

Protocols

Brief Communication

**MEGA-GPT: Artificial Intelligence Guidance and Building Analytical Protocols using
MEGA software**

John B. Allard^{1,2} and Sudhir Kumar^{*,1,2}

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

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

*Corresponding author (s.kumar@temple.edu)

Abstract

Over the past three decades, the Molecular Evolutionary Genetics Analysis (MEGA) software has evolved into a powerful tool with an ever-expanding suite of functionalities. Yet, despite its user-friendly design and widespread adoption by researchers and students, the software's extensive feature set can overwhelm new and experienced users unfamiliar with its latest capabilities. To address this challenge, we developed MEGA-GPT, an AI-driven resource that leverages ChatGPT augmented with retrieval techniques to guide users through MEGA's analytical workflows via natural language queries. By integrating MEGA's help documentation, version-specific articles, and other key publications, MEGA-GPT enhances ChatGPT's standard responses to deliver step-by-step protocols, clarify analytical settings, and recommend optimal workflows. Our evaluations indicate that MEGA-GPT offers significantly improved guidance while minimizing the hallucinations and inaccuracies observed in standard ChatGPT outputs. We propose that such customized, retrieval-augmented query interfaces can substantially enhance the usability of complex scientific computing packages. MEGA-GPT is freely available to all users with a ChatGPT account by accessing the URL <https://tinyurl.com/gpt-mega>, which is also integrated in MEGA's graphical user interface.

Introduction

Molecular Evolutionary Genetics Analysis (MEGA) software is widely used for evolutionary analysis in biological research (Kumar 2022). Originally developed to facilitate a few types of molecular evolutionary and phylogenetic analyses, MEGA has evolved into a comprehensive suite of models, methods, and algorithms for small-scale sequence comparisons to whole-genome studies (Caspermeyer 2018; Kumar et al. 2024). MEGA is being applied for research in diverse fields, including Virology, Bacteriology, General disease studies, Plant biology, Conservation biology, Systematics, Developmental evolution, and Population genetics (Kumar et al. 2004). It is also popular in classroom settings, where it helps students grasp the fundamentals of molecular evolution and phylogenetics (Kumar et al. 2024).

Despite its accessible graphical user interface, MEGA's extensive feature set can overwhelm new users. It can also be challenging for experienced users, especially as new methods and models are added. While various articles and books have provided guidance, such printed resources can become outdated due to MEGA's continuous development (Nei and Kumar 2000; Hall 2013; Caspermeyer 2018; Mello 2018; Kumar et al. 2024). Therefore, we have developed MEGA-GPT, a tool to give users a better understanding of MEGA's features and facilitate the development of protocols to use MEGA optimally.

Development of MEGA-GPT

MEGA-GPT is built on a Retrieval-Augmented Generation (RAG) framework that combines dynamic information retrieval with language generation (Lewis et al. 2020). It takes a hybrid approach, which is particularly effective for domain-specific applications, as it enables large language models to access up-to-date, specialized information beyond their static training data (Lewis et al. 2020). RAG reduces well-known issues such as hallucination and domain insensitivity related to using artificial intelligence (Lewis et al. 2020).

The RAG architecture comprises two key components: a retriever and a generator. The retriever scans indexed databases containing domain-specific content, from scientific articles to comprehensive technical documentation, to extract information relevant to the user's query. In the first version of MEGA-GPT, we have included MEGA help documentation, articles covering various MEGA versions since 1993, and some key articles related to MEGA and methods included therein (Kumar et al. 1994; Nei and Kumar 2000; Kumar et al. 2001; Tamura et al. 2007; Kumar et al. 2008; Tamura et al. 2011; Kumar et al. 2012; Tamura et al. 2013; Stecher et al. 2014; Caspermeyer 2018; Kumar et al. 2018; Mello 2018; Tamura et al. 2018; Stecher et al. 2020; Tao et al. 2020; Tamura et al. 2021; Kumar et al. 2024; Sharma and Kumar 2024).

Their content is retrieved in a vector representation and passed to the generator that synthesizes a response by merging the user's query vectorized representations with this authoritative data.

This two-step process enhances factual accuracy by grounding responses in current and specialized sources (Borgeaud et al. 17--23 Jul 2022; Guu et al. 13--18 Jul 2020) and reduces errors. Moreover, the RAG framework allows continuous updates to the knowledge base without requiring extensive model retraining (Lewis et al. 2020; Shuster et al. 2021). This adaptability is crucial in rapidly evolving scientific fields and for software under continuous development, ensuring that MEGA-GPT delivers guidance that reflects the latest methodologies included in MEGA.

