Parametric bootstrap under maximum likelihood

Mattia Giacomelli and Davide Pisani

Parametric bootstrap is a statistical method that can be used to evaluate the performance of a model in describing the data (= testing model adequacy) under **Maximum Likelihood.**

Parametric bootstrap should not be confused with standard (non-parametric) bootstrap, which is used to estimate support for clades in a tree. To perform a parametric bootstrap analysis, datasets are simulated (of the same size of the original dataset) under the model used to analyse the data. It is fundamental that the model used to simulate the data is parametrised exactly as the one used to analyse the data. A defined number of datasets is simulated (e.g. 100 datasets), and a statistic of interest is measured from each simulated dataset to generate a distribution of values. After that, the same statistic is calculated for the original dataset. If the values observed for the original data fall within the distribution of the simulated datasets, the model is said to adequately describe (or fit) the data. If the value calculated for the original data does not fall within the distribution of values generated from the simulated data, the model does not adequately describe the data. Deviations of real values from the average of the distribution is usually expressed using standard deviates (Z-scores). Phylogenetic model adequacy is more frequently tested in a Bayesian framework, using Posterior Predictive Analysis (Lartillot et al., 2007; Feuda et al, 2017; Giacomelli et al., 2022), the Bayesian equivalent of Parametric Bootstrap. However, given the current tendency for the development of complex models in Maximum Likelihood (e.g. the CAT-PMSF models; Szánthó et al., 2023), it is important that we devise tools to test the fit of models that are implemented in a Maximum Likelihood framework, which is what this tutorial aims to do.

Posterior predictive analysis and Parametric Bootstrap are implemented through a series of steps that are conceptually correspondent.

In this tutorial we describe the steps necessary to complete a parametric bootstrap analysis and test model adequacy under maximum likelihood. All the relevant data and scripts to perform this tutorial can be found here: https://github.com/mgiacom/tardigrades_catpmsf/tree/main/tutorial/data. We use IQ-TREE 2 (Mihn et al., 2020) and AliSim (Ly-Trong et al., 2022, Ly-Trong et al., 2023), as well as custom python scripts to first simulate, and then to compare simulated and real datasets. As an example, we use the dataset built to illustrate the long branch behaviour of nematodes (nematode.fasta) by Lartillot (2007). As shown in previous studies, this dataset is appropriate to investigate the performance of different models as it returns different trees when analysed using different models, and there is general agreement in the community that Nematoda are members of Ecdysozoa. Hence, the ability of different models to recover the correct placement for the nematodes can be tested unambiguously.

We perform simulations under four models: LG+G4, Poisson+C10+G4, LG+C10+G4 and LG-CAT-PMSF+G4. LG+G4 is an empirical across-site compositionally homogeneous

model, while Poisson+C10+G4, LG+C10+G4 and LG+CAT-PMSF+G4 are across-site compositionally heterogeneous, mixture models (see Szánthó et al., 2023 for all the details).

Step 1. Infer the model parameters:

The first step is parametrising the considered model using the real data. For this step one could use a fixed topology, or infer the tree topology during tree search. Here we decided to use the second approach to be able to compare the trees inferred using the four considered models.

For LG+G4, Poisson-C10+G4 and LG-C10+G4, we can achieve this by simply running a standard phylogenetic analysis in IQ-TREE 2, specifying the model we need (you can run this analysis in parallel using the *-T AUTO* flag, or similar, depending on the version of IQ-TREE used).

Lines of code are progressively numbered.

1.1: *iqtree2 -s nematode.fasta -m LG+F+G4 -bb 1000 -pre LG*

1.2: *iqtree2 -s nematode.fasta -m C10+F+G4 -bb 1000 -pre Poisson-C10*

1.3: *iqtree2 -s nematode.fasta -m LG+C10+F+G4 -bb 1000 -pre LG+C10*

For LG+CAT-PMSF+G4, the process is more complex. First we need the site frequencies which are estimated using Phylobayes (Lartillot et al., 2004). Here we used the site frequencies from the study of Szantho et al. (2023) – *lg_nematode_icc_chain1.sitefreq,* which we downloaded from https://github.com/drenal/cat-PMSF-paper/tree/main/step3_iqtree/nematode/IQTree_LG-CAT-PMSF . The frequencies in *lg_nematode_icc_chain1.sitefreq* had been previously inferred by Szantho et al. (2023) using Phylobayes.

