



MEGA 12.1: Cross-Platform Release for macOS and Linux Operating Systems

Glen Stecher¹ · Michael Suleski¹ · Qiqing Tao¹ · Koichiro Tamura^{3,4} · Sudhir Kumar^{1,2} 

Received: 23 October 2025 / Accepted: 30 October 2025 / Published online: 17 November 2025
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2025

Abstract

The Molecular Evolutionary Genetics Analysis (MEGA) software is widely used for molecular evolutionary and phylogenetic analyses. We present *MEGA* version 12.1, a cross-platform release that operates natively on macOS (Intel and Apple M-series processors) and modern Linux distributions. This version incorporates all the methodological and computational improvements of *MEGA* 12 for Microsoft Windows, including techniques that markedly reduce computational time during maximum likelihood (ML) analyses. These features include a filtered best-fit ML model test that bypasses evaluating derivative models unlikely to be optimal, an adaptive bootstrap test of phylogeny that automatically determines the necessary number of replicates, and fine-grained parallelization of ML algorithms for better multi-core performance. *MEGA* 12.1 has an enhanced graphical user interface, supporting high-resolution displays and improving analysis progress reporting and result visualization. A significant addition in *MEGA* 12.1 is an improved *Calibration Editor* that integrates seamlessly with the *TimeTree* database of molecular divergence times for easy retrieval of calibration points for molecular dating. This version also supports full cross-platform session file compatibility, allowing seamless sharing of analysis sessions across macOS, Linux, and Windows. These updates enhance accessibility, computational efficiency, and usability of *MEGA* across diverse computing environments. *MEGA* 12.1 is available for free at <https://www.megasoftware.net>.

Keywords Phylogenomics · Software · Molecular evolution · Cross-platform · MacOS · Linux · TimeTree

Introduction

The Molecular Evolutionary Genetics Analysis (MEGA) software provides an integrated suite of tools for analyzing DNA and protein sequences from an evolutionary

perspective. Since its first release over three decades ago, *MEGA* has continuously expanded in scope to include diverse methods for sequence alignment, substitution model selection, phylogenetic inference, divergence time estimation, and ancestral state reconstruction (Kumar et al. 1994, 2024). Over time, its graphical and command-line interfaces have been re-engineered and re-imagined to leverage advances in computational power better and keep pace with evolving research needs (Caspermeyer 2018; Kumar et al. 2024).

The 12th major version of *MEGA*, initially released for Microsoft Windows, introduced significant improvements in computational efficiency and user experience (Kumar et al. 2024). These include adaptive heuristics that accelerate maximum likelihood (ML) analyses by skipping unnecessary evaluations during substitution model selection and dynamically determining the number of bootstrap replicates needed to achieve stable phylogenetic support values. In addition, *MEGA* 12 implemented additional fine-grained parallelization of ML algorithms to better utilize multi-core processors, substantially reducing runtime. Other advances

Handling Editor: David Liberles

✉ Koichiro Tamura
ktamura@tmu.ac.jp

✉ Sudhir Kumar
s.kumar@temple.edu

¹ Institute for Genomics and Evolutionary Medicine, Temple University, Philadelphia, PA 19122, USA

² Department of Biology, Temple University, Philadelphia, PA 19122, USA

³ Department of Biological Sciences, Tokyo Metropolitan University, Tokyo, Japan

⁴ Research Center for Genomics and Bioinformatics, Tokyo Metropolitan University, Tokyo, Japan

included enhanced high-resolution interface support, improved *Tree Explorer*, and the integration of *DrPhylo*, an external tool for evaluating clade stability and identifying gene-species combinations that may distort phylogenetic inference (Sharma and Kumar 2024; Kumar et al. 2024).

Previously, these computational and methodological advances were only available in the Windows edition of *MEGA 12*, while users of macOS and Linux systems relied on earlier cross-platform versions (Kumar et al. 2018; Stecher et al. 2020; Tamura et al. 2021). The release of *MEGA* version 12.1 addresses this limitation by providing a fully cross-platform edition that natively supports macOS (including Intel and Apple M-series processors) and modern Linux distributions. *MEGA 12.1* incorporates all enhancements introduced in *MEGA 12* for Windows noted above. In addition, *MEGA 12.1* introduces cross-platform session file compatibility for seamless sharing of analysis sessions among Windows, macOS, and Linux users for collaborative research. Another significant addition is the updated *Calibration Editor* that now connects directly to the *TimeTree* database (Kumar et al. 2022) to retrieve secondary calibrations for divergence time estimation. These developments ensure that *MEGA*'s advanced analytical capabilities and usability are accessible to all researchers, independent of their computing environment, thereby promoting reproducibility and inclusivity in molecular evolutionary and phylogenetic studies.

