# Correspondence



http://doi.org/10.11646/zootaxa.5642.1.10 http://zoobank.org/urn:lsid:zoobank.org:pub:B904C0E6-0AD9-4AC4-B67E-8DA1B53203D1

# guoyi.run: One-step TNT script for maximum parsimony phylogenetic analysis

# GUOYI ZHANG<sup>1,2</sup>

<sup>1</sup>Evolution & Ecology Research Centre, School of Biological, Earth and Environmental Sciences, University of New South Wales, Sydney 2052, Australia.

<sup>2</sup>Australian Museum Research Institute, Australian Museum, 1 William St, Sydney 2010, Australia.

starsareintherose@gmail.com, guoyi.zhang@austmus.gov.au; https://orcid.org/0000-0002-3426-9273

Maximum parsimony (MP) analysis remains a cornerstone in phylogenetics, particularly for studies based on morphological characters. TNT is the fastest and most efficient software package available for maximum parsimony (MP) analysis. It is a very flexible and interactive program that can be used for many different tasks, and this very flexibility and number of options are what makes the learning curve steep. Here, I introduce guoyi.run, a TNT script designed to provide a fully automated, one-step pipeline for MP analysis using the TNT macro interpreter language. It can be downloaded from https://github.com/starsareintherose/TNT\_Script.

Maximum parsimony (MP), as a non-model-based phylogenetic method, has been widely applied in phylogenetics based on morphological characters (Kluge & Farris, 1969; Farris, 1970; Fitch, 1971). In addition, MP methods have demonstrated remarkable speed and performance when applied to genomic datasets, providing a fast and robust alternative to model-based phylogenetic methods (Kolaczkowski & Thornton, 2004; Wheeler *et al.*, 2017; Torres *et al.*, 2022).

TNT (Tree analysis using New Technology) stands out as the fastest and most efficient software package for MP analysis (Goloboff *et al.*, 2022). It incorporates a variety of advanced search algorithms, such as fuse, drift, ratchet, and others that have been specifically designed to navigate the vast tree space effectively, even for large and complex datasets (Goloboff *et al.*, 2008; Goloboff & Catalano, 2016; Wheeler *et al.*, 2017; Torres *et al.*, 2022; Goloboff & Morales, 2023). Despite its exceptional computational capabilities and rich command set, TNT's command-line interface can be daunting for beginners. Although a graphical user interface (GUI) exists (Goloboff *et al.*, 2008; Goloboff & Morales, 2023), it lacks a fully automated pipeline that integrates all the necessary steps into a single, user-friendly workflow.

The new TNT script guoyi.run offers a complete, one-step automated pipeline for MP analysis based on the macro interpreter language of TNT (Goloboff & Morales, 2020). The pipeline is flexible to most phylogenetic parameters including dataset types, weighting settings, K values, consensus types, and resampling methods. This script makes TNT's powerful methods accessible to beginners. I encourage users to follow and understand each step in the guoyi.run script, which will help them gradually master TNT and leverage its advanced methods independently.

# **Basic settings**

The TNT script guoyi.run is developed as an integrated solution for maximum parsimony analysis, supporting various data matrix formats such as the native Hennig86/TNT, FASTA, and NEXUS formats. It operates in two distinct modes: a shell command line mode and an interpreter mode. In the shell command line mode, the script is executed with the command "<TNT command line executable binary path> run <guoyi.run path> <data matrix path> <; or ,>", where Unix/Linux users should terminate the command with a comma and Windows users with a semicolon. Alternatively, in the interpreter mode, the script is run using the command "run <guoyi.run path> <data matrix path>;". If the script is placed in the same directory, "run <guoyi.run path>" can be replaced with "guoyi". All path names should not contain any non-English and space characters.

# **Advanced settings**

A prominent feature of guoyi.run is its flexible parameter configuration. An extended command structure is provided as "<TNT command line executable binary path> run <guoyi.run path> <data matrix path> <data type> <weighting type> <0 for equal weighting / K value for (extended) implied weighting> <consensus type> <resample methods> <prefix> <; or ,>", and the script supports nearly all data types available in TNT—excluding landmarks—such as discrete numbers (2–32 states), continuous numbers (cont), nucleic acids (dna), and amino acids (port).

