topology), the same setting of this command has to be used in both runs and the trees must have the same basal split to be 100 percent sure to find the same cost as before (for 'nopreference', the trees must in addition preserve the left/right order of siblings). The 'nopreference' setting is just kept for backward compatibility.
[When the two sequences are equally long, they are compared numerically to determine which is the shorter/longer (base 'a' bitcoded as 1, 'c' as 10, 'g' as 100, and 't' as 1000; for polymorphisms the bitcodes are added)].
This setting also affects −iterativepass: with 'nopreference', the behavior depends on order of siblings (as in previous versions); with the two other settings, this dependence on sibling order is removed (in two different ways)

−dpm [nodpm]
Under −parallel, use Dynamic Process Migration to move processes form relati#ely over−utilized processors to relatively less active processors. This is likely to improve load balancing in the cluster. It may improve speed of calculations significantly when different POY scripts run simultaneously or when the cluster consists of processors with significantly different speeds.

see −dpmacceptratio −dpmautoadjustperiod −dpmmaxperiod −dpmnumslaveprocesses −dpmjobspernode −dpmminperiod −dpmperiod −dpmproblemsize

−dpmacceptratio n [1.5]
Threshold in the ratio of instantaneous processor performance that must be reached to trigger a task migration under dynamic process migration (−dpm). The performance is a combined measure of processor load and intrinsic processor speed.

When the master node or any of the controllers need a slave task to perform some calculations, POY calculates the performance of the first available regular task and the performance of all current reserve tasks. If the ratio of the performances of the best reserve task and that regular task is at least n, the regular job is relegated to the pool of reserve tasks while the best reserve job is recruited into the pool of regular jobs. The work to be done is then sent to the new regular job. In practice, these calculations do not occur on every possible occasion but with intervals of −dpmperiod, the 'bid' period.

−dpmautoadjustperiod [nodpmautoadjustperiod]
Under dynamic process migration (−dpm), automatically adjust the bid period to send out requests to the reserve jobs for calculations of performance. If these requests lead to task migration, the period is reduced. If bids are unsuccessful, it is increased.

−dpmautoperiod
Undocumented.

−dpmjobspernode n [1]:
Under dynamic process migration (−dpm), set the average number of reserve processes that will be spawned on PVM nodes other than the master node. See Setup of a parallel run

−dpmmaxperiod n [100]
Under dynamic process migration (−dpm), the maximum length bid period allowed when −dpmautoadjustperiod is on.

−dpmmaxprocessors n
Deprecated. See Setup of a parallel run

−dpmminperiod n [1]
Under dynamic process migration (−dpm), the minimum length bid period allowed when −dpmautoadjustperiod is on.

−dpmnumslaveprocesses n [(number of entries in PVM configuration − 1) times −dpmjobspernode n]
Under −dpm, the maximum number of reserve jobs to spawn on PVM nodes other than the masternode.See Setup of a parallel run

−dpmonannum n [0]
See Setup of a parallel run

−dpmperiod n [10]

Under dynamic process migration (−dpm), the bid period (or starting bid period iwhen −dpmautoadjustperiod is on).

−dpmproblemsize n [5000]
Under dynamic process migration (−dpm), the size of the calculation that is used to assess the performance of a job on a particular processor. The bigger the n, the larger the calculation and the better the assessment of performance.

−dpmsolospawn n [1]
Under −parallel, set the number of reserve tasks to b# spawned in a PVM environment with only a single node. See Setup of a parallel run

−driftequallaccept n [50]
Set the probability (percentagewise) that a candidate tree that has the same cost as the current best tree is acepted during tree drifting (better candidate trees are always accepted; see command −driftlengthbase for worse candidate trees).

see −drifttbr −driftspr

−driftlengthbase n [2]
Set 'n' in the acceptance probability for suboptimal trees during drifting. This acceptance probability is 1 / (n + c − b), with c the cost of the suboptimal candidate tree and b the cost of the best tree thus far.

see −drifttbr −driftspr

−driftspr [nodriftspr]
Perform −ndriftspr n rounds of tree drifting (Goloboff 1999) based on SPR search during replicate refinement (default: 1 round).

see −approxdrift −driftspr −driftequallaccept −driftlengthbase

−drifttbr [nodrifttbr]
Perform −ndrifttbr n rounds of tree drifting (Goloboff 1999) based on TBR search during replicate refinement (default: 1 round).

see −approxdrift −drifttbr −driftequallaccept −driftlengthbase

−dropconstraints [nodropconstraints]
Drop constraints (see −constrain) before going final tree refinement.

−editnames string
By default, the program aborts when it encounters a −terminalsfile or a character data file with a terminal name that contains any of the six special characters that are used in parenthetical notation of trees ('(', ')', '[', ']', '*', and ';'). When −editnames is specified, the program will not abort but replace these special characters by the six characters of the string argument of this command (matched in the order ()[]*;). Note that this substitution does NOT occur in tree topologies as specified using −topofile and −topology.

−enabletmpfiles [noenabletmpfiles]
With temporary files enabled, POY will create and use three temporary files in the current directory. These files have fixed names (poy1.tmp, poy2.tmp, and poy3.tmp), so simultaneously running two poy jobs with −enabletmpfiles on from the same directory is a good recipe to let them both crash.

If such files exist, they are overwritten without warning, irrespective of −overwriteprotection. When running poy with command −parallel, only the master task will create these temporary files.

Currently the temporary files are used for plotting trees or calculating consensus trees when no character data are specified (see −terminalsfile for more information)

see also −topology −topofile

−estimatep [estimatep]
Under −likelihood, estimate base and indel frequencies from all pairwise alignments of input sequences (or fragments) and fix these estimates for the duration of the search. With −noestimatep, base frequencies are estimated for and from pairs of sequences as they are encountered.

−estimateparamsfirst [estimateparamsfirst]
Under −likelihood, estimate likelihood parameters from all pairwise alignments of input sequences (or fragments) and fix these estimates for the duration of the search. When −noestimateparams#irst is invoked, parameters are re−estimated for each pairwise sequence comparison.

−estimateq [noestimateq]
Under −likelihood, estimate transition probabilities from all pairwise alignments of input sequences (or fragments) and fix these estimates for the duration of the search. With −noestimateq, transition probabilities are estimated for and from each pair of sequences as they are encountered.

−exact [noexact]
Use exact tree length calculation based on the heuristic homologies of the down pass. Used to improve tree length calculations whenever slop features find a better or equal length tree during a search. Trades execution speed for accuracy. (Janies &Wheeler 2002)

−extensiongap n [gap cost as set by −gap or −molecularmatrix fname]
Set extension gap cost. Without −extensiongap specified, the cost of a gap of length m is m times the gap cost as specified by −gap or as set by −molecularmatrix. With −extensiongap specified, the gap opening cost is as set by −gap or −molecularmatrix, and any next subsequent contiguous missing base has a cost of −extensiongap. This gives a total cost for the gap equal to (gap cost + (m−1) * extension gap cost.

−fastashortname [fastashortname]
For sequence files in fasta format, use only the first stretch of non−blanks after '>' as taxonname. This allows to enter some annotation on the '>' line. Example: with −fastashortname

```
> taxona some comments on this sequence
aaacgt
aac
> taxonb some comments on this sequence
aaacgt
```

will be read as a sequence aaacgtaac for taxona and a sequence aaacgt for taxonb; with −nofastashortname

```
> taxona some comments on this sequence
aaacgt
aac
> taxonb some comments on this sequence
aaacgt
```

will be read as a sequence aaacgtaac for taxona_some_comments_on_this_sequence and a sequence aaacgt for taxonb_some_comments_on_this_sequence.

The general format of a fasta sequence file:
a line with a taxon name starts with '>' (has to be first character on the line)
every following line (blank lines included) until the next line with a name or until the end of the file is taken as sequence data for the preceding name. So they must consist of valid iupac codes + the characters '*' and '#' (see below).

Compare this to the other format for sequences that poy uses (some constraints are removed now compared to older poys):
entries for taxa are separated by an empty line (i.e. a line that consist of blanks only − so spaces are allowed).
In this format, the first stretch of non−blanks in a file is considered the first terminal name, and all following lines until end−of−file or until an empty line occurs, are read as data for terminal. The first stretch of non−blanks on the first following non−blank line is the name for the second terminal, and so on. In this case, the data can have interspersed integers (cf old BLOCK format), but these integers are NOT read as part of the data.

Sequences can now have the characters '#' and '*' in them. These characters are used to indicate multiple fragments in single files. '#' is the fragment separator, '*' can be used to exclude individual fragments from the analysis.

Example

```
> a
acc#aacgt−t#tttttt#a*t
>b
*#aacgtcc#ttt*tt#
```

is read as involving four fragments for taxa a and b.

```
          frag1     frag2    frag3    frag4
taxon a   acc       aacgt_t  tttttt   at
taxon b   missing   aacgtcc  ttttt    missing
```

Of these, only fragment 2 is used (the other are excluded#with the *)

With −deletgapsfrominput, the second fragment for taxon a would be read as aacgtt.