Results

Cross-Platform Development for macOS

The release of *MEGA X* marked the first natively compiled version of *MEGA* for Apple computers, a platform popular among many biologists and bioinformaticians (Stecher et al. 2020). That milestone introduced a single-source, cross-compiled architecture using the Lazarus/Free Pascal development environment, enabling identical core functionality across platforms. *MEGA 11* extended these capabilities and analytical methods and provided full graphical parity across platforms (Tamura et al. 2021). Together, these macOS versions have been downloaded more than 370,000 times for research and education, which underscores the importance of maintaining native cross-platform support for newer versions of *MEGA*.

Producing a robust and reliable macOS edition of *MEGA 12.1* after *MEGA 11* involved several significant challenges. Because Mac computers with both Intel processors and Apple's M-series SoCs (System on a Chip) are widely used, it was necessary to generate universal (dual-binary) application bundles that run natively on both x86_64 and

ARM64 architectures. This required compiling all *MEGA* suite components (GUI, CC, and browser subprocesses) for both architectures, then fusing them into a single Mach-O file containing slices for each CPU type. The same process had to be applied to all third-party binaries used in *MEGA*, including the *MUSCLE* alignment system and *DrPhylo*, as well as the shared libraries that support the Chromium Embedded Framework (CEF) integrated web browser. Furthermore, all binary files in the application bundle required code signing, notarization, and stapling to meet Apple's macOS security requirements.

Another major challenge was resolving incompatibilities between the Lazarus widgetset and Apple's Cocoa API, which is an issue well recognized by application developers. Many visual components in the Lazarus Component Library exhibit inconsistent behavior across operating systems, particularly on macOS. These idiosyncrasies included differences in event lifecycles, where variations in the timing and ordering of widgetset actions occasionally led to random crashes that were difficult to reproduce and debug. Additionally, random crashes involving modal dialogs were eliminated by deferring their display using queued procedure calls in the main application loop. This approach ensures that dialog execution occurs only after the current event stack has completed, thereby avoiding re-entrant run-loop issues in the Cocoa framework.

Collectively, these refactoring efforts resolved longstanding sources of instability from earlier releases, resulting in a more stable, responsive, and consistent experience for *MEGA* macOS users.

Cross-Platform Development for Linux

Linux is also widely used in computational biology, particularly in high-performance and cloud environments. Native Linux editions of *MEGA* have been downloaded more than 120,000 times to date, motivating the porting of *MEGA 12*. This required major architectural modernization of the integrated web browser, which relies on CEF. It handles importing online data and displaying results. Legacy fpCEF3 bindings to CEF were replaced with the newer CEF4Delphi interface, enabling full HTML5 rendering, improved security, and long-term maintainability. The transition introduced several platform-specific challenges, including conflicts between the GTK widget set and CEF's multi-process architecture. These issues were resolved by offloading CEF asynchronous multi-process initialization to a dedicated non-GUI process. GPU-accelerated rendering also caused instability on some Linux distributions, so the rendering pipeline was re-implemented with selective GPU disabling and custom keyboard focus management. Consequently, many options dialog boxes, result viewers,

and embedded browser panels now operate more smoothly and consistently in *MEGA* 12.1, which is now distributed as native packages for Debian/Ubuntu (.deb) and Fedora/Red Hat (.rpm) systems with glibc 2.34 or later.

Performance Enhancements for Maximum Likelihood (ML) Analyses

MEGA 12 originally introduced two key optimizations for ML-based inference on Windows: Filtered Best-Fit ML Model Selection and Adaptive Bootstrapping (Kumar et al. 2024). The former heuristically skips sub-optimal derivative models, reducing runtime without sacrificing accuracy. The latter dynamically determines the number of bootstrap replicates required to generate stable bootstrap support values for all clades. In *MEGA* 12.1, these algorithms have been ported and extensively stress-tested for macOS and Linux. Platform-specific updates included custom thread-synchronization logic to optimize multi-core computing and ensure thread safety on macOS. However, *MEGA* is not yet programmed for HPC environments that use multiple compute nodes. *MEGA* uses fine and coarse-grained parallelization, but only among the cores/processors available on a single computer.

