Jun 13, 2025

# AU Test with IQTree

DOI
dx.doi.org/10.17504/protocols.io.e6nvw5w7zvmk/v1

Dakota Betz[1]

[1]ucsd

Rouse Lab

Marina McCowin
UCSD- Scripps Institution of Oceanography

**DOI:** **dx.doi.org/10.17504/protocols.io.e6nvw5w7zvmk/v1**

**Protocol Citation:** Dakota Betz 2025. AU Test with IQTree . **protocols.io**
**https://dx.doi.org/10.17504/protocols.io.e6nvw5w7zvmk/v1**

**Protocol status:** Working
**We use this protocol and it's working**

**Created:** October 27, 2021

**Last Modified:** June 13, 2025

**Protocol Integer ID:** 54586

## Disclaimer

Our protocols are constantly evolving and old versions will be deleted.
The documents here are not intended to be cited in publications

## Abstract

Our protocols are constantly evolving and old versions will be deleted.
The documents here are not intended to be cited in publications

## Programs and Dependencies

1 **IQ-Tree**
Download: **http://www.iqtree.org/**
Additional information/tutorials: http://www.iqtree.org/doc/Advanced-Tutorial

**Mesquite**
Download: https://www.mesquiteproject.org/Installation.html

**RAxML-ng**
Download: https://github.com/amkozlov/raxml-ng
GUI 2.0 (compatible with ng): **https://antonellilab.github.io/raxmlGUI/**

**Text Editor** (options below)
SublimeText: https://www.sublimetext.com/3
BBEdit: https://www.barebones.com/products/bbedit/download.html

2 This protocol assumes you have already completed initial tree inference. It is designed to test another topology (constrained) against your unconstrained analysis. Before beginning this protocol, you will need:

- a copy of your unconstrained tree (.tre file)
- the data you used to create the unconstrained tree (.phy file, probably nucleotide data)
- knowledge/list of model(s) applied to your data
- if you're running a partitioned analysis (multiple genes or data types), you will also need a partition file (.txt or .nex file)

## File Preparation

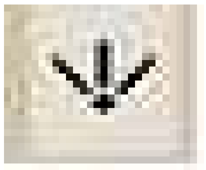
3 Make sure you have a copy of your original RAxML tree (against which you would like to test your constraint tree, which you have not created yet).

Open your original tree in FigTree if it is a .tre file and convert it to a nexus file. To do this, click:
**File > Export Trees**
And choose **Nexus** from the dropdown menu.

Append the filename of this nexus file with "_original.nex" so that it's easy to tell apart from the constraint you will make.

4 You will use the original file as a base to start your constraint. Open your tree file "_original.nex" in Mesquite. Re-save it as "_constraint.nex" before you begin making changes.

5 Collapse all of the nodes in your tree into a single polytomy, including the outgroup. See image below of the collapse tool:



6 Now select *only the taxa you want to constrain* and make those a *single polytomy* using the **arrow cursor** tool (they should be separate from the other polytomy which is the rest of the tree). The tree should now have a total of 2 polytomies.

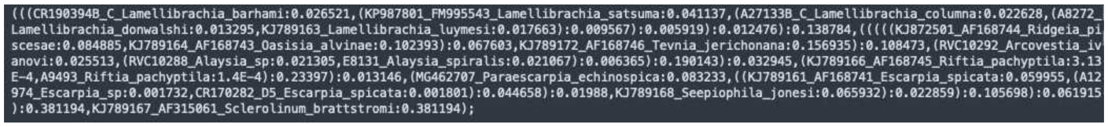
Reminder: the outgroup is not separated out--it can be in the polytomy with the remaining taxa that you are not intending to constrain. Save again just in case (remember filename should now be "_constraint.nex").

7 Select **Tree > Store tree in other block as** and give your tree a name (e.g. "constraint"). Then **Store in new tree block** and name the new tree block. Select **switch to new tree** and select your new tree. To export this, click **File > Export > Export Nexus tree file from tree source > stored trees**. Make sure you select the tree you want (constraint). Check the box to **include taxa block**, then click **Export > Save**.