Possible use of this is the following heuristic search strategy:
output some implied alignments for some decent set of trees, minimally edit these alignments to indicate fragments (possibly eliminating fragments, such as non−homologous stretches at the beginning or end of sequences), and use the result as input for a next round of (constrained) search (the constraint here is in the primary homology statement is that are implied in the fragment delimitation; bypass the constraint by checking unconstrained tree length at the very end (using −topofile, −topodiagnoseonly, (−exact), and data files without #'s // or even do additional swapping)).

Note that # is a fragment separator (as opposed to delimiter):

```
>a
 #aac#aat#
>b
#aat#ac#
```

involves four fragments: the two that are explicitly entered, a third one before the first sharp; and a fourth one following the this sharp. These last two are considered missing in both a nd b.

Each taxon must have the same number of sharps (1 less than the number of fragments), otherwise poy stops with an error message. The stars can appear anywhere in a fragment, in any taxon for that fragment.

see −printccode for formatting requirements for morphological data
see also −terminalsfile −minterminals

−fitchtrees [nofitchtrees]
Ensure that trees kept in buffer (set by −maxtrees) are a random subset of those that would have been kept if the treebuffer would have been larger. This is based on an algorithm suggested by W. Fitch. (Janies &Wheeler 2002)

see −maxtrees

−fixedstates [nofixedstates]
Optimize sequences or fragments using the fixed states procedure of Wheeler (1999). This is a toggle: must be invoked before data file to be considered under fixed states, and all names of datafiles that follow

are optimized using fixed states until the command is turned off using −nofixedstates. E.g., in

```
poy datafile1 −fixedstates datafile2 −seed −1 −nofixedstates datafile3 −replicates 10
```

datafile1 and datafile3 will be analyzed using optimization alignment or direct optimization (Wheeler 1996; default), datafile2 with the fixed states procedure. As a check, POY writes the setting of each datafile to stderr at he start of a run. Also used as part of search−based optimization (see −newstates)

−freqmodel "f5", "f2", or "f1" [f5]
Under −likelihood, number of base frequencies to estimate. f5 allows all base (and indel) frequencies to vary, f2 allows two classes (bases and indels) , and f1 sets all to 0.2.

−fuselimit n [no limit]
When doing treefusing (see −treefuse) at any point in the program, try to fuse at most n pairs of trees. By default (an memory permitting) all possible pairwise fusings are done.

see −treefuse

−fusemaxtrees n [value of −maxtrees n]
Maximum number of trees kept while doing treefusing.

see −treefuse −fitchtrees

−fusemingroup n [5]
Minimum number of taxa in a subtree to be exchanged in tree fusing (Janies &Wheeler 2002).

see −treefuse

−fusingrounds n [2]
Under −treefuse, perform tree fusi#g (fusing−round 1) and fuse the resulting trees (n−1) times. Currently n can only be one or two (higher or lower values are set to 2 and 1, viz.)

−gammaalpha n
Under −likelihood and if specified, the value of the alpha parameter of the gamma distribution, if the user choses to set gamma classes > 0.

−gammaclasses n [0]
Under −likelihood the number of rate classes modeled under the discrete gamma.

−gap n [2]
Set insertion−deletion cost, also known as gap cost. If −extensiongap is set as well, −gap specifies the gap opening cost. If a −moleculermatrix is specified as well, gap costs in that matrix take precedence over

−gap.

−gc [nogc]
Output (stderr) Ocaml garbage collection statistics at the end of a run.

−getnewmatrix
Undocumented.

−goloboff "ck", "cm", or "ri" [no implied weighting]
Do implied character weighting (Goloboff 1993).

With k as set by −kfactor k,

- −goloboff ck performs implied weighting with implied weights as described in Goloboff (1993): (k + 1)/(k + 1 + h + m)
- −goloboff cm uses the product of Goloboff's implied weight and the minimum character length m
- −goloboff ri uses the unit retention index as implied weight

All weights described here are smaller than one. To keep cost calculations integer the weights are multiplied by a value that is set with the −multiplier command, and only the integer part is kept (this introduces a small and subtle systematic bias as function of m and g).

−help
Print (stderr, but see −helpfile) an overview of commands for getting more information.

Same as −−help

−helpdumpfile fname
Dump all available command line help (see −cat and list commands) in file fname.html (and fname.png).

−helpfile fname
Send requested command line help (see −cat and list commands) to file fname, not to stderr.

−helpwidth n [80]
Set the line width used for command line help.

Output from help comands (see −cat and list commands) is chopped into lines of at most n characters. Positive values below 60 are read as 60, negative number as +infinity. The latter is useful in combination with −helpfile. This comand has no effect for −helpdumpfile

−holdmaxtrees n [the value of −maxtrees n]
Maximum number of trees kept over all replicates.

see −maxtrees

−hypancfile filename
Set the name for the file to which −printhypanc and −printlotshypanc send their output.

−hypancname [nohypancname]
Identify inner nodes in the output of −printhypanc and −printlotshypanc. Do not use this when the aim is to generate input for −newstates.

−iafiles [−noiafiles]
Create a series of output files that have implied alignments (ia) for the different trees of a run. These trees can be input trees (using −topology or −topofile + −topodiagnoseonly), tree# that result from searches, or both. For each sequence input file, there will be as many output files as there are trees. Assuming that file 'trees' contains 3 trees, and that files test.data.1 and testdata2 contain sequence data,

```
poy test.data.1 testdata2 −iafi −topof trees −topodiagnoseonly
```

will create the following outptfiles:

```
ia.trees:              trees in parenthetical notation
test.data.1.ia.tree0:  ia for test.data.1 and first tree of ia.trees
test.data.1.ia.tree1:  ia for test.data.1 and second tree of ia.trees
test.data.1.ia.tree2:  ia for test.data.1 and third tree of ia.trees
testdata2.ia.tree0:    ia for testdata2 and first tree of ia.trees
testdata2.ia.tree1:    ia for testdata2 and second tree of ia.trees
testdata2.ia.tree2:    ia for testdata2 and third tree of ia.trees
```

All these output files are in a format that can be read by poy. If a sequence input file contains more than 1 fragment (cf "#", under −fastashortname), then the implied alignments for that input file will be separate implied alignments for these different fragments (instead of a single implied alignment for the concatenated fragments). In the output files, different fragments are separated by a single space. Fragments that have been excluded with * will not be used to calculate the cost of the trees, and will not be included in the implied alignments that are produced with −iafiles.

If, e.g. file test has the following contents:

```
> a
a#aa
> b
a#aa
> c
aa#a
> d
aa#a
```

then

```
poy test −topology "(a b c d)[];" −iafi −topodiagnoseonly
```

will produce output files ia.trees and test.ia.tree0. file ia.trees will contain a binary resolution of (a b c d) (note that is in general not a good idea to have polytomies in input trees), with the cost of the tree between square brackets; file test.ia.tree0 will have contents

```
> a
    a_ aa
> b
    a_ aa
> c
    aa a_
> d
    aa a_
```

This can be turned into an implied alignment for the concatenated original fragments as follows:

```
poy test.ia.tree0 −deletegapsfrominput −topodiagnoseonly
     −topology "(a c d b)[];" −iafil test
```

This will produce

```
> a
    aaa
> b
    aaa
> c
    aaa
> d
    aaa
```

On the other hand,

```
poy test.ia.tree0 −topodiagnoseonly −topology "(a c d b)[];" −iafil test
```

will produce a very unsatisfactory global implied alignment and cost.

The −iafiles command may be helpful for partly automating successive rounds of the heuristic that is described under −fastashortname without loosing track of what one is doing, a.o. because it keeps the fragments grouped as they are grouped in the input files (but without '*'−excluded fragments). The caveat expressed there holds here as well. And as the examples above show, it is very easy to concoct absurd brews of '#', −nodeletegapsfrominput, −iafiles and related commands.

Note that −overwriteprotection and −overwritedataprotection for −iafiles only check for file ia.trees; if that file does not exist, other −iafiles that might exist will be overwritten. This is the case in some example scripts above.

−impliedalignment [noimpliedalignment]
Under direct optimization (default heuristic for tree alignment), output (stdout) a topology specific multiple alignment based on the inner node state asignments that can be output using −diagnose. This is currently only done for the first tree at the end of a run (but see −iafiles)

see −phastwincladfile −iafiles

−incremental
Undocumented.

−indices [noindices]
At the end of a run, output (stderr) the best and worst length of this run. For the best trees, also output (an estimate of) consistency index CI and retention index RI.

−intermediate [nointermediate]
During tree search, output (stdout) intermediate best trees from time to time. As −repintermediate, but with more frequent output.

−invariantsitesadjust [noinvariantsitesadjust]
Under −likelihood , adjust likelihoods for theta fraction invariant sites.

−iterativeinitfinal
Undocumented.