Enhanced Calibration Editor

Clock calibrations are necessary to construct molecular timetrees calibrated to absolute time (Kumar and Hedges 2016), yet identifying molecular clock calibrations can be tedious and challenging (Mello 2018). *MEGA* 12.1 introduces a redesigned and expanded *Calibration Editor* with direct integration to the *TimeTree* database (*TTDB*) via its RESTful API (Kumar et al. 2022). *TTDB* stores and synthesizes molecular timetrees from thousands of published articles encompassing more than 140,000 species. Users can now map taxa in their phylogeny to NCBI Taxonomy identifiers, query *TTDB* for the most recent common ancestor (MRCA) of those taxa, and automatically retrieve median (molecular) divergence-time estimates with confidence intervals (Fig. 1). The workflow allows flexible keyword or ID-based searches, interactive node selection, and real-time feedback within the *MEGA* GUI. Median time estimates are based on study times that are topologically consistent with the synthetic timetree of life (TT5), and confidence intervals (or ranges) are generated when divergence-time estimates of the MRCA are available in multiple studies. Notably, using different subsets of taxa within a clade can alter the MRCA provided by *TTDB* due to discordance between the user's tree and TT5 topology. The new system also supports calibration-density specification using a uniform distribution in addition to minimum/maximum bounds, providing

users with greater statistical control over divergence-time estimation.

User Interface Updates

The *Calibration Editor* now features intuitive mouse interactions (right-click or double-click) for adding or modifying node calibrations, contextual tooltips, and visual indicators for calibrated nodes. Dialog boxes appear dynamically as needed, reducing clutter and guiding users through the process. By providing a structured workflow and immediate visual feedback, *MEGA* 12.1 lowers barriers for new users while improving precision for experts.

Reusable Calibrations

Calibration sets can now be saved and reused across analyses without re-running the RelTime method (Tamura et al. 2017) for divergence time estimation. This enables consistent calibration of multiple gene alignments that share the same phylogeny, facilitating cross-gene comparisons of evolutionary rates and their heterogeneity.

Cross-Platform Session File Compatibility

Earlier versions of *MEGA* restricted session-file exchange across platforms and users due to differences in floating-point representations. *MEGA* 12.1 introduces an automatic detection and conversion system that harmonizes floating-point formats during file loading. Although negligible precision loss may occur beyond the 15th decimal place, the impact on evolutionary estimates is expected to be insignificant. This feature removes a persistent barrier to collaboration among mixed-platform research teams and ensures that archived session files remain usable as computing architectures evolve.

Additional Updates in *MEGA* 12.1

Several additional features have been added to *MEGA* 12.1, some of which are noted below.

Absolute Substitution Rate Estimation

The *Estimate Site-by-Site Rates* option now computes absolute substitution rates at each site when the user-specified topology has branch lengths in absolute time units. This feature will enable site-wise evolutionary rate comparisons across genomic regions and proteins.

