Tutorial: Four Metrics of the Parametric Bootstrap Test

Yutong Shao

Contents

1 Introduction

In phylogenetic inference, profile mixture models are widely used to model site-specific compositional heterogeneity in amino acid substitution processes. These models allow the amino acid frequencies to vary across sites by introducing multiple profiles, thereby better capturing site-specific evolutionary preferences in the sequence. However, such models typically involve a large number of free parameters—for example, each additional profile introduces 20 new parameters—making their estimation computationally expensive, especially when applied to large datasets or highly complex models.

To alleviate this computational burden, Wang et al. (2017) proposed the PMSF (Posterior Mean Site Frequency) model. Based on a pre-estimated guide tree, the PMSF model approximates the original profile mixture model by calculating the posterior mean frequency for each site. This approximation can significantly reduce the computational complexity of likelihood calculations.

However, since PMSF does not explicitly estimate the number of parameters, conventional model selection criteria such as AIC or BIC cannot be applied. To address this, Giacomelli et al. (2025) introduced a new evaluation approach: the parametric bootstrap test. This method assesses model fit by comparing the observed and simulated values of average amino acid type count per site (i.e., diversity, abbreviated as div) under different profile mixture models.

Building upon the method proposed by Giacomelli et al., this study further extends the evaluation framework by introducing three additional metrics for the parametric bootstrap test:

- the **mean entropy** of the dataset;
- and **CvM (Cramér–von Mises) tests** comparing the distributions of **sitewise entropy** and **sitewise diversity** between observed and simulated datasets.

Our experimental results demonstrate that, compared to simply using the average div value of a dataset, applying the CvM test to compare the sitewise distributions of entropy or diversity provides a more accurate evaluation of the model adequacy for mixture models.

This tutorial presents the complete workflow for performing phylogenetic inference using different models, as well as conducting parametric bootstrap tests under four different metrics. The example dataset used in this tutorial, `data/alignment.fasta`, is a BUSCO gene sequence derived from the Cyanobacteria dataset described in Pardo-De la Hoz et al. (2023).

2 Parameter Inference and Model Fitting

I used IQ-TREE3 (Wong et al., 2025) to perform inference under each model. All output results were stored in the `data/model_inference/` folder.

2.1 Homogeneous model

I applied the ModelFinder function in IQ-TREE and restricted the search to the LG substitution matrix. From all possible combinations, the best-fitting homogeneous model was selected.

```
iqtree3 -s alignment.fasta -m MF -mset LG -pre infer/MF
```

2.2 LG+C60

Use the C60 profile mixture model combined with the FreeRate model (R3) to account for site rate heterogeneity:

```
iqtree3 -s alignment.fasta -m LG+C60+R3 -pre infer/LG-C60-R3 -bb 1000 --wbtl
```

2.3 LG+C60fix-PMSF

The PMSF method can be divided into two phases: In the first phase, given a guide tree, IQ-TREE infers the site-specific frequency profile for each site. In the second phase, the inferred site frequencies are then used to perform phylogenetic inference.

```
# Step 1: Infer site-frequency
iqtree3 -s alignment.fasta -ft infer/MF.treefile \
  -m LG+C60+R3 -safe -pre infer/LG-C60fix-R3-PMSF-step1 -n 0

# Step 2: Continue the analysis using the inferred site frequencies
iqtree3 -s alignment.fasta -fs infer/LG-C60fix-R3-PMSF-step1.sitefreq \
  -m LG+C60+R3 -safe -pre infer/LG-C60fix-R3-PMSF-step2 -wsr -bb 1000 --wbtl
```

2.4 LG+C60opt-PMSF

This approach also uses the C60 profile mixture model together with the PMSF method, but here the profile weights of the C60 model are optimized beforehand:

```
# Step 0-1: Optimize the mixture weights using the guide tree
iqtree3 -s alignment.fasta -m LG+C60+R3 \
  -te infer/LG-C60-R3.contree -me 0.99 -safe -mwopt \
  -pre infer/LG-C60opt-R3-PMSF-step0 -prec 10
```