Example Use Cases

To evaluate the effectiveness of MEGA-GPT compared to the basic version of ChatGPT (GPT-4o, OpenAI et al. 2024), we posed queries related to MEGA's functionality and assessed the usefulness of the responses. The GPT-4o did very well in many queries, but we found cases where MEGA-GPT performed much better. We describe three examples below.

Task 1: Diagnosing the Pathogenicity of a Protein Variant.

Our query was: "How can I get pathogenicity scores for variants in the CFTR gene using MEGA?" (**Fig. 1a**). GPT-4o incorrectly responded that MEGA lacks any functionality to provide pathogenicity scores (**Fig. 1b**). In contrast, MEGA-GPT correctly identified a dedicated functionality (Mutational Diagnosis) for predicting the functional impact of nonsynonymous single nucleotide variants within MEGA (Stecher et al. 2014) (**Fig. 1c**). Moreover, MEGA-GPT offered a good step-by-step guide on how to obtain pathogenicity scores for a CFTR variant, explaining that the tool integrates PolyPhen-2, SIFT, and EvoD for mutation impact assessment. It also suggested that MEGA produces CADD scores, but that was a minor inaccuracy. This

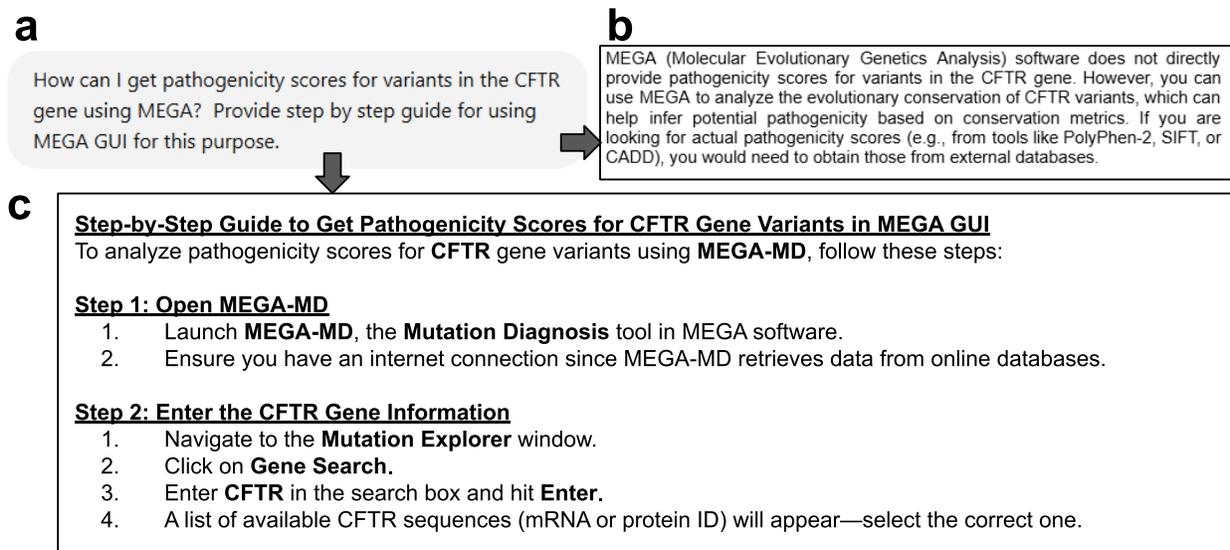


Figure 1. Comparison of outputs for Task 1. (a) The query given to both models. (b) An excerpt from the response given by GPT-4o which incorrectly claimed that MEGA cannot compute pathogenicity scores. (c) Response from MEGA-GPT. *Supplementary Figure S1* gives additional information provided by MEGA-GPT.

comparison highlights a key limitation of general-purpose AI assistants like GPT-4o: they may overlook critical features of specialized scientific software, potentially leading to misinformation that could misguide researchers.