In the IQ-Tree analysis, for the LG+CAT-PMSF+G4 run, we have to add the flag *-wsr* to generate an output file with the site-rates (.rate) under the Gamma model, which will be needed to simulate the data in step 2.

The complete IQ-Tree command is:

1.4: *iqtree -s nematode.fasta -m LG+G4 -fs lg_nematode_icc_chain1.sitefreq -wsr -bb 1000 -pre LG_CAT_PMSF*

Once the analyses are finished we can compare the topologies found by the four models. It's interesting to note that the site-homogeneous models (LG) fail to recover the currently accepted (Ecdysozoa) topology, finding instead the nematodes sister to (arthropoda+chordata – UFB= 85). The three mixture models, differently, recover Ecdysozoa (UFB= 65 - LG-C10+G4; UFB=98 - Poisson-C10+G4; UFB= 100 - LG+CAT-PMSF+G4).

Step 2. Simulate the datasets.

Step 2.1. Generation of simulated datasets:

In the second step, datasets are simulated using AliSim (Ly-Trong et al., 2022, Ly-Trong et al., 2023), which is part of IQ-TREE.

Simulated datasets are generated using the specified evolutionary models (in our case LG+G4, Poisson-C10+G4, LG-C10+G4, LG-CAT-PMSF+G4). Values for the model parameters are those that were estimated from the original dataset in Step 1 – this includes the tree topology and branch lengths. For example, for the LG+G4 simulations, we used the tree topology, AA Frequencies, and Gamma parameter (alpha) inferred from the original data in Step 1 (Line of code 1.1; output file *.iqtree and *treefile).

We specify AliSim to simulate 100 amino acid datasets in fasta format. We include the *--length* flag to simulate replicates of the same size of the real dataset (35,371 AA positions).

The complete line of code to simulate datasets using LG+G4 is:

2.1: *iqtree --alisim LG --seqtype AA -m LG+F{20your_freq}+G4{your_gamma} -t LG.treefile --length 35371 --out-format fasta --num-alignments 100*

As you can see, when specifying the model, we must include the empirical frequencies estimated from the dataset, as well as the gamma parameter (alpha), and your tree (with branch lengths). You can find this information in the *.iqtree* and *.treefile* output files from line of code 1.1.

To simulate under standard precomputed mixture models (Poisson-C10+G4 and LG-C10+G4), a couple of extra steps need to be completed first. This is because the current implementation of Alisim does not allow to include the empirical AA frequencies directly in the command line for these models. To solve this problem, we need to generate a nexus file where we specify them as a component (in this case the 11th component) of the mixture. As above, we can find the frequencies, and their relative weights under the mixture, in the *.iqtree* file generated in Line of code 1.2 and 1.3.

The model is then imported in Alisim with the flag *-mdef,* as in the lines of code below:

2.2: *iqtree --alisim Poisson_C10 --seqtype AA -m new_poisson_c10 -mdef my_c10_model.nex -t C10.treefile --length 35371 --out-format fasta --num-alignments 100*

2.3: *iqtree --alisim LG_C10 --seqtype AA -m new_lg_c10 -mdef my_lg_c10_model.nex -t LG_C10.treefile --length 35371 --out-format fasta --num-alignments 100*

To generate simulated data under PMSF (in our case LG-CAT-PMSF+G4) requires one extra step. Since each site has its own frequency profile, as well as its own substitution rate, we simulate every site as a separate partition. To achieve this goal we generate a nexus file with site-specific parameters. This nexus file is large and cannot be simply generated by hand (as we did for the simulations under Poisson-C10+G4 and LG-C10+G4. Instead we use a

Python script that takes two arguments: the *.sitefreq* file and the *.rate* files. The *.sitefreq* file is the one generated using Phylobayes and used in Analysis 1.4. The *.rate* file is an outcome of line of code 1.4. You can find the script in the tutorial repository https://github.com/mgiacom/tardigrades_catpmsf/tree/main/tutorial/data,'create_nexus_PMSF_simulation.py' and run it as below.