After this step, you can use `extract_profiles_weights.py` to extract the optimized weights and manually write them into a custom model file, like `data/nex_file/C60opt.nex`.

```
# Step 0-2: Extract optimized weights
python extract_profiles_weights.py ./data/infer

# Step 1: Infer site-frequency
iqtree3 -s alignment.fasta -ft infer/LG-C60opt-R3-PMSF-step0.treefile \
  -m C60opt -mdef C60opt.nex -safe \
  -pre infer/LG-C60opt-R3-PMSF-step1 -n 0

# Step 2: Continue the analysis using the inferred site frequencies
iqtree3 -s alignment.fasta -fs infer/LG-C60opt-R3-PMSF-step1.sitefreq \
  -m C60opt -mdef C60opt.nex -safe \
  -pre infer/LG-C60opt-R3-PMSF-step2 -wsr -bb 1000 --wbtl
```

2.5 GTR20+C60fix-PMSF

This model combines dataset-specific exchangeability with the fixed weights of the C60 mixture profile.

There are several ways to estimate GTR20, for example:

(1) Based on a homogeneous model

(2) GTRpmix approach: estimate different exchangeabilities under different profile mixture models

In the following example, I use the best homogeneous model selected by ModelFinder and its resulting tree as the guide tree:

```
# Step 0: estimate GTR20 exchangeability
iqtree3 -s alignment.fasta -m GTR20+R3 -te infer/MF.treefile \
  --model-joint GTR20+FO --init-exchange LG -pre infer/GTR20
```

At this stage, you should manually extract the newly estimated exchangeability from the `GTR20.iqtree` file produced in step 0. Next, create a custom model file (e.g., `data/nex_file/GTR20.nex`) and write these exchangeability into it for use in the subsequent analysis.

```
# Step 1: Infer site-frequency
iqtree3 -s alignment.fasta -ft infer/MF.treefile \
  -m GTRPMIX+C60+R3 -mdef infer/GTR20.nex -safe \
  -pre infer/GTR20-C60fix-R3-PMSF-step1 -n 0

# Step 2: Continue the analysis using the inferred site frequencies
iqtree3 -s alignment.fasta -fs infer/GTR20-C60fix-R3-PMSF-step1.sitefreq \
  -m GTRPMIX+C60+R3 -mdef infer/GTR20.nex -safe \
  -pre infer/GTR20-C60fix-R3-PMSF-step2 -wsr -bb 1000 --wbtl
```

2.6 GTR20+C60opt-PMSF

Here, I used the same exchangeability as in the GTR20+C60fix-PMSF approach described above, but with an additional step to optimize the profile weights.

```
# Step 0-1: Optimize the mixture weights using the guide tree
iqtree3 -s alignment.fasta -m GTRPMIX+C60+R3 \
  -te infer/LG-C60-R3.contree -me 0.99 -safe -mwopt \
  -pre infer/LG-C60opt-R3-PMSF-step0 -prec 10

# Step 0-2: Extract optimized weights
python extract_profiles_weights.py ./data/infer

# Step 1: Infer site-frequency
iqtree3 -s alignment.fasta -ft infer/MF.treefile \
```

```
  -m GTRC60opt -mdef infer/GTRC60opt.nex -safe \
  -pre infer/GTR20-C60opt-R3-PMSF-step1 -n 0

# Step 2: Continue the analysis using the inferred site frequencies
iqtree3 -s alignment.fasta -fs infer/GTR20-C60opt-R3-PMSF-step1.sitefreq \
  -m GTRC60opt -mdef infer/GTRC60opt.nex -safe \
  -pre infer/GTR20-C60opt-R3-PMSF-step2 -wsr -bb 1000 --wbtl
```

3 Generate Simulated Sequences

After obtaining the model parameters, I used the Alisim function (Ly-Trong et al., 2023) to generate 100 simulated alignments for each model. These simulated datasets serve as input for the downstream analysis using `PBT.py`.