Task 2: Building a Timetree

MEGA-GPT's response also outperforms GPT-4o's on common tasks such as generating a timetree. When asked, "How can I build a timetree from a sequence alignment in MEGA?" (**Fig. 2a**). GPT-4o provided an answer that omitted critical features and introduced significant inaccuracies (**Fig. 2b**). Notably, it failed to mention MEGA's built-in RelTime method for estimating divergence times (Tamura et al. 2018). Instead, it misleadingly recommended using BEAST, an independent software environment. It also suggested incorrect steps, such as using the nonexistent "Test Molecular Clock" option under the *Phylogeny* menu. It even stated that "Bayesian analysis" could be used to build a timetree in MEGA by linking to MrBayes, another external software tool unrelated to MEGA. These errors illustrate a broader issue of hallucination, where GPT-4o fabricates non-existent workflow steps, potentially leading users astray and frustrated when they cannot find those features.

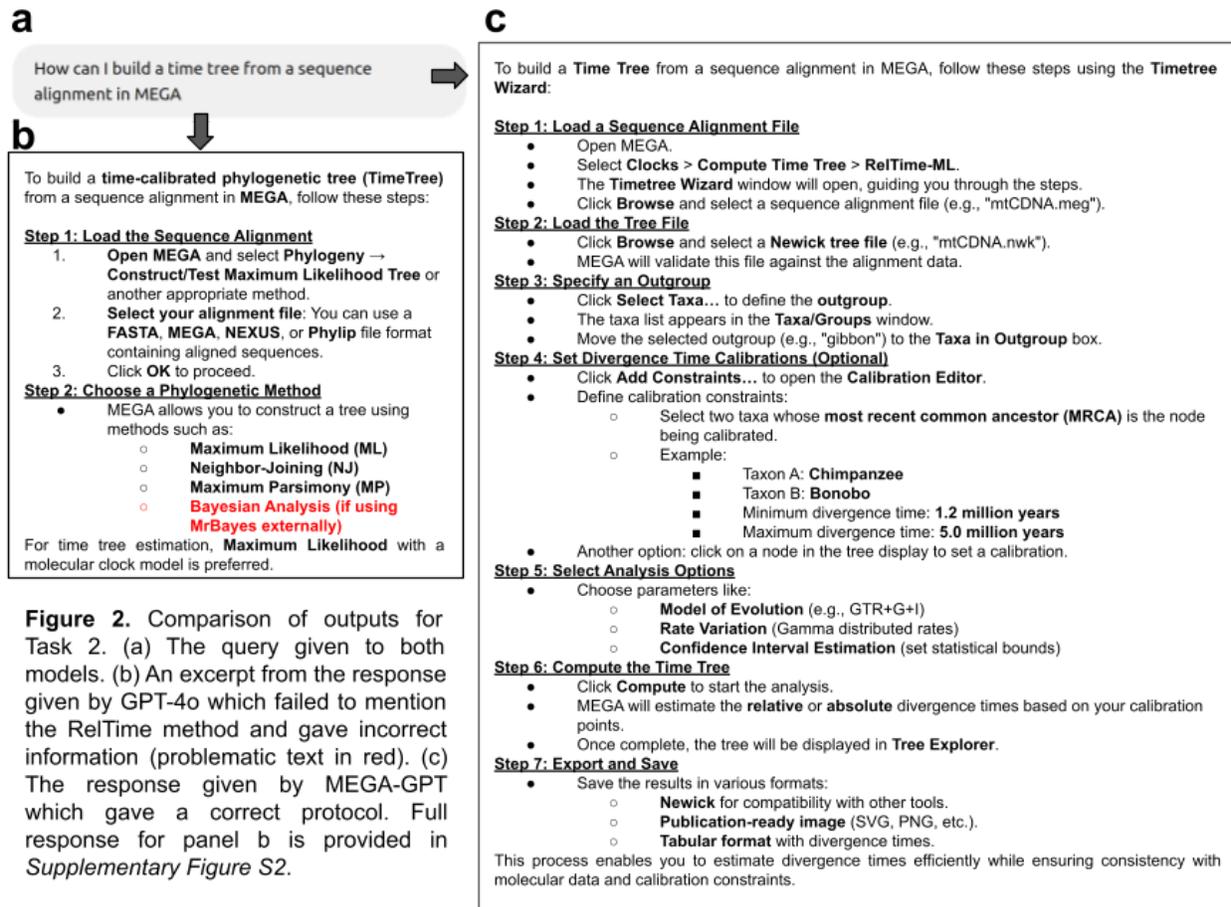


Figure 2. Comparison of outputs for Task 2. (a) The query given to both models. (b) An excerpt from the response given by GPT-4o which failed to mention the RelTime method and gave incorrect information (problematic text in red). (c) The response given by MEGA-GPT which gave a correct protocol. Full response for panel b is provided in *Supplementary Figure S2*.