2.4: *python create_nexus_PMSF_simulation.py lg_nematode_icc_chain1.sitefreq LG_CAT_PMSF.rate > LG_CAT_PMSF.nex*

After having generated the nexus file, we can simulate the datasets using Alisim, specifying the nexus with the *-p* flag.

2.5: *iqtree --alisim LG_CAT-PMSF --seqtype AA -p LG_CAT_PMSF.nex -t LG_CAT_PMSF.contree --length 35371 --out-format fasta --num-alignments 100*

Step 2.2. Adding gaps to the simulated datasets:

Gaps have an effect on the scores of model adequacy tests. Alisim does not simulate gaps. To obtain comparable results It is key to add gaps to the simulated datasets at the same sites where they are observed in the real dataset.

To add gaps to the simulated datasets obtained using Alisim (the *.fa* files), we use a custom Python script: 'transfer_gaps.py'. This script takes 3 arguments: real data (in this case *nematode.fasta*), a simulated dataset (e.g. *LG_1.fa*) and the name of the output file (e.g. *LG_1.fa_gaps*).

Assuming all our simulated datasets are in the same directory, we can use a for loop to add gaps to the simulated data (in the same positions in which they are found in the real data).

2.6: *for i in *fa; do python transfer_gaps.py nematode.fasta $i ${i}_gaps; done*

Code line 2.6 will have to be repeated in each of the four directories where we have simulated data. At the end of this final step we will have generated 400 simulated datasets (100 for each model). The name of each of these datasets will end in *fa_gaps*.

Step 3. Compare Simulated and real data

We are now ready to calculate the fit of the considered models. As explained in the introduction of this tutorial, to evaluate the absolute fit of the model to the data, we measure a statistic of interest (representing a property of the data we are interested in), across the simulated datasets and for the real dataset. After that, we test whether the value estimated for the real data falls within the distribution of values measured on the simulated data. If a model is adequately describing the data, datasets simulated using that model should be undistinguishable (with reference to the statistics of interest) from the real data. That is, the value estimated for the real data is expected to fall within the distribution of the simulated data. If the model is not an adequate descriptor of the data, the value measured for the real

data is expected to fall outside of the distribution of values generated from the simulated data. To evaluate if a model fit or not the data, and estimate the extent of the misspecification when a model does not fit the data, we can either graphically visualise the distribution of simulated values and identify where the value calculated for the real data maps with reference to the distribution of simulated data, or we can express the distance from the observed value from the mean of the distribution of simulated values using standard deviates (expressed as Z-scores). The first approach is more robust as the distribution of simulated values might not be normal. The second approach (using Z-scores) is less robust as it assumes that the distribution of simulated values is normal. Z-scores have the advantage that allow for a simpler quantification of the results without the need of a figure and while Z-scores might not be ideal they represent a valid approximation. It should however be considered good practice to also visualise the results of parametric bootstrap analyses to make sure that, given the distribution of simulated values, the Z-scores are not misleading.

In the study associated with this tutorial (Giacomelli et al. 2024 GBE in press), we wanted to test the ability of the considered models to describe the across site compositional heterogeneity of the data. To this scope we used **amino acid diversity** (the mean number of distinct amino acids per site across the sequence alignment), as the measure to calculate on the data. Note that estimation of **amino acid diversity** is model-independent, even though we expect the amino acid diversity simulated using different models to vary. The use of the **amino acid diversity** of an alignment to test the ability of models to describe across-site compositional heterogeneity was introduced by Lartillot in the Bayesian software Phylobayes (Lartillot et. al, 2013), where it is used to test model adequacy using Posterior Predictive analysis.