−iterativeinitsingle [noiterativeinitsingle]
Under −iterativepass, initialize HTU nodes with reconstructed final states with any ambiguities resolved.

−iterativekeepbetter [iterativekeepbetter]
Under −iterativepass, keep better HTU reconstructions between normal up/down pass and iterativepass.

−iterativelowmem [noiterativelowmem]
Under −iterativepass minimally allocate memory. This can be a big (factor of 100) savings in memory, but there is overhead.

−iterativepass [noiterativepass]
Perform 3−dimensional (using 3 vertices attached to any internal node) optimization−alignment to improve HTU estimation and tree length. Sequences must be similar in length.

−iterativepassfinal [noiterativepassfinal]
Under −iterativepass, use iterativepass to re−estimate final states.

−iterativerandom [noiterativerandom]
Under "iterativepass" reconstruct HTU final states but resolve ambiguities with random preference.

−jackboot [nojackboot]

Under −nojackstart (default), perform Farris et al. (1996) parsimony jackknife procedure with −replicates n pseudoreplicates (note that −jackboot changes the regular meaning of the −replicates command). The tree search strategy within each pseudoreplicate can be set by using the commands that affect trees search within regular replicates. E.g.

```
poy mydata −replicates 200 −oneasis −buildperreplicate 5 −tbr −buildspr
```

constructs and analyzes 200 pseudoreplicates; for each pseudoreplicate, POY does five addition sequences (for the first, the taxon order is as in the first dataset; the following four use a pseudo−random order according to the −seed option), each followed by spr; the resulting best trees are subjected to tbr.

The default output (stdout) consists of the n strict consensus trees for the pseudoreplicates and a majority consensus trees of these strict consensus trees. The frequencies of clades on this tree are the jackknife percentages. The default output can be changed with the commands −jackoutgroup, −jackpseudoconsensustrees, −jackpseudotrees, −jacktree, −jackfpseudoconsensustrees, −jackfpseudotrees, −jackftree, and −jackwincladfile. Tree search strategies that are too shallow often lead to understimation of the jackknife support values.

see also −jackstart

−jackcharfile fname
If −jackboot and −nojackstart, output to file fname the majority rule consensus tree of the strict conse#sus trees of all jackknife pseudoreplicates, as an xread statement with group inclusion characters. The resulting file can be used as a −constrain file.

see −jackftree −poystrictconsensuscharfile

−jackfpseudoconsensustrees fname
If −jackboot and −nojackstart, output to file fname the strict consensus trees for the individual pseudoreplicates (parenthetical notation; for each consensus tree the number of best trees on which it is based is indicated between square brackets).

see −jackpseudoconsensustrees

−jackfpseudotrees fname
If on, and if −jackboot and −nojackstart, output to file fname the individual best trees for all pseudoreplicates (parenthetical notation with length indication between square brackets).

see −jackpseudotrees

−jackfrequencies "off", "majority", or "all" [majority]
If −jackboot and −nojackstart, output (stdout) a list of frequncies of all clades that are in the set of strict consensus trees of pseudoreplicatesclades (all); of all clades in the jacktree (majority); or suppress output of this list (off).

−jackftree fname
If −jackboot and −nojackstart, output to file fname the majority rule consensus tree of the strict consensus trees of all jackknife pseudoreplicates (parenthetical notation followed by text−based graphics; the latter is included as a comment, so the file can be used as an inputfile for −topofile). The clade frequencies in this tree are the jackknife support values.

see −jacktree −jackcharfile

−jackoutgroup taxonname.
Set the single taxon to be used as outgroup for jackknifing. Defaults to −outgroup if the −outgroup command is present, otherwise to the first taxon in use (see −terminalsfile).

see −jackboot −outgroup

−jackpseudoconsensustrees [jackpseudoconsensustrees]
If on, and if −jackboot and −nojackstart, output (stdout) the strict consensus trees for the individual pseudoreplicates (parenthetical notation; for each consensus tree the number of best trees on which it is based is indicated between square brackets).

see −jackfpseudoconsensustrees

−jackpseudotrees [nojackpseudotrees]
If on, and if −jackboot and −nojackstart, output (stdout) the individual best trees for all pseudoreplicates (parenthetical notation with length indication between square brackets).

see −jackfpseudotrees

−jackstart [nojackstart]
Use jacknifing when doing the −replicates but revert to −nojackboot before final tree refinement. This is useful to generate a diversity of starting trees as input for final refinement.

−jacktree [jacktree]
If on and if −jackboot and −nojackstart, output (stdout) the majority rule consensus tree of the strict consensus trees of all jackknife pseudoreplicates (parenthetical notation followed by text−based graphics). The clade frequencies in this tree are the jackknife support values.

see −jackftree

−jackwincladfile filename
If −jackboot and −nojackstart, create the file filename wit# a tread statement that describes the strict consensus trees of the individual pseudoreplicates in parenthetical notation (this file cannot be used as an argument for −topofile because it does not contain an xread statement (see −topofile for more information); use −jackpseudoconsensustrees for this purpose).

−jobspernode n [1]
Under −parallel, set the average number of regular slave processes per node that will be spawned on PVM nodes other than the master node. See Setup of a parallel run

−kfactor n [0]
See −goloboff

−leading [leading]
Count leading and trailing gaps. Often leading and trailing gaps result from sequencing with various primers by various investigators and are not evolved differences in sequence length. If this is the case, −noleading might help (but see Smith et al. 1981: 41).

With −noleading, leading and trailing gaps are counted and accounted for during builds and refinements (to prevent trivial alignments) but discounted when determining tree length.

see also −trailinggap −gap

−likelihood [nolikelihood]
Use likelihood as optimality criterion. The default, −nolikelihood, is parsimony analysis. Under likelihood, morphological characters are treated as in Tuffley &Steel (1997). Likelihoods L are reported as −ln L.

−likelihoodconvergancevalue n [1]
Under −likelihood, similarity threshold for two likelihood scores to be considered equal. That is, the difference in two likelihood must be less than this number to be considered equal. Increase in this value will speed up likelihood calculations. The units are 1/likelihoodroundingmultiplier, which is set to 100 by default. Hence the default identity threshold would be 0.01 log likelihood units.

−likelihoodestimationsize n [default under −parallel = all pairwise; in sequential = number of taxa]
Under −likelihood the number of pairwise sequence comparisons used to estimate likelihood parameters. In parallel, the default is all pairwise comparisons; in sequential mode the default is the number of taxa.

−likelihoodesttranseachtime [likelihoodesttranseachtime]
Under −likelihood, estimate transition matrix (q) for each comparison between two sequences.

−likelihoodextensiongap n
Under −likelihood, increase in likelihood of extension (affine) gaps, Akin to −extensiongap n in parsimony analysis. This parameter is a means of including non−linear gap costs in likelihood analyses, but beware that it is a kludge rather than a true model parameter.

−likelihoodmaxnumiterations n [100]
Under −likelihood , the maximum number of branch length iterations to be performed.

−likelihoodnoncodinggap
Undocumented.

−likelihoodroundingmultiplier n [100]
Under −likelihood, internal optimization alignments are performed using double precision floating point arithmetic, but are converted to integers on return. This value sets the rounding value to 1/n. The default precision then 0.01 log likelihood units.

−likelihoodstep n [5]
Under −likelihood, the size of iteration steps during branch length optimization. A larger number denotes a smaller step size.

−likelihoodstepinterval
Undocumented.

−likelihoodtrailinggap n
Under −likelihood, increase in likelihood of leading and trailing gaps. As to −trailinggap in parsimony analysis, this parameter is a means of including differential gap costs in likelihood analyses, but beware that it is a kludge rather than a true model parameter.

−list xyz
Print (stderr) a list of all commands that contain the string xyz. Use "−list −" for all commands

see −help −cat_commandbrowsing −cat_helptopics

−llist xyz
Print (stderr) a list of all commands that contain the string xyz an include their documentation. Use "−llist −" for all commnds.

see −help −cat_commandbrowsing −cat_helptopics

−lowmemthreshold
Undocumented.

−maxiterations n [no limit]
Under −iterativepass, this is a stopping rule to limit the number of iterative passes to perform if stability is not achieved.

−maxprocessors n
Deprecated. See Setup of a parallel run

−maxtrees n [no limit]
Set the default value for −buildmaxtrees, −fusemaxtrees, −holdmaxtrees, −sprmaxtrees, and −tbrmaxtrees.

see also −fitchtrees

−minstop n [0]
Minimum number of replicates that must be completed before −stopat will come into effect (see Giribet et al. 2001 for an example). For example:

```
poy −parallel −controllers 8 −multirandom −replicates 64 datafile
```

will perform 64 replicates, distributed over the 8 controllers.

```
poy −parallel −controllers 8 −multirandom −replicates 64
     −minstop 15 −stopat 3 datafile
```

will start out in the same way but will stop searching as soon as the same minimum tree length is found at least three times and after having completed at least 15 tree searches.