In contrast, MEGA-GPT provided a detailed and accurate protocol using MEGA's Timetree Wizard (**Fig. 2c**). It correctly outlined the workflow: selecting the alignment file, loading a Newick tree, specifying an outgroup, applying divergence time calibrations, and running the RelTime method to estimate divergence times. Therefore, the RAG technology enabled the responses to focus on MEGA's features. MEGA-GPT demonstrates a clear advantage in delivering accurate, application-specific guidance.

Task 3: Testing for Recombination

An essential function of an effective AI assistant is accurately communicating the limitations of the software it supports. Users may sometimes inquire about functionalities MEGA does not offer, and the assistant must convey these constraints to avoid confusion and frustration. To assess this, we asked MEGA-GPT and GPT-4o, "How do I use MEGA to test for recombination in a nucleotide alignment?," a function for which MEGA does not include a built-in tool. GPT-4o fabricated a protocol for this non-existent feature, claiming that MEGA offers a "Recombination Identification Program (RIP)" and even provided plausible-sounding but false menu options (**Supplementary Fig. S3a**). This example highlights a common pitfall of general-purpose AI assistants—the tendency to hallucinate credible yet inaccurate information.

In contrast, MEGA-GPT correctly stated, "MEGA does not have a direct built-in recombination detection tool." While it mentioned some alternative tests that MEGA can perform, it avoided introducing misleading or fabricated protocols (**Supplementary Fig. S3b**). This accurate acknowledgment of MEGA's limitations underscores the value of a tailored, retrieval-augmented generation-based assistant in providing reliable, domain-specific guidance.

Task 4: Beyond Building Protocols

MEGA-GPT is more than a protocol provider, it serves as a versatile assistant for MEGA users by offering detailed and use-case-tailored explanations, troubleshooting guidance, and strategic recommendations tailored to specific research needs. Users can inquire about MEGA's capabilities, including available phylogenetic methods, comparisons of different evolutionary models, and supported sequence formats. For example, suppose a user is unsure which substitution model to apply. In that case, MEGA-GPT can explain the differences, such as when to choose the Tamura-Nei (1993) model over the Jukes-Cantor (1969) model or how Gamma-distributed rate variation impacts branch length estimates.

In addition to setting up analyses, MEGA-GPT assists with interpreting results. If a user encounters unexpected bootstrap support values or discrepancies in divergence time estimates, MEGA-GPT can help diagnose potential issues like missing calibration points or unsuitable outgroup selection. For researchers working with large datasets, it also suggests strategies to optimize computational efficiency, guides file format conversions, troubleshoots common errors and summarizes new features and improvements in the latest version of MEGA. While

general-purpose AI addresses some of these questions, MEGA-GPT's specialized focus on MEGA's functionalities ensures more accurate and context-specific guidance.

Conclusion and Future Directions

By integrating artificial intelligence, MEGA-GPT enhances the accessibility of MEGA for users ranging from novices to seasoned bioinformaticians. It represents another step toward making computational tools for evolutionary genetics more accessible. This would lower barriers to entry, improve efficiency, and reduce user error, empowering new and experienced users to harness MEGA's advanced capabilities more fully. By enabling users to enter into a dialog with the help documentation, which is otherwise static, MEGA-GPT allows the users to seek clarification and ask follow-up questions. These features advance beyond the traditional paradigm where text searches and lists of frequently asked questions, which MEGA-GPT automates. This also significantly enhances MEGA's application in education settings, helping to train future bioinformaticians. MEGA-GPT is accessible from the MEGA GUI alongside traditional help docs, making it readily available as an AI assistant that leverages retrieval-augmented generation to deliver better, context-aware responses.

With the development of MEGA-GPT, we see a shift in scientific software development. Many advanced computational tools can be intimidating without significant training, limiting their adoption and impact. AI-powered assistants, specifically tailored to domain-relevant knowledge, offer a promising solution to bridge this gap, especially as tools like ChatGPT become more widespread across the sciences and in the general population. While MEGA-GPT is an early forerunner in our field, we expect many more such assistants to become available soon.

Availability

MEGA-GPT is freely available to all users with a free or paid ChatGPT account at <https://chatgpt.com/g/g-RmeN18Ssp-mega-gpt> and <https://tinyurl.com/gpt-mega>. It is also accessible through the MEGA GUI interface, in the *Help* menus, and in the main MEGA window.