Before running the simulations, for each model that applies the PMSF method, you need to first combine the `.sitefreq` file from Step 1 with the `.rate` and `.iqtree` files from Step 2. This is done using the `combine_sitefq_rate.py` script, which extracts a partition model file that defines one partition per site. This partition file is required for simulating sequences under the PMSF model. The process is as follows:

1. Make sure the `.sitefreq`, `.rate`, and `.iqtree` files corresponding to the same model are stored in the same folder and share the same prefix. For example:

```
model_1/
    model_1-step1.sitefreq
    model_1-step2.rate
    model_1-step2.iqtree
    ...
```

2. Run the `combine_sitefq_rate.py` script:

```
python combine_sitefq_tate.py ./model_1
```

This will generate a file named `model_1.nex`, which is the partition definition file used for the simulation.

3.1 Homogeneous model

```
iqtree3 --alisim simulation/LG+R3/seq \
  --seqtype AA -t infer/MF.treefile \
  -m "LG+R3{0.7610,0.1354,0.1775,2.6626,0.0615,6.8982}" \
  --length 1321 --out-format fasta --num-alignments 100
```

3.2 LG+C60

```
iqtree3 --alisim simulation/LG+C60+R3/seq \
  --seqtype AA -t infer/LG-C60-R3.contree \
  -m "LG+C60+R3{0.7622,0.1144,0.1782,2.4763,0.0595,7.9200}" \
  --length 1321 --out-format fasta --num-alignments 100
```

3.3 LG+C60fix-PMSF

```
iqtree3 --alisim simulation/LG+C60fix+R3-PMSF/seq \
  --seqtype AA -p infer/LG-C60fix-R3-PMSF.nex \
  -t infer/LG-C60fix-R3-PMSF-step2.contree \
  --length 1321 --out-format fasta --num-alignments 100
```

3.4 LG+C60opt-PMSF

```
iqtree3 --alisim simulation/LG+C60opt+R3-PMSF/seq \
  --seqtype AA -p infer/LG-C60opt-R3-PMSF.nex \
  -t infer/LG-C60opt-R3-PMSF-step2.contree \
  --length 1321 --out-format fasta --num-alignments 100
```

3.5 GTR20+C60fix-PMSF

```
iqtree3 --alisim simulation/GTR20+C60fix+R3-PMSF/seq \
  --seqtype AA -p infer/GTR20-C60fix-R3-PMSF.nex \
  -t infer/GTR20-C60fix-R3-PMSF-step2.contree \
  --length 1321 --out-format fasta --num-alignments 100 \
  -mdef infer/GTR20.nex
```

3.6 GTR20+C60opt-PMSF

```
iqtree3 --alisim simulation/GTR20+C60opt+R3-PMSF/seq \
  --seqtype AA -p infer/GTR20-C60opt-R3-PMSF.nex \
  -t infer/GTR20-C60opt-R3-PMSF-step2.contree \
  --length 1321 --out-format fasta --num-alignments 100 \
  -mdef infer/GTR20.nex
```

4 Parametric Bootstrap Test

After completing the simulation process described above, we can obtain a directory with the following structure:

```
-- simulation/
      LG+R3/
          seq_1.fa
          seq_2.fa
          ...
          seq_100.fa
      LG+C60+R3
      ...
      GTR20+C60opt+R3-PMSF
```

Next, we evaluate how well each model fits the empirical data by running the following command:

```
python PBT.py <SimRootFolder> <OriginalAlignment>
```

This `PBT.py` script compares the empirical dataset with the simulated datasets using the following four model adequacy metrics:

- Mean diversity
- Mean entropy
- CvM test on sitewise diversity
- CvM test on sitewise entropy

The results of model comparison will be summarized in the output log file `PBT.log`. For example, here is the file `data/model_results/PBT.log`:

```
Model Adequacy Evaluation
----------------------

Best-fit model according to Mean Difference (entropy): GTR20+C60fix+R3-PMSF

List of models sorted by |original - simulated| for entropy:

Model                          Entropy   AvgSim   SD     Z      |Diff|
-------------------------------------------------------------------------

GTR20+C60fix+R3-PMSF           0.2717    0.2631  0.0149 -0.58   0.0086
LG+C60fix+R3-PMSF              0.2717    0.2809  0.0143  0.65   0.0092
GTR20+C60opt+R3-PMSF           0.2717    0.2820  0.0149  0.69   0.0103
LG+C60+R3                      0.2717    0.2864  0.0155  0.95   0.0147
LG+C60opt+R3-PMSF              0.2717    0.2973  0.0145  1.77   0.0256
LG+R3                          0.2717    0.3163  0.0166  2.68   0.0446
```