We developed a custom python script 'calculate_zscore.py' to calculate the number of amino acids per column in a dataset. The script calculates the mean value for the real data, and for each simulated dataset. The mean values estimated for the simulated data are then used to calculate a global average and standard deviation, and the distance from the global average of the mean value estimated for the real data is calculated as a Z-score. See Giacomelli et al., 2022 for all the details of how Z-scores can be interpreted. This script takes one single argument, the name of the file containing the real data, assuming that all simulated datasets finish with *.gaps*. If the script, the real data and the simulated data are in the same directory, the script can be run as follow:

3.1: *python calculate_zscore.py nematode.fasta*

The output of Line of code 3.1 includes two files *diversity.pbr_gaps* and *diversity_scores_bootstrapped_data.txt_gaps*. The first is a summary of the results, the second contains the amino acid diversity score (*div*) of each simulated dataset.

Step 4. Interpreting the results

Let's look at output of diversity.pbr_gaps for each model and at the distributions of the simulated *div* values in *diversity_scores_bootstrapped_data.txt_gaps*.

Step 4.1. Summary of the model fit analysis for the considered models

This is the content of the *diversity.pbr_gaps* files.

LG+G4:

Diversity original data: 3.0102626445393117

Average diversity simulated data: 3.6171063865878814

SD simulated data: 0.013443359538433675

Z-score: 45.14078049565242

Poisson-C10+G4

Diversity original data: 3.0102626445393117

Average diversity simulated data: 3.4308888637584456

SD simulated data: 0.013145565639355617

Z-score: 31.99757475325745

LG-C10+G4

Diversity original data: 3.0102626445393117

Average diversity simulated data: 3.405902575556248

SD simulated data: 0.01250266061177053

Z-score: 31.64445899174968

LG-CAT-PMSF+G4

Diversity original data: 3.0102626445393117

Average diversity simulated data: 3.32613949280484

SD simulated data: 0.02709095067287328

Z-score: 11.659865764025115

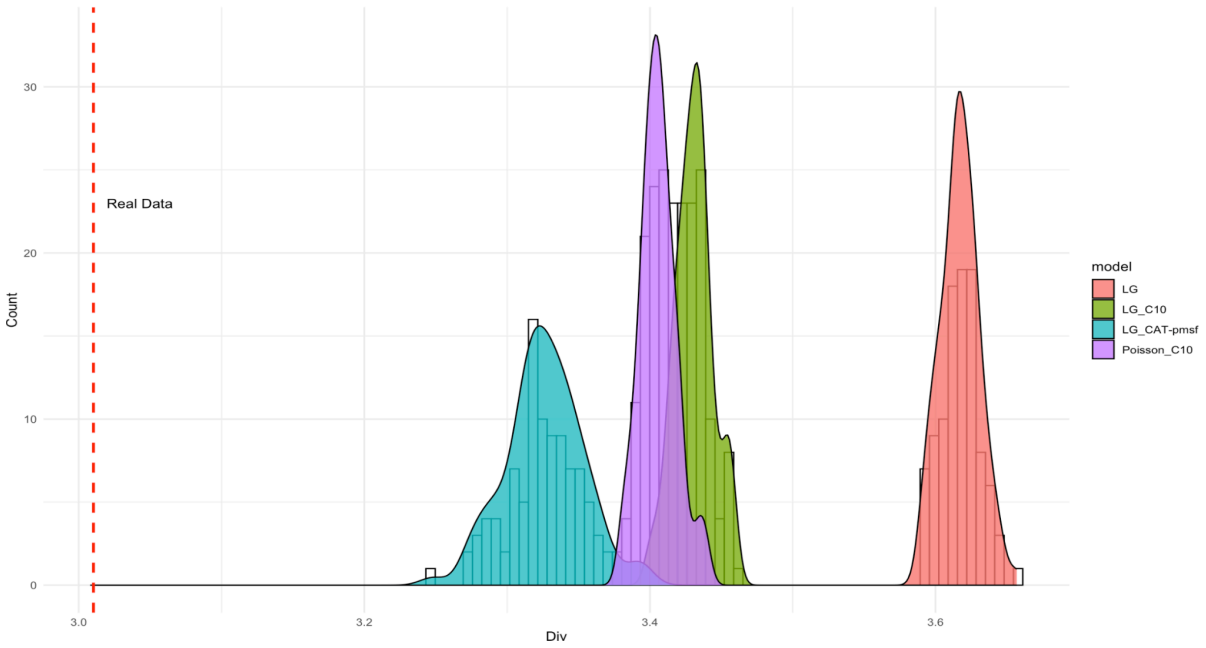
The inspection of the *diversity.pbr_gaps* files illustrates a few points. First the **Diversity original data** is always the same. This is obviously expected as it is the average across-site compositional heterogeneity of our real dataset. The **Average diversity simulated data** show that LG+G substantially overestimate across-site compositional heterogeneity and that as we use more complex across-site compositionally heterogeneous models, the estimated amino acid diversity decreases, becoming progressively more comparable to that observed for the real data. This indicates a progressively better fit of the tested models to the data: as the model fit improves, the ability of the model to simulate datasets of greater similarity to the real ones increases. Finally the **Z-scores** show that none of the considered models adequately describe the data, despite LG-CAT-PMSF+G4 being the least misspecified model (Z-score ~ 11). LG+G4 is the most misspecified model (Z-score ~ 45). Notably, as briefly discussed in the introductory part of the tutorial LG+G4 (the most poorly fitting model) fails to recover Ecdysozoa, while the other three models, which progressively fit the data better, and all fits it better than LG+G4 recover the target clade (Ecdysozoa).