References

- Borgeaud S, Mensch A, Hoffmann J, Cai T, Rutherford E, Millican K, Van Den Driessche GB, Lespiau J-B, Damoc B, Clark A, et al. 17--23 Jul 2022. Improving Language Models by Retrieving from Trillions of Tokens. In: Chaudhuri K, Jegelka S, Song L, Szepesvari C, Niu G, Sabato S, editors. Proceedings of the 39th International Conference on Machine Learning. Vol. 162. Proceedings of Machine Learning Research. PMLR. p. 2206–2240.
- Caspermeyer J. 2018. MEGA software celebrates silver anniversary. *Mol. Biol. Evol.* 35:1558–1560.
- Guu K, Lee K, Tung Z, Pasupat P, Chang M. 13--18 Jul 2020. Retrieval Augmented Language Model Pre-Training. In: Iii HD, Singh A, editors. Proceedings of the 37th International Conference on Machine Learning. Vol. 119. Proceedings of Machine Learning Research. PMLR. p. 3929–3938.
- Hall B. 2013. Building phylogenetic trees from molecular data with MEGA. *Mol. Biol. Evol.* 30:1229–1235.
- Kumar S. 2022. Embracing green computing in molecular phylogenetics. 39:msac043.
- Kumar S, Nei M, Dudley J, Tamura K. 2008. MEGA: a biologist-centric software for evolutionary analysis of DNA and protein sequences. *Brief. Bioinform.* 9:299–306.
- Kumar S, Stecher G, Li M, Knyaz C, Tamura K. 2018. MEGA X: molecular evolutionary genetics analysis across computing platforms. *Mol. Biol. Evol.* 35:1547–1549.
- Kumar S, Stecher G, Peterson D, Tamura K. 2012. MEGA-CC: computing core of molecular evolutionary genetics analysis program for automated and iterative data analysis. *Bioinformatics* 28:2685–2686.
- Kumar S, Stecher G, Suleski M, Sanderford M, Sharma S, Tamura K. 2024. MEGA12: Molecular Evolutionary Genetic Analysis version 12 for adaptive and green computing. *Mol. Biol. Evol.* 41:msae263.
- Kumar S, Tamura K, Jakobsen IB, Nei M. 2001. MEGA2: molecular evolutionary genetics analysis software. *Bioinformatics* 17:1244–1245.
- Kumar S, Tamura K, Nei M. 1994. MEGA: molecular evolutionary genetics analysis software for microcomputers. *Bioinformatics* 10:189–191.
- Kumar S, Tamura K, Nei M. 2004. MEGA3: Integrated software for Molecular Evolutionary Genetics Analysis and sequence alignment. *Briefings in bioinformatics* 5:150–163.
- Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih W-T, Rocktäschel T, et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, editors. *arXiv [cs.CL]* [Internet]:9459–9474. Available from: <https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html>
- Mello B. 2018. Estimating timetrees with MEGA and the TimeTree resource. *Mol. Biol. Evol.* [Internet]. Available from: <https://academic.oup.com/mbe/article-abstract/35/9/2334/5042667>
- Nei M, Kumar S. 2000. Molecular Evolution and Phylogenetics. New York, NY: Oxford University Press
- OpenAI, Hurst A, Lerer A, Goucher AP, Perelman A, Ramesh A, Clark A, Ostrow AJ, Welihinda A, Hayes A, et al. 2024. GPT-4o System Card. *arXiv [cs.CL]* [Internet]. Available from: <http://arxiv.org/abs/2410.21276>
- Sharma S, Kumar S. 2024. Discovering fragile clades and causal sequences in phylogenomics by evolutionary sparse learning. *Mol. Biol. Evol.* [Internet] 41. Available from:

<http://dx.doi.org/10.1093/molbev/msae131>

- Shuster K, Poff S, Chen M, Kiela D, Weston J. 2021. Retrieval Augmentation Reduces Hallucination in Conversation. *arXiv [cs.CL]* [Internet]. Available from: <http://arxiv.org/abs/2104.07567>
- Stecher G, Liu L, Sanderford M, Peterson D, Tamura K, Kumar S. 2014. MEGA-MD: molecular evolutionary genetics analysis software with mutational diagnosis of amino acid variation. *Bioinformatics* 30:1305–1307.
- Stecher G, Tamura K, Kumar S. 2020. Molecular Evolutionary Genetics Analysis (MEGA) for macOS. *Mol. Biol. Evol.* 37:1237–1239.
- Tamura K, Dudley J, Nei M, Kumar S. 2007. MEGA4: Molecular Evolutionary Genetics Analysis (MEGA) software version 4.0. *Mol. Biol. Evol.* 24:1596–1599.
- Tamura K, Peterson D, Peterson N, Stecher G, Nei M, Kumar S. 2011. MEGA5: molecular evolutionary genetics analysis using maximum likelihood, evolutionary distance, and maximum parsimony methods. *Mol. Biol. Evol.* 28:2731–2739.
- Tamura K, Stecher G, Kumar S. 2021. MEGA11: Molecular Evolutionary Genetics Analysis version 11. *Mol. Biol. Evol.* 38:3022–3027.
- Tamura K, Stecher G, Peterson D, Filipski A, Kumar S. 2013. MEGA6: Molecular Evolutionary Genetics Analysis version 6.0. *Mol. Biol. Evol.* 30:2725–2729.
- Tamura K, Tao Q, Kumar S. 2018. Theoretical foundation of the RelTime method for estimating divergence times from variable evolutionary rates. *Mol. Biol. Evol.* 35:1770–1782.
- Tao Q, Tamura K, Kumar S. 2020. Efficient methods for dating evolutionary divergences. In: *The Molecular Evolutionary Clock*. Cham: Springer International Publishing. p. 197–219.

Supplementary Figures

Supplementary Figure S1. Complete response from MEGA-GUI for Task 1.

Step-by-Step Guide to Get Pathogenicity Scores for CFTR Gene Variants in MEGA GUI

To analyze pathogenicity scores for **CFTR** gene variants using **MEGA-MD**, follow these steps:

Step 1: Open MEGA-MD

1. Launch **MEGA-MD**, the **Mutation Diagnosis** tool in MEGA software.
2. Ensure you have an internet connection since MEGA-MD retrieves data from online databases.

Step 2: Enter the CFTR Gene Information

1. Navigate to the **Mutation Explorer** window.
2. Click on **Gene Search**.
3. Enter **CFTR** in the search box and hit **Enter**.
4. A list of available CFTR sequences (mRNA or protein ID) will appear—select the correct one.

Step 3: Specify the Variants

You can specify variants in two ways:

Option 1: Manually Enter Variants

- Click on the **Sequence Data Explorer**.
- Navigate to the amino acid position of interest.
- Select the **mutant allele** (e.g., F508del, G551D).
- Click **Diagnose Variant**.

Option 2: Upload a Variant List

Create a **text file** with variant details in the format:

```
NP_000492 508 F
NP_000492 551 G
```

- Here, **NP_000492** is the **RefSeq protein ID**, **508** is the **amino acid position**, and **F** is the **mutant allele**.
- Go to **File** → **Import Query Data From File** and select your prepared file.
- MEGA-MD will validate the file and retrieve results.

Step 4: Retrieve Pathogenicity Scores

1. MEGA-MD queries the **MEGA-MDW** (a web-based variant diagnosis server).
2. The results will be displayed in the **Predictions Tab** with:
 - **Mutation** details
 - **Predicted impact score** (PolyPhen-2, SIFT, EvoD)
 - **Evolutionary context**
3. You can sort, search, and export results using the provided options.

Step 5: Explore Ancestral Information (Optional)

- Click **Explore Ancestors** to infer ancestral states of the CFTR variant.
- Choose **Maximum Likelihood (ML)** or **Maximum Parsimony (MP)** to estimate its evolutionary history.

Step 6: Save and Export Results

- Click **Export Table to CSV** or **Export Table to Excel** to save your analysis.
- You can also generate a **detailed view** for each variant.

Step 7: (Alternative) Use the MEGA-MDW Web Interface

If analyzing many variants, use **MEGA-MDW** directly:

1. Go to **www.mypeg.info/evod**.
2. Upload the same **variant list** file.
3. Wait for an email with the pathogenicity scores.

Supplementary Figure S2. Complete response from ChatGPT-4o for Task 2.

To build a **time-calibrated phylogenetic tree (TimeTree)** from a sequence alignment in **MEGA**, follow these steps:

Step 1: Load the Sequence Alignment

1. **Open MEGA** and select **Phylogeny** → **Construct/Test Maximum Likelihood Tree** or another appropriate method.
2. **Select your alignment file**: You can use a **FASTA**, **MEGA**, **NEXUS**, or **Phylip** file format containing aligned sequences.
3. Click **OK** to proceed.