For weighting fragments, three options are available: extended implied weighting (eiw), standard implied weighting (iw), and equal weighting (ew). The default configuration utilizes extended implied weighting with a K value of 12, as recommended by Goloboff *et al.* (2018), because this approach effectively mitigates the influence of homoplastic characters and improves tree reliability; however, users may select standard implied or equal weighting to suit different datasets and analytical hypotheses.

The tree search strategy employed by the script is adaptive, adjusting to the dataset's size and complexity. For datasets containing 20 or fewer taxa under non-extended implied weighting, the script uses implicit enumeration to exhaustively search the tree space, ensuring that all most-parsimonious trees (MPTs) are identified. For intermediate datasets, comprising 20 to 75 taxa, or for datasets with 20 or fewer taxa under extended implied weighting, a tree bisection and reconnection (TBR) approach is implemented, which conducts 1,000 iterations of branch swapping to thoroughly explore the topology space. In the case of larger datasets, with 75 or more taxa, the search proceeds until 50 best-length tree scores are hit independently, beginning with 20 replications and followed by 10 cycles each of drifting, ratchet, and fusing to further refine the tree exploration.

Once the tree search is completed, guoyi.run proceeds to construct consensus trees and calculate support metrics. By default, the most-parsimonious trees are summarized using the strict consensus method (str), though users may opt for the majority rule (mjr) or half strict consensus (hlf) approaches. Custom resampling methods can be applied through a straightforward summation: the default values being relative Bremer support at 0.1, Bremer support at 0.2, jackknifing at 1, bootstrapping at 2, and symmetric resampling at 4. The script primarily employs, jackknifing, bootstrapping, and symmetric resampling (value = 7), with each resampling procedure repeated 1,000 times to produce statistically robust clade support estimates. In addition, the script maps apomorphic character changes onto the consensus tree and computes essential tree metrics, including tree length (TL), consistency index (CI), retention index (RI), and each character's homoplasy. When non-equal weighting is used, it further reports character concavities.

Upon completion of the analysis, guoyi.run produces a series of output files intended to facilitate further interpretation and downstream applications. These include tree files that either include or exclude taxon names, as well as files (designated as "prefix.>winclada.ss") formatted for direct import into WinClada to analyse further apomorphic and homoplasic characters. Additionally, SVG format output files are generated, ensuring seamless compatibility with vector graphic editors such as Inkscape.

#### **External software**

Along with guoyi.run, an external command line tool named tht2figtree written in C++ is provided. It can convert TNT's nexus-format output with group support into a format that FigTree can readily visualize, embedding support values as nexus node labels. This tool is cross-platform, ensuring compatibility with various operating systems.

#### Conclusion

The script guoyi.run provides intuitive and user-friendly access to the advanced technological phylogenetic software TNT using native TNT macro programming language. The implementation allows researchers with different levels of computer skills to use good performance and advanced algorithms to reconstruct phylogenetics. Moreover, while the default TNT pipeline employs a relatively exhaustive search strategy, running the equivalent analysis in PAUP\* (Swofford, 2003) would be markedly slower. In practice, most PAUP\* users perform only a single random-addition replicate followed by branch swapping, an approach that completes in less wall-clock time than 1,000 TNT replicates, despite TNT's faster perreplicate performance (Goloboff *et al.*, 2022). Its extensive customization options ensure that it can be readily adapted to meet the needs of diverse datasets and research questions, ultimately contributing to better practices of phylogenetic analysis. The script and external software tnt2figtree are available on GitHub (https://github.com/starsareintherose/TNT\_ Script) and self-host git website (https://git.malacology.net/malacology/TNT\_Script). Additionally, guoyi.run can be installed from BioArchLinux (Zhang *et al.*, 2025) under the name tnt-guoyi-script.

# Acknowledgements

Initially, this project aimed to support my friend Encui Wang's MSc research on the morphological evolution of Lepidoptera. Subsequently, increasing requests from my friends and the availability of new datasets prompted a complete redesign of the script. I am grateful for the insightful feedback provided by Encui Wang, Qi Feng, Yurong Jiang, Zhipeng Zhao, and other users; their valuable input and patience were crucial to its completion. Special thanks are extended to Zhiping Wang for his encouragement throughout the development process. I am grateful to Pablo Goloboff for his valuable feedback and coding support, and to Juan J. Morrone for his careful consideration.