−minterminals n [80]
See terminalsfile

−molecularmatrix filename
Read matrix for molecular characters from file filename. The format is five lines each with five integers signifying the transformation costs among molecular character states followed by an optional closing semicolon (if this semicolon is present, text after the semicolon is not parsed; this can be used to add a comment to the file). If the commandline has more than one −molecularmatirix, only the last one is used (there is currently no way to assign different −molecularmatrices to different datasets).

Example: a file with contents

```
0  1  1  1  1
 1  0  1  1  1
 1  1  0  1  1
 1  1  1  0  1
 1  1  1  1  0
```

specifies the following cost matrix:

```
from\to A  C  G  T  gap
    A    0  1  1  1  1
    C    1  0  1  1  1
    G    1  1  0  1  1
    T    1  1  1  0  1
  gap    1  1  1  1  0
```

The costs specified with −molecularmatrix take precedence over −gap and −change.

If −extensiongap is specified as well, the last row and last column are gap opening costs.

see also −leading −trailinggap

−multibuild [nomultibuild]
Under −parallel, modify the parallelization of −buildreplicates over slave tasks (coarser granularity). {as of 3.0.10, the old command '−multibuild n' no longer exists}

−multidrift [nomultidrift]
Under parallel, modify the parallelization of −driftspr and −drifttbr over slave tasks (coarser granularity).

−multiplier n [100]
See −goloboff

−multirandom [nomultirandom]
Under −parallel, modify the behavior of −replicates by sending out complete replicates to other tasks. These can be regular slave tasks (controllers 0) or controller tasks (controllers n, n > 1). In the latter case, the controller tasks will parallelize any replicate that they deal with over the regular slave tasks that they have at their disposal. Under the default (−nomultirandom), the master task directly parallelizes replicates over all available regular slaves (in that case, controllers are never used).

−multiratchet [nomultiratchet]
Under −parallel, modify the parallelization of −ratchetspr and −ratchettbr over slave tasks (coarser granularity).

−newstates filename
Use the sequences in file filename as additional states for fixed states sequence optimization ('search−based optimization'; Wheeler, *in prep.*). If −newstates is used, each −fixedstates character data file must have its own −newstates file. Both the character datafiles and the newstates files are ordered as they are encountered on the commandline, and associated accordingly. For example, in

```
poy data1 −newstates data2.additional −fixedstates data2 data3 −nofixedstates
     data4 −newstates data3.additional
```

newstates file data2.additional is ssociated with character data file data2, newstates dile dta3.additional with character data dile data3 (make sure that the numbers of files match − there is not test yet). The format of a newstates file is just a series of unnamed sequences, one per line (empty lines are allowed). The last sequence can be followed by a line that starts with a semicolon (if the semicolon is present, there can be comments on the following lines).

see −printhypanc−printlotshypanc

−noapproxbuild see −approxbuild

−noapproxdrift see −approxdrift

−noapproxquickspr see −approxquickspr

−noapproxquicktbr see −approxquicktbr

−nobayesroundup see −bayesroundup

−nobremer see −bremer

−nobremerspr see −bremerspr

−nobuild
Deprecated; use −topodiagnoseonly instead.

−nobuildspr see −builspr

−nobuildtbr see −buildtbr

−nocharacterweights see −characterweights

−nocompressstates see −compressstates

−noconstrain #ee −constrain

−nocutswap see −cutswap

−nodeletegapsfrominput see −polyaddconverttorange

−nodiagnose see −diagnose

−nodiscrepancies see −discrepancies

−nodp see −dp

−nodpm see −dpm

−nodpmautoadjustperiod see −dpmautoadjustperiod

−nodriftspr see −driftspr

−nodrifttbr see −drifttbr

−nodropconstraints see −dropconstraints

−noenabletmpfiles see −enabletmpfiles

−noestimatep −see estimatep

−noestimateparamsfirst see −estimateparamsfirst

−noestimateq −see estimateq

−noexact see −exact

−nofastashortname see −fastashortname

−nofitchtrees see −fitchtrees

−nofixedstates see −fixedstates

−nogc see −gc

−nogetnewmatrix see −getnewmatrix

−nohypancname see −hypancname

−noiafiles see −iafiles

−noimpliedalignment see −impliedalignment

−noincremental see −noincremental

−noindices see −indices

−nointermediate see −intermediate

−noinvariantsitesadjust see −invariantsitesadjust

−noiterativeinitfinal see −iterativeinitfinal

−noiterativeinitsingle see −iterativeinitsingle

−noiterativekeepbetter see −iterativekeepbetter

−noiterativelowmem see iterativelowmem

−noiterativepass see −iterativepass

−noiterativepassfinal see −iterativepassfinal

−noiterativerandom see −iterativerandom

−nojackboot see −jackboot

−nojackpseudoconsensustrees see −jackpseudoconsensustrees

−nojackpseudotrees see −jackpseudotrees

#

−nojackstart see −jackstart

−nojacktree see −jacktree

−noleading see −leading

−nolikelihood see −likelihood

−nolikelihoodesttranseachtime see −likelihoodtranseachtime

−nomultibuild see −multibuild

−nomultidrift see −multidrift

−nomultirandom see −multirandom

−nomultiratchet see −multiratchet

−noncodinggap
[have to add]

−noonan see −onan. Deprecated

−nooneasis see −oneasis

−nopairmatrix see −pairmatrix

−noparallel see −parallel

−noplotinputtrees see −plotinputtrees

−nopolyaddconverttorange see −polyaddconverttorange

−nopoyini
Don't automatically use the file poy.ini in the current directory as a commandfile. The contradiction that arises when the command is in poy.ini itself, or in a commandfile that is reached through poy.ini, is arbitrarily resolved by assuming that in this case poy.ini contains the single command −nopoyini. In this way the outcome is the same, whether the file is used or not or anything in between. Yet this can be overruled by explicitly entering −− poy.ini on the commandline further on.

−noprealigned see −prealigned

−noprintccode see −printccode

−noprinthypanc see −printhypanc

−noprintlotshypanc see −printlotshypanc

−noprintqmat see −printqmat

−noprinttree see −printtree

−noquick see −quick

−norandomizeoutgroup see −randomizeoutgroup

−norandomizeslaves see −randomizeslaves

−norecode see −recode

−norepintermediate see −repintermediate

−norerootafterbuild see −rerootafterbuild

−noshowiterative see −showiterative

−nospewbinary see −spewbinary

−nospr see −spr

−nostaticapprox see −staticapprox

−nostaticapproxbuild see staticapproxbuild

−nostats see −stats

−notbr see −tbr

−notime see −time

−notopodiagnoseonly see −build

−notopolist see −topolist

−notoposkipidentical see −toposkipidentical

−nototallikelihood see −totallikelihood

−notreefuse see −treefuse

−notreefusespr see −treefusespr

−notreefusetbr see −treefusetbr

−notrullytotallikelihood see −trullytotallikelihood

−noupincremental see −upincremental

−noverbose see −verbose

−numdriftchanges n [20]

Number of topological changes to accept per drift round (Janies &Wheeler 2002.

−numdriftspr n [1]
Under −driftspr, set the number of drift rounds.

−numdrifttbr n [1]
Under −drifttbr, set the number of drift rounds.

−numslaveprocesses n [(number of entries in PVM configuration − 1) times −jobspernode n]
Under −parallel, the maximum number of regular slave jobs to spawn on PVM nodes other than the masternode.See Setup of a parallel run

−onan [noonan]
See Setup of a parallel run

−onannum n [0]
See Setup of a parallel run

−oneasis [oneasis]
For the first addition sequence in the build step of the first replicate, use the order of the terminals as found in the first data file (no −terminalsfile) or as found in the (concatenated) terminalfile(s).

−onversionconflict "exit", "deletehost", or "warn"[deletehost]
Under −parallel, set what to do when POY detects a difference in version numbers or version dates (as reported to stderr when tasks start) between the master task and a slave task.

- "exit": immediately exit the program (this can be problematic whem the nodes are added to the PVM configuration while POY is running because the addition of one node with a conflicting POY version near the end of a long run will make POY abort immediately)
- "deletehost": remove the host on which the conflicting slave task runs from the pool of hosts that is used (in a PVM environment with a single host this will make the program exit) [not yet implemented; fttb this is converted to warn]
- "warn": just issue a warning to stderr

.

−outgroup taxonname
Set the terminal that will be used as outgroup for building trees by stepwise addition under −norandomizeoutgroup and −nooneasis; and as outg#oup for −rerootafterbuild. The taxon that is specified with this command also changes the default taxon for −plotoutgroup and −jackoutgroup. If −outgroup is not specified the default outgroup is determined as follows:

- −terminalsfile is specified: the first terminal in the −terminalsfile
- −terminalsfile is not specified: the first terminal in the first character dataset on the commandline.

see also −topooutgroup

−overwritedataprotection "on" or "off" [on]:
Allow POY to overwrite input files that are used in the current run (the input is read before the file is overwritten). For −iafiles, the default (on) only checks for file ia.trees. The use of −overwritedataprotection on is doubtful. E.g.,

```
poy mydata −agree −constraint mygroups −poystrictconsensuscharfile
     mygroups −overwritedataprotection off
```

would overwrite existing constraints in file mygroups with the groups that are in the strict consensus tree of the current run (default tree search strategy).