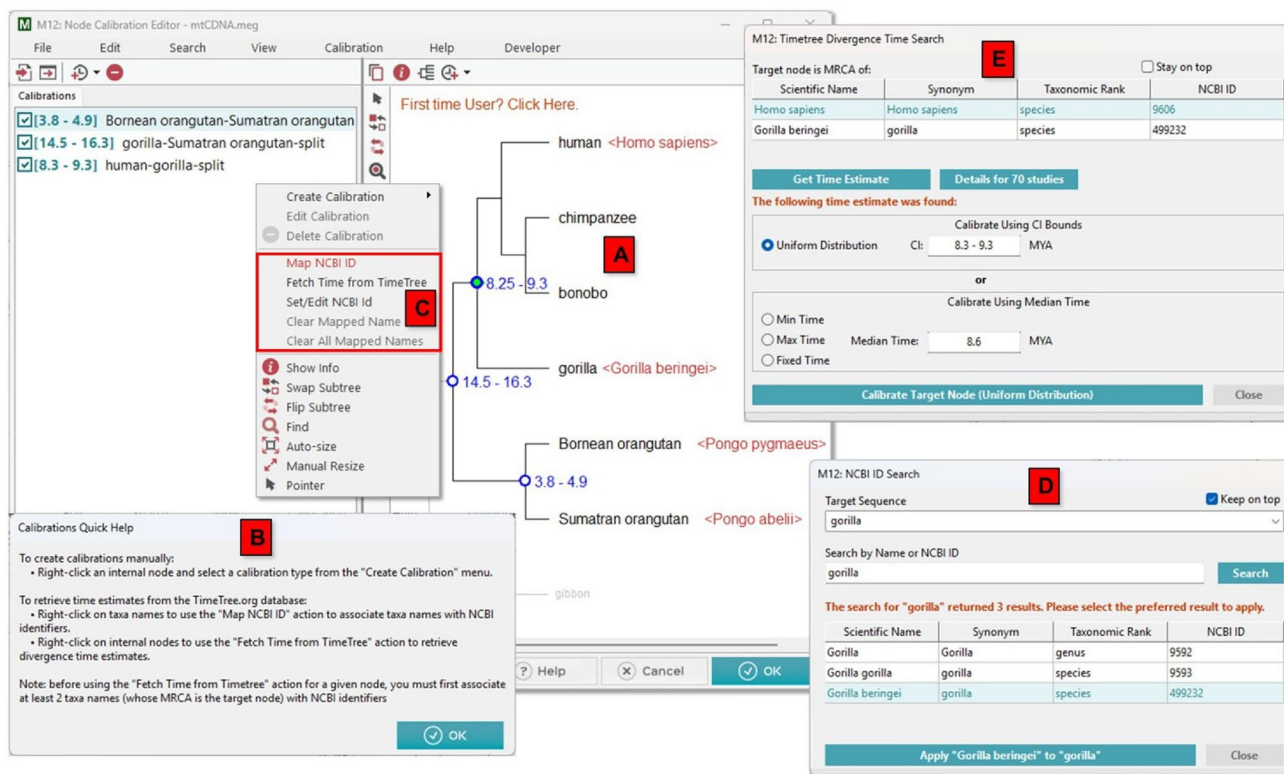


Fig. 1 Enhanced *Calibration Editor* in *MEGA 12.1*. The *Calibration Editor* (A) displays an interactive tree of great apes, with several tip names already mapped to *NCBI* IDs. Scientific names for mapped IDs are shown next to tip names in angle brackets, and applied constraints are shown next to circled nodes. The *Quick Help* dialog (B) provides brief instructions for defining calibration constraints using either the *TimeTree* RESTful API or manually. Users can right-click a tip or node in the tree to open the context menu (C) to make *TTDB* queries for the selected element. The *NCBI ID Search* dialog (D) is used to map

tip names in the tree to *NCBI* Identifiers used by *TTDB*. After at least two tip names are mapped to identifiers, time estimates for the MRCA of mapped tip names can be retrieved from *TTDB* using the *Divergence Time Search* dialog (E). Calibrations derived from the *TimeTree* resource can be applied as a uniform density distribution, a minimum time constraint, a maximum time constraint, or a fixed time. A list of studies used for deriving a *TTDB* time estimate can be retrieved by clicking the *Details for studies* button

Efficient Detection of Identical Sequences

The previous entropy-based algorithm has been replaced by a 64-bit FNV-1a non-cryptographic hash function (Wikipedia 2025), which avoids computationally intensive floating-point logarithms and eliminates ambiguity for identical-entropy sequences. This approach accelerates identity checks, which are crucial for calculating population-level mean diversity and filtering redundant sequences across datasets.

Ancestral States Export to FASTA Files

Users can now export Maximum Likelihood (Bayesian) ancestral-state reconstructions (Yang et al. 1995) as FASTA sequence alignments, with the option to specify a posterior-probability cutoff to report low-confidence reconstructions as missing data.

Improved Alignment Explorer Display

FASTA files obtained from molecular sequence databases often include all essential identifiers and additional descriptive information in the sequence headers, making them very long and unwieldy. Headers are frequently constructed with an accession number or organism name, followed by a space, and then descriptive information such as gene name and product, genomic coordinates, data type, genome build, and keywords. To facilitate working with long sequence names, the *Alignment Explorer* has been updated with a feature that allows toggling between full sequence headers and just the content before the first whitespace character by clicking the *Display-> Show Full Sequence Names* menu item.