As shown above, the model **GTR20+C60fix+R3-PMSF** is selected as the best-fit model under the **mean entropy** criterion, *having the smallest absolute difference between the empirical and simulated entropy values*. Models are ranked based on the magnitude of this difference (|Diff|), from smallest (best) to largest (worst).

The model selection approach using mean diversity as the metric follows the same logic as described above for mean entropy.

In contrast, when using the CvM test on sitewise diversity as the evaluation metric, the results are as follows:

```
Best-fit model according to Cvm Test (entropy): GTR20+C60opt+R3-PMSF

List of models sorted by Cvm Test (entropy) W2_statistic:

Model                          W2_statistic
----------------------------------------
```

GTR20+C60opt+R3-PMSF	0.0917
LG+C60+R3	0.1032
LG+C60opt+R3-PMSF	0.1087
LG+R3	0.1605
LG+C60fix+R3-PMSF	0.2486
GTR20+C60fix+R3-PMSF	0.3130

The CvM test (Cramér–von Mises test) is used to compare two distributions—in this case, the distribution of per-site entropy between the simulated datasets and the empirical dataset. The W^2 statistic is the test statistic, where smaller values indicate greater similarity between the two distributions.

As shown in the table above, the model **GTR20+C60opt+R3-PMSF** is selected as the best-fit model under the **CvM test on site-wise entropy** criterion, having the smallest W^2 statistic value. All models in the table are ranked in ascending order based on their W^2 statistic values.

5 Optional Analysis: Visualization

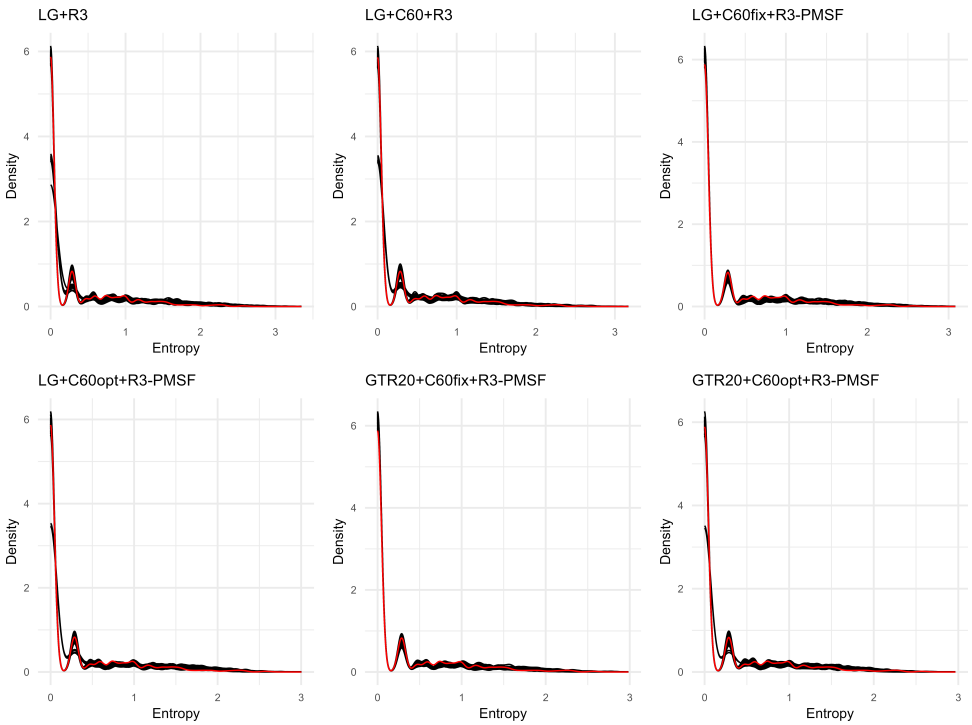

Figure 1: Per-site entropy distribution of example data.