The default (−overwritedataprotection on) obviously has better use. As with −overwriteprotection, these checks are performed before POY starts tree calculations, and the program exits immediately if any input file would be overwritten later on (but see −overwriteprotection for shell redirection of output and write permissions).

Contrary to −overwriteprotection, these checks, fttb, are simple lexical comparisons of filenames as entered on the command line, taking into account the −datadir setting. So, working in the /home/me home directory of a gnu/linux box, the two following will prevent datafiles to be overwritten:

```
poy −mydata −molecularmatrix 111 −printtree −plotfile 111
poy −mydata −molecularmatrix ../111 datadir ../ −plotfile 111
```

But this one will not:

```
poy −mydata molecularmatrix 111 −printtree −plotfile ../me/111
```

−overwriteprotection "on" or "off" [off]
Protect existing files from being overwritten with output files from POY; if the option is on, and an existing file would be overwritten, POY gives a notification and exits (but see −iafiles for the exception). These checks occur at the beginning of a run, so tree calculations won't even start if files would be overwritten and there is no danger of loosing trees after lengthy calculations.

Contrary to −overwritedataprotection, this command checks physical existence of files. So if POY is ran from directory /home/me on a gnu/linux box, and there is a file 'myfile' in this home directory, then all of the following will make POY exit with an error message before tree calculations are started:

```
poy mydata −printtree −plotfile myfile −overwriteprotection on
poy mydata −printtree −plotfile ../../home/../home/me/myfile −overwriteprotection on
poy mydata −printtree −plotfile ./myfile −overwriteprotection on
```

This one will not (provided there is no existing myfile in /usr/bin):

```
poy mydata −plotfile /usr/bin/myfile −overwriteprotection on
```

But that would not be of great help if you don't have write permission in that directory. In case you don't, POY will do all calculations and then crash as it tries to write that file, fttb.

Note also that these checks occur in POY, not in the shell from which you run your scripts. So, using bash standard output redirection,

```
poy mydata -overwritedataprotection on > mydata
```

would still physically transform your data into the results that you obtain with the default tree search strategy.

```
poy my#ata -overwritedataprotection -printtree -plotfile mydata
```

on the other hand would not because the tree calculations won't start in the first place.

-pairmatrix [nopairmatrix]
output (stdout) a matrix of pairwise distances between terminals in use (see -terminalsfile). The distance is over all character data files. The distance dor a sequences (or fragment) is the exact minimum edit cost.

-parallel [noparallel]
Execute in parallel using PVM. The code for parallel execution is fault tolerant: the master task detects if slave tasks die or become unreachable and in that case automatically adapts to the new situation. Partial results that might get lost by slave failure are recalculated on other slave tasks. See Setup of a parallel run

see -controllers -dpm -multibuild -multidrift -multirandom -multiratchet

-phastwincladfile fname
At the end of a run, write to file fname (1) an implied alignment for the first tree in the tree buffer; (2) all trees, in parenthetical notation. The format used is understood by Phast (Goloboff 1996b, Winclada (Nixon 2002), and poy: data in xread format (see -printccode), trees in parenthetical notation using numeric codes for the terminals (see -printccode for information on the additional output). In addition to the implied alignment(s), the xread also contains any morphological data that are used. Different sequences (or sequence fragments) and morphological datasets are separated by a blank. As the xread can have numerically coded morphological data, nucleotides are recoded: acgt becomes 0123; polymorphisms are indicated between square brackets (e.g., N becomes [0123]); gaps are coded as a fifth state (numeric code 4)

see -impliedalignment -iafiles

-plotechocommandline [plotechocommandline]
Echo the commandline to -plotfile at the start of -printtree output.

The command -printtree appends its output to the -printtree file; with -plotechocommandline, it is easier to identify -printtree output from different runs.

−plotencoding "ascii" or "UTF−8" [ascii]
Set the characters that −printtree uses to plot trees:

ascii: only ascii−characters

UTF−8: UTF−8 encoded unicode characters

UTF−8 gives nicer output than "ascii" but requires unicode fonts and a file viewer that understands utf−8 character encoding (e.g. import as utf−8 encoded text in msword; or use vim with ":set encoding=utf−8" in a utf−8 supporting terminal (e.g. xterm −u8).

−plotfile filename
Set the filename to which −printtree directs its output. If the file already exists, output is appended at the end. If it does not exist, it is created. With filename = stdout (case sensitive), output is directed to standard output. If −printtree is on but −plotfile is not specified, −printtree uses file "poy.tree" for its output.

−plotfrequencies "off", "majority", or "all" [off]
Set if or how −printtree tabulates the frequencies of clades in the best trees.

off: no clade frequencies

majority: majority clade frequencies (absolute and percentagewise number of occurrences)

all: printtree tabulates frequencies of all clades (absolute and perce#tagewise number of occurrences) other values are interpreted as 0

−plotmajority "off", "short", or "long" [off]
Set if or how −printtree will plot the majority rule consensus tree.

off: majority rule consensus tree is not plotted

short: majority rule consensus tree is plotted without clade identification numbers

long: majority rule consensus tree is plotted with clade identification numbers; in this case each branch is labeled with a/b, in which a is the identification number and b the percentagewise clade frequency.

−plotoutgroup taxonname
Set the single taxon that is used as outgroup for −printtree. If −outgroup taxon_x is used, −plotoutgroup defaults to taxon_x instead of the first taxon that is in use (see −terminalsfile). Useful in combination with −randomizeoutgroup (in which case the calculation of, e.g., a strict consensus tree would otherwise make little sense).

−plotstrict "off", "short", or "long" [long]
Set if or how −printtree will plot the strict consensus tree.

off: strict consensus tree is not plotted

short: strict consensus tree is plotted without clade identification numbers

long: strict consensus tree is plotted with clade identification numbers.

−plottrees "off", "short", or "long" [long]
Set if or how −printtree will plot the best trees.

off: best trees are not plotted

short: best trees are plotted without clade identification numbers

long: best trees are plotted with clade identification numbers.

−plotwidth n [80]
Set how many columns −printtree uses for its output. If a tree requires more than n columns to be plotted, it is broken into subtrees of appropriate sizes. Minimum value that can be set is 25. Names of terminals that are too long (about n−5 characters) are truncated.

−polyaddconverttorange [polyaddconverttorange]
Set how to deal with polymorphisms in additive characters (see −printccode) when the states that occur are not contiguous. With −polyaddconverttorange, the polymorphism is converted into the smallest range that encompasses the original polymorphism (as an example, [3578] becomes [345678]). With −nopolyaddconverttorange, the whole character is turned inadditive. If this happens with −phastwincladfile specified, the data in the phastwincladfile will reflect this conversion.

−poybintreefile fname
At the end of a run, write the tree(s) in poy parenthetical notation to file fname, with polytomies resolved (and with siblings ordered as to give correct diagnosis when the trees are re−imported later on in case '−dogaptie nopreference' is in effect; see −dogaptie).

Beware of spurious resolution when using this command to create a file to export trees to tree drawing programs. In that case −poytreefile might be better. If you want graphics of your trees right away, use −printtree.

see also −poystrictconsensustreefile −poystrictconsensuscharfile

−poystrictconsensuscharfile fname
At the end of a run, write the strict consensus tree of all trees in hennig86 xread format (see −printccode) to file fname; this file can be used as a −constrain file later on.

see also −poytreefile −poybintreefile −poystri#tconsensustreefile −jackcharfile

−poystrictconsensustreefile fname
At the end of a run, write the strict consensus tree of the trees in poy parenthetical notation to file fname; see −poytreefile for possible use. Do not use this tree to optimize characters in other programs!

see also −poytreefile −poybintreefile −poystrictconsensuscharfile

−poytreefile fname
At the end of a run, write the tree(s) in poy parenthetical notation to file fname; this can be useful when exporting, for the purpose of preparing figures, to a program that displays trees exactly as they are read (i.e., the program does not change the resolution on the basis of some optimality criterion that it applies to character data that it may or may not have).

If you want to use the trees trees later on to rediagnose in POY, use −poybintreefile instead. If you want graphics of your trees right away, use −printtree instead.

see also −poystrictconsensustreefile −poystrictconsensuscharfile

−prealigned [noprealigned]
During optimization of sequences and fragments in character data files that follow, apply some shortcuts for equal length sequences (do not use gap characters to make those sequences of equal length! this may cause undefined behavior). Make sure to turn of this toggle (using −noprealigned) in the command line before enterinf files with sequences and/or fragments of unequal length.

−printccode [printccode]
For every next hennig86/nona datafile (until −noprintccode), print a ccode−like summary of the characters settings in that file.