It is notable that in this specific example, none of the considered models adequately describes the data.

Step 4.2. Plotting *Div* values.

This uses the content of *diversity_scores_bootstrapped_data.txt_gaps* files.

Plotting the *div* values found in the *diversity_scores_bootstrapped_data.txt_gaps* files (one for each simulated datasets) we can visualise distributions of *div* values. This provides further insights on the fit of alternative models to the data (see Figure 1). The dotted red line indicates the *div* value of the real data and it represents our reference point: a model adequately describes the data if the *div* value for the real data falls within the distribution of *div* values for the simulated datasets. The distributions of div values for the simulated datasets clearly confirm the conclusions based on the interpretations of the Z-scores. These are: (1) None of the considered models adequately describes the data. (2) LG+G4 fits the data much worse than the across-site compositionally heterogeneous models. (3) LG-CAT-PMSF+G4 fits the data better than the other models, but still fails to adequately describe the data. In addition, the distribution of div values allows us to clearly demonstrate that Poisson-C10+G4, LG-C10+G4 and LG-CAT-PMSF+G4 have comparable fit, their *div* scores distributions overlap. However, distributions of *div* scores also allow us to conclude that the fit of Poisson-C10+G4 is marginally worse than that of of LG-C10+G4 and LG-CAT-PMSF+G4, and that that of LG-CAT-PMSF+G4 is marginally better.



An alternative example where one of the models tested adequately fit the data

Parametric bootstrap based model adequacy testing allows us to conclude that none of the models considered in the nematode example, adequately fit the data. This is in contrast to the results we present in (Giacomelli et al. 2024, *in press*) where we find that CAT-PMSF+G4 adequately fit the data. This is not surprising as results of model-adequacy tests are dataset dependent and explicitly designed to allow discovering whether the model used was a good fit to the data or not. It is useful to show what the results of model adequacy tests look like when the model adequately fits the data. To achieve this goal we report here the results we obtained for the Tardigrade dataset of (Giacomelli et al. 2024, *in press)*. The pipeline used to obtain these results is exactly the same used in Steps 1 to 3 above. What we report here is only Step 4 (the interpretation of the results).

Note that in Giacomelli et al. (2024) the compared models were LG+G4, Poisson-C60+G4, LG-C60-PMSF+G4 and CAT-PMSF+G4, and here are the outputs from the *diversity.pbr_gaps* files.