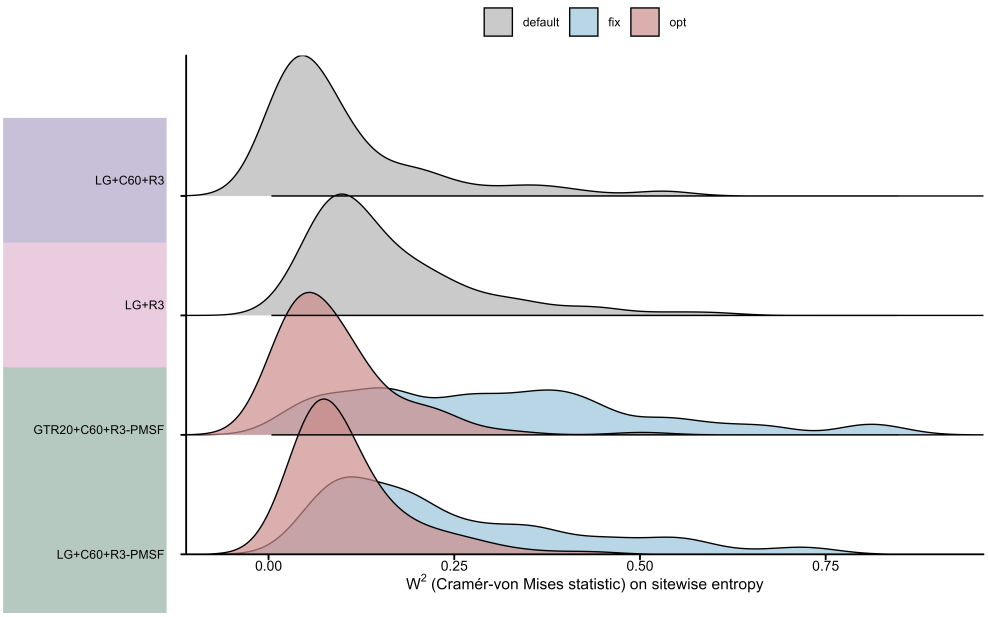

Figure 2: Cvm test on site-wise entropy distribution plot.

References

Giacomelli, M., Vecchi, M., Guidetti, R., Rebecchi, L., Donoghue, P., Lozano-Fernandez, J. and Pisani, D. (2025). CAT-posterior mean site frequencies improves phylogenetic modeling under maximum likelihood and resolves Tardigrada as the sister of Arthropoda plus Onychophora. *Genome Biology and Evolution*, 17(1). https://doi.org/10.1093/gbe/evad003

Ly-Trong, N., Barca, G.M.J. and Minh, B.Q. (2023). AliSim-HPC: parallel sequence simulator for phylogenetics. *Bioinformatics*, 39, btad540. https://doi.org/10.1093/bioinformatics/btad540

Pardo-De la Hoz, C.J., Magain, N., Piatkowski, B., Cornet, L., Dal Forno, M., Carbone, I., Miadlikowska, J. and Lutzoni, F. (2023). Ancient rapid radiation explains most conflicts among gene trees and well-supported phylogenomic trees of Nostocalean Cyanobacteria. *Systematic Biology*, 72(3), pp.694–712. https://doi.org/10.1093/sysbio/syad009

Wang, H.C., Minh, B.Q., Susko, E. and Roger, A.J. (2018). Modeling site heterogeneity with posterior mean site frequency profiles accelerates accurate phylogenomic estimation. *Systematic Biology*, 67(2), pp.216–235. https://doi.org/10.1093/sysbio/syx068

Wong, T.K.F., Ly-Trong, N., Ren, H., Banos, H., Roger, A.J., Susko, E., Bielow, C., De Maio, N., Goldman, N., Hahn, M.W., Huttley, G., Lanfear, R. and Minh, B.Q. (2025). *IQ-TREE 3: Phylogenomic inference software using complex evolutionary models*.