# Data availability statement

The script, guoyi.run, and external software tnt2figtree are available on GitHub (https://github.com/starsareintherose/TNT\_Script) and self-host git website (https://git.malacology.net/malacology/TNT\_Script) with MIT license.

#### References

- Farris, J.S. (1970) Methods for computing Wagner trees. *Systematic Zoology*, 19, 83–92. https://doi.org/10.1093/sysbio/19.1.83
- Fitch, W.M. (1971) Toward defining the course of evolution: Minimum change for a specific tree topology. *Systematic Zoology*, 20, 406–416.
  - https://doi.org/10.2307/2412116
- Goloboff, P.A. & Catalano, S.A. (2016) TNT version 1.5, including a full implementation of phylogenetic morphometrics. *Cladistics*, 32, 221–238.
- https://doi.org/10.1111/cla.12160
  Goloboff, P.A., Catalano, S.A. & Torres, A. (2022) Parsimony analysis of phylogenomic datasets (II): Evaluation of PAUP\*, MEGA and MPBoot. *Cladistics*, 38, 126–146.
  https://doi.org/10.1111/cla.12476
- Goloboff, P.A., Farris, J.S. & Nixon, K.C. (2008) TNT, a free program for phylogenetic analysis. *Cladistics*, 24, 774–786. https://doi.org/10.1111/j.1096-0031.2008.00217.x
- Goloboff, P.A. & Morales, M.E. (2020) A phylogenetic C interpreter for TNT. *Bioinformatics*, 36, 3988–3995. https://doi.org/10.1093/bioinformatics/btaa214
- Goloboff, P.A. & Morales, M.E. (2023) TNT version 1.6, with a graphical interface for MacOS and Linux, including new routines in parallel. *Cladistics*, 39, 144–153. https://doi.org/10.1111/cla.12524
- Goloboff, P.A., Torres, A. & Arias, J.S. (2018) Weighted parsimony outperforms other methods of phylogenetic inference under models appropriate for morphology. *Cladistics*, 34, 407–437. https://doi.org/10.1111/cla.12205
- Kluge, A.G. & Farris, J.S. (1969) Quantitative phyletics and the evolution of anurans. *Systematic Zoology*, 18, 1–32. https://doi.org/10.1093/sysbio/18.1.1
- Kolaczkowski, B. & Thornton, J.W. (2004) Performance of maximum parsimony and likelihood phylogenetics when evolution is heterogeneous. *Nature*, 431, 980–984. https://doi.org/10.1038/nature02917
- Swofford, D.L. (2003) *PAUP\**. *Phylogenetic Analysis Using Parsimony (\*and Other Methods)*. *Version 4*. Sinauer Associates, Sunderland, Massachusetts. [program]
- Torres, A., Goloboff, P.A. & Catalano, S.A. (2022) Parsimony analysis of phylogenomic datasets (I): Scripts and guidelines for using TNT (Tree Analysis using New Technology). *Cladistics*, 38, 103–125. https://doi.org/10.1111/cla.12477
- Wheeler, W.C., Coddington, J.A., Crowley, L.M., Dimitrov, D., Goloboff, P.A., Griswold, C.E., Hormiga, G., Prendini, L., Ramírez, M.J., Sierwald, P., Almeida-Silva, L., Alvarez-Padilla, F., Arnedo, M.A., Benavides Silva, L.R., Benjamin, S.P., Bond, J.E., Grismado, C.J., Hasan, E., Hedin, M., Izquierdo, M.A., Labarque, F.M., Ledford, J., Lopardo, L., Maddison, W.P., Miller, J.A., Piacentini, L.N., Platnick, N.I., Polotow, D., Silva-Dávila, D., Scharff, N., Szűts, T., Ubick, D., Vink, C.J., Wood, H.M. & Zhang, J. (2017) The spider tree of life: Phylogeny of Araneae based on target-gene analyses from an extensive taxon sampling. *Cladistics*, 33, 574–616. https://doi.org/10.1111/cla.12182
- Zhang, G., Ristola, P., Su, H., Kumar, B., Zhang, B., Hu, Y., Elliot, M.G., Drobot, V., Zhu, J., Staal, J., Larralde, M., Wang, S., Yi, Y. & Yu, H. (2025) BioArchLinux: community-driven fresh reproducible software repository for life science. *Bioinformatics*, 41, btaf106. https://doi.org/10.1002/hisinformatics/htsf106

https://doi.org/10.1093/bioinformatics/btaf106