LG+G4:

Diversity original data: 4.395595869540684

Average diversity simulated data: 5.082196563593275

SD simulated data: 0.012984484127109789

Z-score: 52.87855007031543

Poisson-C60+G4

Diversity original data: 4.395595869540684

Average diversity simulated data: 4.634387907685365

SD simulated data: 0.009412032682771007

Z-score: 25.370931677893225

LG-C60-PMSF+G4

Diversity original data: 4.395595869540684

Average diversity simulated data: 4.566540740322764

SD simulated data: 0.04141919534929521

Z-score: 4.127189563690753

Poisson-CAT-PMSF+G4

Diversity original data: 4.395595869540684

Average diversity simulated data: 4.376493905879697

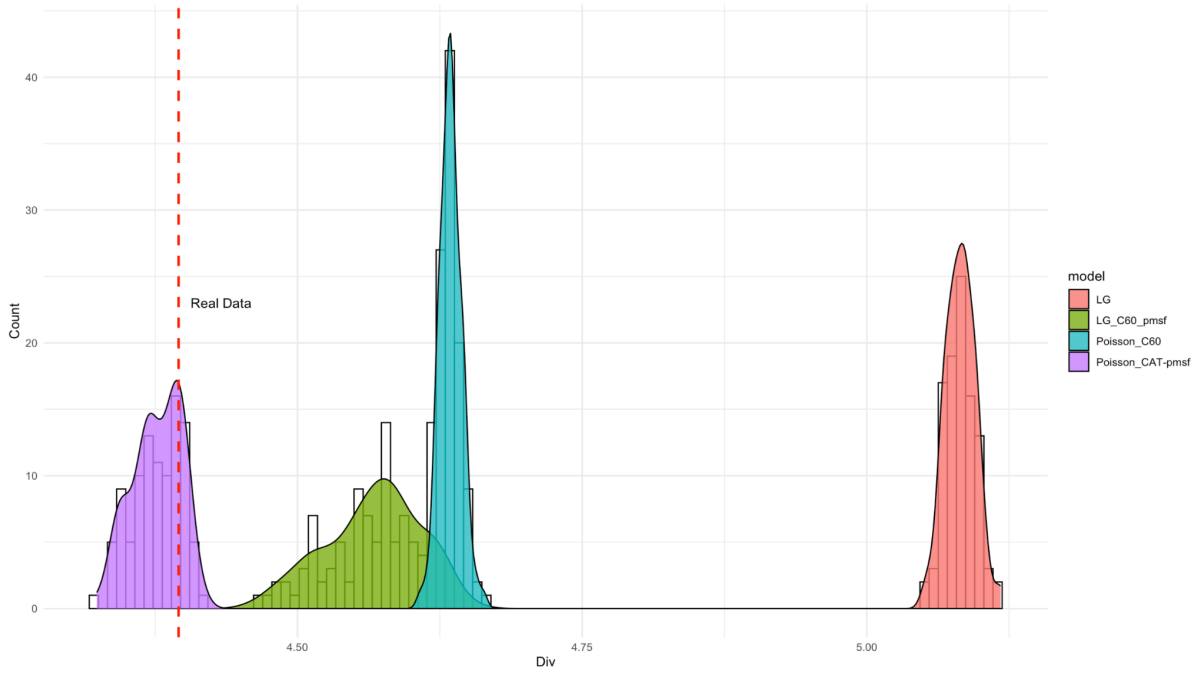
SD simulated data: 0.021295340237854104

Z-score: -0.897002041180417

The results above illustrate (as in the nematodes case) that LG+G4 overestimates amino acid diversity, and that as across-site compositionally heterogeneous models with more site-frequency categories are used, the simulated datasets achieve a level of across-site compositional heterogeneity that is better comparable to that of the real data. CAT-PMSF+G4 is the model that more closely approximates the amino acid diversity observed for the real data, very mildly underestimating it. Z-scores clearly show that LG+G fit the data very poorly (Z-Score ~ 52), while CAT-PMSF+G4 adequately describes the data (Z-score ~ -0.9). **Note: -2< Z-score <2 are usually considered to indicate that the model fits the data (Giacomelli et al. 2022)**. This is because, when the distribution is normal, such Z-scores would indicate that the *div* value observed for the real data falls within the distribution of the simulated data. However, this conclusion is contingent on the assumption that *div* distributions are normally distributed, which is usually not the case (see Fig. 1 and Fig 2). Accordingly, Z-scores should not be interpreted stringently (see Giacomelli et al. 2022), for a discussion. Visualising the distribution of *div* scores helps interpreting the results (as discussed for the nematodes). Here for example, we see that indeed the div value calculated for the real data falls well within the distribution of values simulated using