Conclusions

MEGA 12.1 represents a significant update in the ongoing development of the *MEGA* software suite. An additional year of programming, debugging, and cross-platform testing since the release of *MEGA* 12 has produced a better, more unified codebase across macOS, Linux, and Windows. By building on all the innovations of *MEGA* 12, including adaptive and energy-efficient ML analyses and an improved user interface across major operating systems, *MEGA* 12.1 provides equal access to the global research community. The technical improvements address the need for dual-architecture support, modern browser integration, thread synchronization, and precision-safe file compatibility. These needs highlight the complexity of modern cross-platform scientific software engineering and their solution reaffirms our dedication to usability, reproducibility, and scientific accuracy. Coupled with cross-platform session portability, the upgraded *Calibration Editor*, and integration with *TTDB*'s API, *MEGA* 12.1 further advances our goals of providing powerful yet user-friendly tools for molecular evolutionary analysis across all computing environments.

Distributions

MEGA 12.1 can be downloaded from <https://www.megasoftware.net> for use on Windows, macOS, and Linux. *MEGA* 12.1 for macOS is packaged as a dual-binary that runs natively on both Intel and M-series processors. The Linux version has been tested on Ubuntu 22.04 and CentOS 7 Linux systems, both running the GNOME desktop environment. *MEGA* can be operated with a GUI that offers robust visual tools for data and result exploration, or with a command line interface (CC) that is more suitable for iterative tasks and automated workflows. The source code is available at <https://github.com/KumarMEGALab/MEGA-source-code>.

Acknowledgements We thank Beatriz Mello, Jack Craig, Sudip Sharma, and many others for testing *MEGA* 12.1 and offering suggestions to improve the manuscript.

Funding This work was supported by a research grant from the US National Institutes of Health (R35GM139540-05) and the US National Science Foundation (DBI 2505985 and 2318917).

Declarations

Conflict of interest None declared.

References

- Caspermeyer J (2018) *MEGA* software celebrates silver anniversary. *Mol Biol Evol* 35:1558–1560
- Kumar S, Hedges SB (2016) Advances in time estimation methods for molecular data. *Mol Biol Evol* 33:863–869
- Kumar S, Tamura K, Nei M (1994) *MEGA*: molecular evolutionary genetics analysis software for microcomputers. *CABIOS* 10:189–191
- Kumar S, Stecher G, Li M et al (2018) *MEGA X*: molecular evolutionary genetics analysis across computing platforms. *Mol Biol Evol* 35:1547–1549
- Kumar S, Suleski M, Craig JM et al (2022) *TimeTree 5*: an expanded resource for species divergence times. *Mol Biol Evol* 39:msac174
- Kumar S, Stecher G, Suleski M et al (2024) *MEGA12*: molecular evolutionary genetic analysis version 12 for adaptive and green computing. *Mol Biol Evol* 41:msae263
- Mello B (2018) Estimating timetrees with *MEGA* and the timetree resource. *Mol Biol Evol* 35:2334–2342
- Sharma S, Kumar S (2024) Discovering fragile clades and causal sequences in phylogenomics by evolutionary sparse learning. *Mol Biol Evol* 41:msae131
- Stecher G, Tamura K, Kumar S (2020) Molecular evolutionary genetics analysis (*MEGA*) for macOS. *Mol Biol Evol* 37:1237–1239
- Tamura K, Tao Q, Kumar S (2017) Theoretical foundation of the Rel-Time method for estimating divergence times from variable evolutionary rates. *Mol Biol Evol* 35:1770–1782
- Tamura K, Stecher G, Kumar S (2021) *MEGA11*: molecular evolutionary genetics analysis version 11. *Mol Biol Evol* 38:3022–3027
- Yang Z, Kumar S, Nei M (1995) A new method of inference of ancestral nucleotide and amino acid sequences. *Genetics* 141(4):1641–1650
- Wikipedia (2025) Fowler–Noll–Vo hash function. In: Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Fowler%E2%80%93Noll%E2%80%93Vo_hash_function

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.