8 You should be able to open the newly saved "_constraint.nex" file in FigTree. If not, make a copy of this file outside mesquite (should be a .nex file) and add the extension .tre instead (works on Mac OS only). Now open the "_constraint.tre" tree in FigTree.

Once you have confirmed that Mesquite/FigTree saved the topology you wanted, export as a .nex file again from FigTree.

9 Drag this file ("_constraint.nex") into a text editor (e.g. SublimeText) and remove everything except the tree notation/newick format (the portion in parentheses). See below for an example of what this looks like. Save this as "_constraint.txt". in case the .nex version fails in RAxML.



```
(((CR190394B_C_Lamellibrachia_barhami:0.026521,(KP987801_FM995543_Lamellibrachia_satsuma:0.041137,(A27133B_C_Lamellibrachia_columna:0.022628,(A8272_
Lamellibrachia_donwalshi:0.013295,KJ789163_Lamellibrachia_luymesi:0.017663):0.009567):0.005919):0.012476):0.138784,(((((KJ872501_AF168744_Ridgeia_pi
scesae:0.084885,KJ789164_AF168743_Oasisia_alvinae:0.102393):0.067603,KJ789172_AF168746_Tevnia_jerichonana:0.156935):0.108473,(RVC10292_Arcovestia_iv
anovi:0.025513,(RVC10288_Alaysia_sp:0.021305,E8131_Alaysia_spiralis:0.021067):0.006365):0.190143):0.032945,(KJ789166_AF168745_Riftia_pachyptila:3.13
E-4,A9493_Riftia_pachyptila:1.4E-4):0.23397):0.013146,(MG462707_Paraescarpia_echinospica:0.083233,((KJ789161_AF168741_Escarpia_spicata:0.059955,(A12
974_Escarpia_sp:0.001732,CR170282_D5_Escarpia_spicata:0.001801):0.044658):0.01988,KJ789168_Seepiophila_jonesi:0.065932):0.022859):0.105698):0.061915
):0.381194,KJ789167_AF315061_Sclerolinum_brattstromi:0.381194);
```

## RAxML Likelihood Estimation

10 For this part of the protocol, you will need the data and partitions you used to create your original/unconstrained tree, i.e.:

- your concatenated dataset (nucleotide or amino acid)
- your partitions for your concatenated dataset (if applicable)

Note

*If you used RAxML-ng and the GUI 2.0 to make your unconstrainted tree, your concatenated file will be generated for you, and should end in "concat.txt". Your partitions file should also be generated for you, ending in "concat.part.txt".*

11

Open the RAxML GUI and ensure you're set to run using RAxML-ng. Load in your concatenated dataset (concat.txt) file from your original/unconstrained phylogenetic analysis in RAxML. Load in your partition file (part.txt) from that analysis as well.

12 Now you will import the *constraint* tree file into RAxMLng as a **multifurcating constraint**. You can do this in the GUI by selecting (at the top menu) **Analysis > Enforce Constraint > Use multifurcating constraint**. When prompted, open your constraint tree file in either .nex or .txt format (either should work, but if .nex fails, try .txt).

13 Make sure you have the following settings in the RAxML GUI: **RAxML-ng, ML Tree Inference, 1 run**. Select the same outgroup that you used in your unconstrained analysis. Change the output name and folder to something that makes sense (e.g. "_RAxMLconstrained").

Note

*Note: You do NOT need to bootstrap anything at this point in RAxML.*

14 **Run** RAxML. The resulting "best tree" will be the constrained tree that you test against your "_original" unconstrained tree for the AU test. Name it something like "_RAxMLconstrained" so you can tell the difference between this and the constraint tree you made in Mesquite. To make the tree easier to open in FigTree, add the .tre extension to the end of the filename.

## Combining Tree Files

15 Ensure both your "_original" and your new "_RAxMLconstrained" trees are in Nexus format. You can do this by opening them in FigTree and exporting them as Nexus files as in this step: go to step #3

16 Open one of your trees in a text editor (e.g. SublimeText). Resave it as "_combined.nex" before you begin to make changes.