CAT-PMSF+G4, confirming that this model adequately describes the data. As in the nematode case LG+G4 dramatically fails to fit the data. However, the cases of Poisson-C60+G4 and LG-C60-PMSF+G4 are interesting and instructive. While the distribution of *div* values of Poisson-C60+G4 overlap with the distribution of *div* values for LG-C60-PMSF+G4, the fit of the two models is very different according to Z-scores. This is because *div* values achieved by LG-C60-PMSF+G4 have a much larger spread and the values for Poisson-C60+G4 are clustered tightly at the tail end of the LG-C60-PMSF+G4 distribution, in the direction of poor fit values. However, when looking at the full distribution of LG-C60-PMSF+G4 it is quite clear that it peaks quite close to the right-side tail end, where Poisson-C60+G4. We conclude that in the case of LG-C60-PMSF+G4 interpreting its Z-score (Z= 4.12) without referring back to the shape of the distribution of its simulated *div* values, would lead to an overestimation of its fit. When the full div scores distributions are considered, it emerges that the fit of LG-C60-PMSF+G4 is more similar to that of Poisson-C60+G4 than to that of CAT-PMSF+G4, despite the Z-Score of LG-C60-PMSF+G4 (Z= 4.12), is more similar to that of CAT-PMSF+G4 (Z= -0.89), than to that of Poisson-C60+G4 (Z= 25.37), which is why we recommend that Z-scores are to be interpreted together with their corresponding *div* values distributions to accurately understand the ability of the considered models to adequately describe the data.

As a final note we would like to suggest that we think this is reflected in the phylogenies inferred from these models. In the nematode case, when all three across-site compositionally heterogeneous models had comparable fit, they inferred the same tree. Differently, in the tardigrades case, CAT-PMSF+G4 infer trees (see Giacomelli et al. 2024, *in press,* for details) that differ from those inferred using the three poorly fitting models (LG+G4, Poisson-C60+G4 and LG-C60-PMSF+G4). However, whether this is a pattern that can be regularly expected would need further testing.



References:

Feuda R, Dohrmann M, Pett W, Philippe H, Rota-Stabelli O, Lartillot N, Wörheide G, Pisani D. 2017. Improved Modeling of Compositional Heterogeneity Supports Sponges as Sister to All Other Animals. *Current Biology* 27:3864-3870.e4.

Giacomelli M, Rossi ME, Lozano-Fernandez J, Feuda R, Pisani D. 2022. Resolving tricky nodes in the tree of life through amino acid recoding. *iScience* 25:105594.

Lartillot N, Philippe H. 2004. A Bayesian Mixture Model for Across-Site Heterogeneities in the Amino-Acid Replacement Process. *Molecular Biology and Evolution* 21:1095–1109.

Lartillot N, Brinkmann H, Philippe H. 2007. Suppression of long-branch attraction artefacts in the animal phylogeny using a site-heterogeneous model. *BMC Evol Biol* 7:S4.

Lartillot N, Rodrigue N, Stubbs D, Richer J. 2013. PhyloBayes MPI: Phylogenetic Reconstruction with Infinite Mixtures of Profiles in a Parallel Environment. *Systematic Biology* 62:611–615.

Ly-Trong N, Barca GMJ, Minh BQ. 2023. AliSim-HPC: parallel sequence simulator for phylogenetics.Schwartz R, editor. *Bioinformatics* 39:btad540.

Ly-Trong N, Naser-Khdour S, Lanfear R, Minh BQ. 2022. AliSim: A Fast and Versatile Phylogenetic Sequence Simulator for the Genomic Era.Crandall K, editor. *Molecular Biology and Evolution* 39:msac092.

Minh BQ, Schmidt HA, Chernomor O, Schrempf D, Woodhams MD, von Haeseler A, Lanfear R. 2020. IQ-TREE 2: New Models and Efficient Methods for Phylogenetic Inference in the Genomic Era. *Molecular Biology and Evolution* 37:1530–1534.

Szánthó LL, Lartillot N, Szöllősi GJ, Schrempf D. 2023. Compositionally Constrained Sites Drive Long-Branch Attraction. *Systematic Biology* 72:767–780.