Step 2: Choose a Phylogenetic Method

- MEGA allows you to construct a tree using methods such as:
 - **Maximum Likelihood (ML)**
 - **Neighbor-Joining (NJ)**
 - **Maximum Parsimony (MP)**
 - **Bayesian Analysis (if using MrBayes externally)**

For time tree estimation, **Maximum Likelihood** with a molecular clock model is preferred.

Step 3: Convert the Tree to a TimeTree

1. **Enforce a Molecular Clock**:
 - After constructing the tree, select **"Test Molecular Clock"** under the **Phylogeny** menu.
 - If the clock is rejected, you may need to use a relaxed clock method in external software like **BEAST**.
2. **Calibrate the Tree**:
 - Select **"Compute TimeTree"** in MEGA.
 - Specify **calibration points**, which can be based on **fossil records** or **secondary calibrations** (e.g., divergence times from prior literature).
 - If available, enter **known divergence times** for one or more nodes.
3. **Run the TimeTree Analysis**:
 - Choose an **evolutionary model** (e.g., **Tamura-Nei, GTR**).
 - Adjust the **rate variation model** (e.g., Gamma distribution).
 - Click **Compute** to generate a time tree.

Step 4: Interpret and Save the Tree

- Once the tree is computed:
 - View branch lengths **scaled to time**.
 - Export the tree in **Newick** or **MEGA format** for further analysis.
 - Visualize divergence times and confidence intervals.

Alternative: Use MEGA for Sequence Alignment, Export to BEAST for TimeTree

If you need more advanced time-tree models (e.g., relaxed clock), consider:

1. **Align sequences in MEGA**.
2. **Export the alignment in Nexus format**.
3. **Use BEAST2** to construct a Bayesian TimeTree.
4. Would you like detailed steps on using **MEGA with BEAST** for better time calibration?

Supplementary Figure S3. Comparison of outputs for Task 3. (a) The query given to both models. (b) An excerpt from the response given by GPT-4o, which fabricated an imaginary tool called “Recombination Identification Program,” and an imaginary protocol to go with it. Incorrect text is highlighted in red. (c) The response given by MEGA-GPT correctly notes that MEGA does not have a built-in tool to test for recombination (highlighted in blue), but it suggests some tests that MEGA does provide that may be of interest to the user.

a

How do I use MEGA to test for recombination in a nucleotide alignment

b

To test for recombination in a nucleotide alignment using MEGA, follow these steps:

Step 1: Prepare Your Alignment

- Ensure that your sequences are **properly aligned**. You can use MEGA's **Alignment Explorer** to align sequences using ClustalW or MUSCLE.
- Save your alignment in MEGA format (.meg) or another supported format such as FASTA.

Step 2: Perform Recombination Analysis

MEGA does not have a direct built-in recombination detection tool, but it provides **tests of selection and neutrality**, which can indicate potential recombination events. You can use Tajima's D test and related neutrality tests:

...

c

You can use MEGA (Molecular Evolutionary Genetics Analysis) to test for recombination in a nucleotide alignment using **RIP (Recombination Identification Program)**, which is included in MEGA. Here's how you can do it:

Steps to Test for Recombination in MEGA

1. **Open MEGA and Load Your Alignment**
 - Launch **MEGA** and go to **File** → **Open a File/Session**.
 - Select your **nucleotide alignment** file (in **FASTA, MEGA, or Phylip** format).
 - Ensure the sequences are aligned properly before proceeding.
2. **Run the Recombination Detection Test (RIP)**
 - Go to **Tools** → **Recombination Detection (RIP)**.
 - In the **RIP Analysis Setup**, you will see options to define the window size, step size, and other parameters.
 - Adjust the **window size** (e.g., 200 bp) and **step size** (e.g., 20 bp) based on your alignment length.
 - Choose a **substitution model** for the analysis (e.g., Tamura-Nei).
3. **Run the Analysis**
 - Click **Compute** to start the recombination detection analysis.
 - MEGA will scan for recombination breakpoints across the alignment.
4. **Interpret the Results**
 - The results will show a **graphical representation** of recombination signals across the alignment.
 - Look for regions where genetic distance patterns shift abruptly, indicating potential recombination breakpoints.
 - If MEGA detects significant recombination events, it will suggest potential breakpoints...