The entry for each character starts with the character number, followed by

```
[ or ]  for activity (active / not active)

+ or −  for additivity (additive / non−additive; additive characters have a linear
     character state tree)

/n      with n the character weight (taking into account −weight)

p       if the character is polymorphic (and does not have all possible states)
```

As an example, using a file mydata with contents

```
Xread
'an example for entering data in hennig86/nona xread format
'
4 5
Cichorium_intybus     0[01] 110
Rudbeckia_laciniata   01
                            101
Quercus_robur         1?    ?29
```

```
Anagallis_arvensis    02    032
;
tread
'my input tree'
(3 (0 1 2))
;
cc+.−1;
ccode ] 2 /2 3.;
tread
'more input trees'
((0 1) (2 3))*
(0(1(2 3));
p/;
this is not parsed
```

the command

```
poy −printccode mydata
```

will, besides echoing the comment, print the following line to stderr:

```
0[+/1     1[−/1p    2]+/2     3]+/2     4]+/2
```

After that, the program proceeds with the default tree search strategy.

There can be multiple ccode statements in a single file and character scopes can be entered in their general form:

```
2 5   −> characters 2 and 5
2.5   −> characters 2 to 5
.5    −> characters 0 to 5
5.    −> all characters from 5 on, 5 included
.     −> all  characters
```

Ccode statements are scanned from left to right, and whatever code specifiers are currently in effect are applied to the characters in the scopes as they are read. Besides [,], +, −, and /, ccode also accepts *, which discards all previous specifiers in a single ccode statement. '(' and ')', used in SPA (Goloboff 1995) and PHAST (Goloboff 1996b; see also Goloboff 1998) to t#ggle Sankoff characters, are read but ignored otherwise. The default settings are [−/1. So

```
cc /2 0.2 ] 3 4 *+ 3;
```

will turn character 4 inactive and assign weight 2 to characters 0, 1, 2, 4, and 5. Character 3 is made active but otherwise keeps its default settings [/1.

In the xread statement, the comment between single quotes is optional, but if present it must be right after the xread keyword. Next come the number of characters and the number of terminals, followed by the data, followed by a semicolon that terminates the xread statement. The data for each terminal start with the name of the terminal, followed by at least one blank character (space, tab, carriage return or linefeed) followed by the character information. Valid character codes are the digits 0 to 9; and '?' or '−' to indicate missing data. Polymorphisms must be enclosed between square brackets (such brackets are optional otherwise). Entries for different characters can but must not be separated by blanks. The file can have only one xread statement.

The input file is read until a "procedure/;" is encountered, anything after that is not parsed. A missing "p/" will cause an end−of−file error. Before the first "p/;", the file can also have tread statements, costs statements (cf Phast), and $ statements (cf. Clados). Other keywords will cause an invalid−keyword error. Commands can be abbreviated as in Hennig86 or Nona.

Costs statements and $ statements are skipped (input resumes after the next semicolon). Trees in tread statements are parsed but then ignored. If you want to use these trees as input trees as well, use

```
poy mydata −topofile mydata
```

In this command line, the first occurrence of mydata sets the data to be used (not using the trees), the −topofile mydata command then instructs poy to use the trees in that file as input trees (skipping the data).

The optional tread comment between single brackets must follow the tread keyword. Next follow trees in parenthetical notation. Trees are separated with '*' (or followed by '[n]', with n an optional integer) and the statement is closed with a semicolon. The trees must use numerical codes for the taxa, following the order of the taxa in the xread statement that must precede the tread statement (counting starts at 0). Parentheses must be balanced and taxon codes must be separated by at least one character (blank or parenthesis). As an example, (0(1(2 3))) in the above input file stands for

```
__ Cichorium_intybus
|____ Rudbeckia_laciniata
  |____ Quercus_robur
    |__ Anagallis_arvensis
```

When reading a data input file, poy first checks if the file starts with an xread statement (or with any other of the accepted keywords for hennig−86 like input files listed above). If it does, poy assumes that the datafile is a hennig86−like input file and parses it accordingly. In that case, if it encounters an error in the xread statement (e.g., wrong character state code, or insufficient number of characters for a particular terminal) or in any other part of the file (e.g., a tree format error in a tread statement), it will print an error message and exit.
If, on the other hand, the file does not start as a hennig86−like file (beware of unsupported keywords like 'dread' or 'bb'), poy will try to read the input file as a sequence input file (see −fastashortname for accepted formats). In case it cannot parse the file as a sequence file neither, poy will mostly print two error messages and exit. The first error message (between square brackets) is the error it encountered when trying to parse the input file as a hennigfile, the second the error when parsing as a sequence file.

see −fastashortname for formatting requirements of sequence data files
see −topofile for an alternative tree format

−printhypanc [noprinthypanc]For the trees that are in memory at the end of a run, write inner node sequence reconstructions to file poy.hypanc. Contrary to reconstructions as obtained with −diagnose, ambiguities are resolved in the reconstructions that are obtaind here. The resulting file can be used for search−based optimization (see −newstates).

see −hypancname −hypancfile −printlotshypanc

−printlotshypanc [noprintlotshypanc]

As −printhypanc, but add hypothetical ancestral sequence reconstructions of intermediate trees during tree search.

−hypancname −hypancfile −printhypanc

−printqmat [noprintqmat]
Under −likelihood, prints transition, base frequency, and other likelihood parameter information to stdout.

−printtree [noprinttree]
At the end of a run, plot (in ascii graphics) the trees and their strict consensus at the end of a poy run to file poy.tree. If that file exists, new output is appended at the end (a line with '>−

With −topodiagnoseonly, inputtopologies (−topofile and /or −topology) are plotted. In that case, there are two possibilities:

1. there are character datafiles. In that case, the inputtrees are diagnosed using all characters that are in use, and zero−length branches are collapsed accordingly (note that POY internally resolves polytomies of inputtopologies in an arbitrary way, which may influence diagnosis and presence of zero−length branches; it is in general not a good idea to import trees with polytomies when the trees are going to be diagnosed).

2. there are no character datafiles. In that case, the inputtrees are used with exactly the same resolution as they are read (uses temporary files, so −enabletmpfiles must be on).

Trees are considered to be unrooted, so the basal node is never resolved (there are some inconsistencies in the curent version).

The default behavior of −printtree can be changed with the options −plottrees, −plotstrict, plotmajority, −plotwidth, −plotencoding, −plotfrequencies, and −plotfile.

−qmatrix filename
Read transition matrix (5 x 5; order A C G T GAP, see −moleculamatrix) for use under −likelihood. The file must contain just 5 lines with each 5 numbers, or start in that way, followed by an optional semicolon (if the semicolon is present, the file can have additional lines that can be used for comments).

−quick [quick]
Do not swap on non−minimal length trees during branch swapping. This forces POY to swap only on minimum length trees despite the fact that other trees have been found in the swapping process. (Janies &Wheeler 2002)

−quote string
Echo the argument to stdout (comes before any other output to stdout). In combination with stdout redirection (see section Output and output files), useful to add a comment to the outfile.

−random n

Deprecated; use −replicates instead.

−randomizeoutgroup [randomizeoutgroup]
Randomize all terminals, including the outgroup, when calculating a random addition sequence to build a tree by stepwise addition.
With −norandomizeoutg#oup, the first terminal of the first data set is used as starting point for adding the other terminals (but see −outgroup).

−norandomizeoutgroup should be specified for constrained runs.

If −randomizeoutgroup is used with −jackboot and −nojackstart, −jackoutgroup should be set as well.

If randomizeoutgroup is used with −printtree to obtain consensus trees, −plotoutgroup should be specified as well.

−randomizeslaves [randomizeslaves]
Under −parallel, randomize the array(s) of slave tasks TIDS (task identification numbers) as returned by PVM after spawning. This may improve load balance on the cluster with multiple simultaneous runs.

−ratchetoverpercent n
Spawn out a number of extra ratchet jobs to accommodate for unequal execution time. Once a full complement of ratchet jobs are complete, unfinished jobs are killed and the search proceeds as specified under defaults or subsequent commands. (Janies &Wheeler 2002)

−ratchetpercent n [15]
Percent of characters to be reweighted in Nixon's (1999) ratchet procedure (Janies &Wheeler 2002).

−ratchetseverity n [2]
Weight multiplier for reweighted characters during parsimony ratchet searches (Janies &Wheeler 2002).

−ratchetslop n [value of −slop n]
Set slop values for ratcheting.

−ratchetspr n [0]
Iterative rounds of ratcheting procedure using SPR (Janies &Wheeler 2002).

−ratchettbr n [0]
Iterative rounds of ratcheting procedure using TBR (Janies &Wheeler 2002).

−ratchettrees n [2]
Number of trees to be saved during ratchet iterations (Janies &Wheeler 2002).

−recode [norecode]
Make non−additive and additive characters faster (Janies &Wheeler 2002).