Make sure the label (probably "tree_1" in the text file) is edited to reflect which tree this is. For ex: if you opened your constrained tree first and renamed the file, make sure "tree_1" in this file is replaced with "constrained".

17 Open your other tree in a text editor. Copy and paste the tree notation (with the name, be sure to edit it to reflect which tree it is like you did in the previous step) into your "_combined.nex" file. Be sure to appropriately label this tree as well (this will be "original" if you opened the constrained first in the above step). Save and close your "_combined.nex" file.

18 When you save this tree in newick format, you'll lose the tree names youo created. This step will help you create a record or which tree was which so you can check back later if needed. Re-save tthe combined.nex file you just created as "_combined.tre". Now delete everything except the tree notation (look for parentheses/at example in this step : go to step #9 ). This includes the tree names and any other symbols, except the parentheses involved with the trees themselves (do *not* delete those) and the semicolon after each tree. Now if you forget which tree is which, you know they're in the same order in this file as in the "_combined.nex" file and can check back there if needed.

19 Open your "_combined.tre" file in FigTree to check that the trees look as expected. Export the tree in Newick format (same procedure as exporting for nexus, but select **newick** from the dropdown menu and "select all trees") with a .txt extension at the end of the filename.

## For Multiple Models: Prepare Partition File

20

Note

*If you plan on only applying a SINGLE model to your data (with or without partitions) for the AU test, ignore this section. You can proceed with a RAxML-style partition file.*

*If you would like to apply different models to different partitions for the AU test, you will need to create the more flexible nexus-style partition file (instead of the RAxML-style). Proceed with this section for detailed instructions.*

21 Open your RAxML-style partition file in a text editor. You will need to convert it to a nexus-style partition file (example below). This file constrains a **SETS** block with **CharSet** and **CharPartition** commands to specify individual genes and the partition, respectively.

```
#nexus
begin sets;
    charset part1 = 1-100;
    charset part2 = 101-384;
    charpartition mine = HKY+G:part1, GTR+I+G:part2;
end;
```

Example of nexus-style partition file. HKY+G model is applied to the first partition, while GTR+I+G is applied to the second partition.

Note

*It is very easy to make typos in this file (missing or extra semicolons, commas, spaces, etc), so if you run into any errors during your AU test and are unsure what the issue is, double/triple-check this file first.*

22 Save your nexus-style partition file as a .nex file (e.g. "partitions.nex")

Note

*IQTree also allows combining sub-alignments from different alignment files, which is helpful if you want to combine mixed data (e.g. DNA and protein) in a single analysis. You can see an example of this at http://www.iqtree.org/doc/Advanced-Tutorial (search "Partitioned analysis with mixed data").*

## Running the AU Test

23 It's easiest to proceed if your terminal working directory is the same one that IQTree is in. For ease of use, open the IQTree folder on your computer and place a copy of the following files:

- your combined trees in newick format ("_combined.txt")
- your nucleotide dataset (concatenated) in phylip format ("nuc_concat.phy")
- your partitions for above dataset ("partitions.txt" or "partitions.nex")

Note

*Alternatively, you can keep your files where they are if you make sure you use **absolute paths** to all files and programs involved (including IQTree, your files, etc).*

24 Use the following code to run the AU Test with a likelihood approximation:

For **partitioned data that uses a single model:**

Command

Note: GTR+I+G is an example model. Other models can be specified in the format MODEL+FreqType+RateType

```
iqtree -s nuc_concat.phy -m GTR+G+I -q partitions.txt -z combined.txt
-n 0 -zb 10000 -au
```

For **partitioned data using multiple models** (remember you will need the nexus-style partition file from Steps 18-20 for this):

Command

```
iqtree -s nuc_concat.phy -spp partitions.nex -z combined.txt -n 0 -zb
10000 -au
```

25 The results of the test are contained in the file ending in ".iqtree".