−repintermediate [norepintermediate]
Output (stdout) intermediate best trees from replicates. As −intermediate, but with less frequent output.

−replicates n [0]
With −nojackboot, perform n replicates (see program structure).

With −jackboot, do jackknifing with n pseudooreplicates.

−rerootafterbuild [rerootafterbuild]
If −randomizeoutgroup is used, the trees that are obtained by stepwise addition are rerooted to the terminal that is specified with −outgroup before they are subjected to branch swapping and other tree search algorithms. Because the reconstructed ancestral states depend on the chosen outgroup (Wheeler 1996), this may change the length of trees as reported at the end of stepwise addition.

−seed n [−1]
Set seed for pseudorandom number generation. −seed −1 will cause the system time, in seconds, to be used. A seed of 0 and higher guarantees exact reproducibility of runs under −noparallel, but not under −parallel> Reason is that the program can in this case not know in advance the order in which various requests that it sends out to slaves w#ll be answered (this depends, a.o. on the processor speed of the slave job and its total load). With superficial tree search strategies and small tree buffers it is therefore possible that two consecutive runs of the same −parallel script give different results. This is then not a defect of the program but an indication that the script is not adequate to analyse the data at hand.

−showiterative [noshowiterative]
Under −iterativepass, print iterative pass progress information. Just for fun.

−slop n [0]
While doing tree refinements, accept all trees that are within n tenths of a percent of the current minimum value. For example

```
poy data −slop 10
```

accepts all trees up to 1% above the current minimum length while trying to refine trees. The effect of this command is twofold: it makes it easier to jump islands, and reduces the effects of shortcuts that at various stages are employed when calculating tree lengths.

See −buildslop, −checkslop, and −ratchetslop for setting different slop values for specific refinement procedures.

−solospawn n [4]
Under −parallel, set the number of regular slave tasks to be spawned in a PVM environment with only a single node. See Setup of a parallel run

−spewbinary [spewbinary]
At the end of a run, after outputting (stdout) the best trees with zero−length branches collapsed, output (stdout) the same trees with polytomies resolved. This (internal) resolution is required to rediagnose the trees later on.
see −poybintreefile

−spr [spr]
Perform spr branch during replicate refinement.

−sprmaxtrees n [value of −maxtrees n]
Set the maximum number of trees held while doing spr.

see −maxtrees −fitchtrees

−staticapprox [nostaticapprox]
Use static approximation for sequence optimization while doing tree refinements. Static approximation consists of using a fixed implied alignment while doing branch swapping and other kinds of tree refinement; every time a shorter tree is hit, a new implied alignment is calculated (based on that shorter tree) and this new alignment is used for further swapping.

see −staticapproxbuild

−staticapproxbuild [nostaticapproxbuild]
[Have to add].

see −staticapprox

−stats [stats]
Toggle for some additional output (stderr) of search statistics.

−stopat n [the value of −replicates]
Set the number of replicates that must have hit the same minimum tree length before the analysis ends. See also −minstop.

−submodel "s6g","s10", "s3g", "s2g", "s1g", or "s1" [s6g]
Under −likelihood enforces symmetries in transition probabilities. S10 has each of the 10 reversible transitions able to vary (a "super−GTR"), s6g is GTR+gaps with all gaps treated equally, s3g, s2g, s1g are 3, and2 parameter models with gaps, and Jukes−Cantor models + gaps, and s1 treats all transitions as

equally probable (a "super−JC").

−tbr [tbr]
Perform tbr branch swapping during replicate refinement.

−tbrmaxtrees n [value of −maxtrees n]
Set the maximum number of trees held while doing tbr.

see −maxtrees −fitchtrees

−terminalsfile fname
Use the entries in this file as names of terminals to be included in the analysis (free format: there my be multiple lines, and each line can have multiple names, or nothing at all; see command −editnames for format restrictions on names of terminals), and set the outgroup to the first terminal in that file. There can be multiple −terminalsfiles on one command line. In that case all entries of all terminalsfiles are added. Double occurences of names are filtered out.

With −terminalsfile specified, character inputfiles and inputtrees (−topology and −topofile) will automatically be conformed to the list of names to be included (without −terminalsfile specified, any inconsistency in taxon names between inputfiles will make the program abort immediately).

For character datafiles, this involves two things:

1. whenever a data file has character data information for a terminal that is not in the list of names, that information will be skipped.
2. whenever a data file has no data information for a terminal that is in the list of names, the program will assume missing data for the sequence or set of fragments that is involved.

Other inconsistencies or errors in names of terminals in datafiles (like double occurrences of names or names that contain parentheses) will still make the program abort.

For each inputfile, an overview of taxa included/excluded is printed to stderr.

This feature reduces the need to edit datafiles (you no longer need to include the names of taxa for which a sequence is missing, and you don't need to delete taxa that you don't want to include in the analysis). Without −terminalsfile, the old rule still applies: all inputfiles need to have exactly the same taxon sample, and missing sequences have to be indicated explicitly).

To prevent running an analysis with too much missing information, the folowing check is performed in each datafile: if the number of taxa to be added from the −terminalsfile list is more than 20% of the number of taxa that are in the file (after discounting the taxa that are NOT in the −terminalsfile namelist), than the program will exit. Use −minterminals to change this to another percentage (−minterminals 0 will basically skip this test).

For inputtrees (see −topology and −topofile), this involves the following:
1. whenever an inputtree has a terminal that is not in the list of names, that terminal will be pruned from the tree.

2. whenever an inputtree lacks a terminal that is in the list of names, the program will add that terminal as a basal branch to the tree.

In case trees have been modified, a list of terminals that are involved is printed to stderr. Without −terminalsfile, all inputtopologies must conform exactly to the set of taxa that are in use (this is more restrictive than in older versions of poy, where trees could have subsets of taxa). There is no −minterminals check for inputtrees. Note that, even with −terminalsfile(s) specified, individual −topology and −topofile commands must still be internally consistent (i.e. all trees in any single −topofile or −topology command must have the same taxon sample). Between −topofile and/or −topology commands taxon samples may differ. The check for identical inputtrees (see −toposkipidentical) occurs after the trees have been conformed to the terminals in use.

When no −terminalsfiles are specified, the set of terminals that is in use (and to which all datafiles must explicitly conform in that case) is determined as follows:

- If there are character datafiles, the set of terminals in the first datafile on the commandline.
- If there are no character datafiles, the set of terminals in the first −topology or −topofile command (without datasets, it is still possible, e.g., to calculate consensus trees and plot trees).

see also −outgroup

−theta n
Under −likelihood sets the proportion of invariant sites, "theta" The user can set theta for the duration of the search.

−time [time]
At the end of a run, output the total execution time to stderr.

−topodiagnoseonly [notopodiagnoseonly]
Build trees. Skip the replicate loop and skip final refinement. Can be used to evaluate or plot or consense input topologies, or to calculate implied alignments for inputtrees.

see −topology −topofile −plotinputtrees −printtree −poystrictconsensustreefile −poystrictconsensuscharfile −iafiles

−topofile filename
Read cladogram topologies (single or multiple) from a file. Cladograms must be in parenthetical notation as in POY output: using full names of terminals and with each tree followed by '*' or by '[n]' (in which n is an optional integer or real number); the entire list of trees must be terminated with a semicolon ';' (the last tree can have but does not need a "*" or a [n]).

Alternatively, trees can be in xread/tread format (see −printccode), as in outputfiles that can be generated with Hennig86, Nona or TNT. In that case the topofile must have one xread statement that precedes any tread statement, and the tread statements must use numerical codes to refer to the terminals (see −printccode for more information).

Names of terminals cannot have parentheses or some other special characters (see command −editnames).

Within any single −topofile, all trees must have exactly the same set of terminals. See −terminalsfile for how to deal with sets of terminals that differ between −topofiles, or with sets of terminals that differ from the terminals that are present in datasets.

Exactly what happens to the inputtrees depends on
1. whether or not character data files are present
2. the setting of −topodiagnoseonly/−notopodiagnoseonly (default is −notopodiagnoseonly).
3. the setting of −replicates.

If no character datafiles are present, the setting of −topodiagnoseonly and −replicates do not matter. In that case, the trees will be read and accepted with the resolution that is provided, with the exception that the basal node will never be resolved, so (a (b (c d))) will be treated as (a b (c d)). Possible uses are, e.g., calculation of consensus trees (−printtree, −poystrictconsensustreefile) and −constrain files (−poystrictconsensuscharfile), combination of several topofiles into one file with filtering out of identical trees (−toposkipidentical, −poytreefile), adding/deleting taxa from/to trees (−terminalsfile, −poytreefile), or rerooting topofiles (−topooutgroup, −poytreefile).

If character datafiles are present, the trees will be diagnosed and zero−length branches will be collapsed accordingly (if for all characters there is at least one optimization that assigns length zero to a particular branch, tha branch is collapsed − this oca#ionally leads to overcollapsing in individual trees but not to loss of resolution in their strict consensus). If −topodiagnoseonly is specified, these trees will further be treated as if no character data are present.

If, on the other hand, −topodiagnoseonly is off, the setting of −replicates matters. If −replicates n is specified and n > 0, poy will do n random addition sequences and add the results of these to he inputtrees before proceeding to final swapping. In the other case, poy directly proceeds to final swapping. Output of −printtree, −poytreefile, −poybintreefile, −poystrictconsensustreefile, and −poystrictconsensuscharfile occurs after final swapping.

If an outgroup taxon is specified with the command −topooutgroup, the trees will be rerooted to that outgroup.

If −toposkipidentical is specified, identical input trees (after rerooting if −topooutgroup is specified) across all −topology and −topofile commands will be filtered out.

see −topology to read tree(s) from the command line and −printccode for how to import xread/tread trees from Hennig86, Nona or TNT
see also −terminalsfile, −printtree, −topokeeprandom −topopwkeeprandom

−topolist [topolist]
After reading and diagnosing input trees, output (stdout) their binary representations with diagnosed lengths.

see −topology, −topofile, −toposkipidentical

−topology topology_string
As −topofile but with the trees on the commandline and not in a file, and with xread/tread format as available in −topofile NOT supported.

The trees must follow the command, and all trees together must be enclosed between double quotes ("). Trees must be in parenthetical notation as in POY output (see −topofile) and the last ree must be fllowed by a semicolon (inside the double quotes; in between the semicolon and the closing double quote comments can be added).

If an outgroup taxon is specified with the command −topooutgroup, the tree will be rerooted to that outgroup.
If −toposkipidentical is specified, identical input trees (after rerooting if −topooutgroup is specified) across all −topology and −topofile commands will be filtered out.

see −topofile and −printtree for interaction with −topodiagnoseonly and absence/presence of character datasets.

see also −printccode −terminalsfile −topofile

−topooutgroup taxonname
With −topooutgroup specified, the inputtrees will be rerooted to the specified taxon immediately after they are read.

As a side effect, sibling clades in the rerooted tree are ordered from small to large (or alphabetically for terminals). For its possible effect on diagnosed lengths and assessment of zero−length branches, see −dogaptie.

As an example, tree (((D C) B) A) will be read and diagnosed as (((D C) B) A) when −topooutgroup is not specified, but as (A (B (C D))) when using −topooutgroup A. With −topooutgroup B, the tree will be read and diagnosed as (B (A (C D))).

When there are no data inputfiles, trees are not diagnosed and branches are not collapsed.

see −topology −topofile −outgroup −dogaptie

−topopickrandom # [number of trees of all −topofile and −topology commands after −toposkipidentical/notoposkipidentical]
Keep only n inputtrees, pseudorandomly (see −seed) selected from all inputtrees after −toposkipidentical/notoposkipidentical. If n is bigger than the number of inputtrees, it is set to the number of inputtrees. Takes precedence over −topopwpickrandom

−topopwpickrandom n [100]
As −topopickrandom, but with n a percentage of all inputtrees after −toposkipidentical/notoposkipidentical. The number of trees to keep is truncated down to the nearest integer.

−toposkipidentical [toposkipidentical]
If on, filter out duplicate inputtrees. The filtering occurs after rerooting (if −topooutgroup is specified) and diagnosis + collapse of zero−length branches (if character data are present). Note that trees are considered as unrooted for this check, so (a(b c)) and (b(a c)) are considered to be the same trees.

−totallikelihood [nototallikelihood]
Under −likelihood, this command determines the likelihood of an optimization alignment by summing all major optimization alignments (to sum all, see −trullytotallikelihood). The default is the likelihood of the single or "dominant" optimization alignment likelihood which may be a very small fraction of the total likelihood.

−trailinggap n
Sets both leading and trailing gap cost to n. If −trailinggap is not specified, leading and trailing gaps are set equal to the overall gap cost specified by −gap n or −molecularmatrix filename. (Janies &Wheeler 2002)

−treefuse [notreefuse]
Perform tree fusing (Goloboff 1999).

see −fuselimit −fusemaxtrees −fusemingroup −fusingrounds −treefusespr −treefusetbr

−treefusespr [notreefusespr]
Perform SPR on new trees found during tree fusing (Janies &Wheeler 2002).

see −treefuse

−treefusetbr [notreefusetbr]
Perform TBR on new trees found during tree fusing (Janies &Wheeler 2002).

see −treefuse

−trullytotallikelihood [notrullytotallikelihood]
Under −likelihood, determine the likelihood of an optimization alignment by summing all optimization alignments. The default is the likelihood of the single or "dominant" optimization alignment, which may be a very small fraction of the total likelihood.

−upincremental
Undocumented.

−verbose [verbose]
Toggle for outputting more or less search progress information to stderr.

−weight n [1]
The next character data input file receives a weight of n.

see −defaultweight

REFERENCES

[Top] [Contents]

Farris, J.S. 1970. Methods for computing Wagner trees. *Syst. Zool.* 19: 83−92.

Farris, J.S. 1988. Hennig86 version 1.5. Program and manual distributed by the author, Naturhistori#ka riksmuseet, Box 50007, SE−104 05 Stockholm.

Farris, J.S., Albert, V.A., Källersjö, M., Lipscomb, D., and A.G. Kluge. 1996. Parsimony jackknifing outperforms neighbor−joining. *Cladistics* 12:99−124.

Fitch, W.M. 1971. Toward defining the course of evolution: minimal change for a specific tree topology. *Syst. Zool.* 20: 406−416.

Frost, D.R., Rodrigues, M.T., Grant, T., and T.A. Titus. 2001. Phylogenetics of the lizard genus *Tropidurus* (Squamata: Tropidurinae): direct optimization, descriptive efficiency, and sensitivity analysis of congruence between molecular data and morphology. *Mol. Phylog. Evol.* 21: 352−371.

Giribet, G., Edgecombe, G.D., and W.C. Wheeler. 2001. Arthropod phylogeny based on eight molecular loci and morphology. *Nature* 413: 157−161.

Goloboff, P.A. 1993. Estimating character weights during tree search. *Cladistics* 9: 83−91.

Goloboff, P.A. 1995. SPA. (S)ankoff (P)arsimony (A)nalysis, version 1.1 (32 bit version). Program and documentation. Computer program distributed by J. M. Carpenter, Dept. of Entomology, American Museum of Natural History, New York.

Goloboff, P.A. 1996a. Methods for faster parsimony analysis. *Cladistics* 12: 199−220.

Goloboff, P.A. 1996b. PHAST. (PH)ylogenetic (A)nalysis for (S)ankovian (T)ransformations, version 1.1 (32 bit version). Program and documentation. Computer program distributed by J. M. Carpenter, Dept. of Entomology, American Museum of Natural History, New York.

Goloboff, P.A. 1998. Tree searches under Sankoff parsimony. *Cladistics* 14: 229−237.

Goloboff, P.A. 1999. Analyzing large data sets in reasonable times: solutions for composite optima. *Cladistics* 15: 415−428.

Goloboff, P.A., and J.S. Farris. 1999. Methods for quick consensus estimation. *Cladistics* 17: S26−S34.

Janies, D.A., and W.C. Wheeler. 2001. Efficiency of parallel direct optimization. *Cladistics* 17: S71−82.

Janies, D., and W. Wheeler. 2002. POY version 3.0 documentation and command summary. Update November 6 2002.

Nixon, K.C. 1999. The parsimony ratchet, a new method for rapid parsimony analysis. *Cladistics* 15: 407−414.

Nixon, K.C. 2002. WinClada version 1.00.08 Published by the author, Ithaca, NY (see http://www.cladistics.com).

Sankoff, D. 1975. Minimal mutation trees of sequences. *SIAM J. Appl. Math.* 28: 35−42.

Sankoff, D., and P. Rousseau. 1975. Locating the vertices of a Steiner tree in an arbitrary metric space. *Mathematical Programming* 9: 240−246.

Smith, T.F., Waterman, M.S., and W.M. Fitch. 1981. Comparative biosequence metrics. *J. Mol. Evol.*18: 38−46.

Tuffley, C. and M. Steel. 1997. Links between maximum likelihood and maximum parsimony under a simple model of site substitution. *Bulletin of Mathematical Biology* 59: 581−607.

Wang, L., and T. Jiang. 1994. On the complexity of multiple sequence alignment. *J. Comp. Biol.* 1: 337−348.

Wheeler W. 1996. Optimization alignment: the end of multiple sequence alignment in phylogenetics? *Cladistics* 12: 1−9.

Wheeler, W. 1999. Fixed character states and the optimization of molecular sequence data. *Cladistics* 15: 379−385.

Wheeler, W., and D. Gladstein. 1994−2000. MALIGN. Software for multiple sequence alignment. Available at ftp://ftp.amnh.org/pub/molecular/malign.

Wheeler, W. and C.Y. Hayashi. 1997. The phylogeny of the extant chelicerate orders. *Cladistics* 14: 173−192.