

# Conflict Workshop

Programs to be used: RaxML (<https://github.com/stamatak/standard-RAxML>)  
Phyx (<https://github.com/FePhyFoFum/phyx>)  
Astral (<https://github.com/smirarab/ASTRAL>)  
Figtree (<http://tree.bio.ed.ac.uk/software/figtree/>)  
Phyparts(<https://bitbucket.org/blackrim/phyparts>) # Included

## **Basic overview of sheet**

In bold I have put example commands for various programs and in the boxes are the commands that should be run for the workshop. The path to the programs may be a bit different but overall if you go step by step through everything in the box the workshop should run with no modifications.

## **Starting out**

The first step will be to download the workshop file and then unzip the folder

```
wget https://github.com/jfwalker/ConflictWorkshop/raw/master/ConflictWorkshop.zip  
unzip ConflictWorkshop.zip
```

Next move into the unzipped folder

```
cd ConflictWorkshop
```

## **Making the gene trees**

There should be a number of folders some of them are empty for now, the first step will be to move into the folder called Fastas

```
cd Fastas
```

These are all aligned fasta formatted files that are ready to be used to create the phylogenetic trees. To create the trees we will use maximum likelihood as implemented in RAxML and the RAxML command we will use works as follows.

**Example Command:** `raxml -T <number of threads> -p <Seed for parsimony tree> -m <Model of Evolution> -s <Alignment> -n <Output>`

Since we have five files we will run a bash forloop in order to make all the trees at once, the command is as follows

```
for x in *.aln; do raxml -T 2 -p 12345 -m PROTCATWAG -s $x -n $x;done
```

The output of this should be 5 gene trees, which will be the RAxML\_bestTree.Fasta\*.aln files. To keep everything in order we can next move the trees into the GeneTrees folder.

```
mv *bestTree* ../GeneTrees/
```

For the rest of the workshop we will not need the other RAxML files, it is usually a good idea to keep these for the sake of being able to replicate the experiment but to avoid having a large number of files we can delete them for the workshop.

```
rm RAxML*  
cd ..
```

## Making the Species Tree

### Maximum Quartet Support Species Tree

First we'll make a species tree using the Maximum Quartet Support Species Tree method as implemented in Astral.

To start create a directory called Astral

```
mkdir Astral
```

Next combine all the gene trees into as single file to use for astral and move that file to the Astral folder

```
cat GeneTrees/* > Astral_in.tre  
mv Astral_in.tre Astral/
```

Now move into the directory

```
cd Astral/
```

Now we can run astral

**Example Command:** `astral -i <input tree> -o <output tree>`

```
java -jar /usr/local/bin/ASTRAL/astral.5.6.2.jar -i Astral_in.tre -o Astral_out.tre
```

This creates the maximum quartet support species tree, it's likely that Astral told you not to pay attention to the posterior probabilities (support given) and that's because we did this with so little data.

Now we can leave the Astral directory and move to the next part

```
cd ..
```

The next step is to create the species tree using a concatenation method, which can be done by making a supermatrix out of the concatenated genes that we have.

First lets create a directory and copy the alignments into the new directory

```
mkdir Supermatrix  
cp Fastas/* Supermatrix/  
cd Supermatrix/
```

In order to concatenate the genes we can use the program pxcats as follows bellow. As a note Phyx will put AA as the model if it is amino acid, instead of a specified to allow the user to specify their own, the line below (which varies from mac to linux) will automatically change the AA's to the AUTO model.

**Example Command:** `pxcat -s <Sequence Files to be concatenated> -o <Outfile> -p <Partition File>`

```
pxcat -s *.aln -o Concatenated.aln -p partition.model  
Linux: sed -i 's/AA/AUTO/g' partition.model  
Mac: sed -i '.bak' 's/AA/AUTO/g' partition.model
```

The output of this will be a file called "Concatenated.aln" that contains the supermatrix, a file called "partition.model" which contains the gene partitions and a Phyx log file where the order of the concatenated genes can be found.

Now that we have the sequences concatenated together we can use RAxML to create the species tree. The use of "PROTCATAUTO" indicates that RAxML will select the best fitting model to the AA data. This is more computationally intensive, as opposed to using a predetermined model such as WAG, but has the benefit or possibly resulting in a better likelihood score.

```
raxml -T 2 -p 12345 -m PROTCATAUTO -s Concatenated.aln -q partition.model -n  
SpeciesTree
```

Now we should have a species tree called RAxML\_bestTree.SpeciesTree, for this we can move the species tree back one level.

```
mv RAxML_bestTree.SpeciesTree ..  
cd ..
```

### **Checking the Trees**

Now lets take a brief moment to look at the tree produced by Astral and the tree produced by the supermatrix ML.

```
figtree RAxML_bestTree.SpeciesTree Astral/Astral_out.tre
```

If you root this on Beet and Spinach you'll notice that the two species tree methods resulted in different topologies regarding Polygonaceae relationship with Tamaricaceae+Frankeniaceae. This creates an interesting problem and can be further analyzed by looking at the gene tree conflict in the next section.

### **Preparing Trees for Conflict Analysis**

For a bipartition based conflict analysis trees should be rooted! If you use genes that don't have outgroups you should not check them for conflict, if you don't have all the outgroups in your genes, you can use the ranked rooting "-r" option from pxrr.

Now in order to do the conflict analysis we need to reroot the gene trees and the supermatrix species tree. We could and should do also the Astral tree but for the sake of this guide we'll stick with just doing the supermatrix tree. To re-root the species tree we can do it as follows using pxrr

**Example Command: pxrr -t <tree(s) to be rooted> -g <outgroups> -o <outfile>**

```
pxrr -t RAxML_bestTree.SpeciesTree -g Beet,Spinach -o RAxML_bestTree.SpeciesTree.rr
```

Now we can do the same thing for the Gene Trees created at the beginning. Then we'll move them to the GeneTreesForAnalysis folder

```
for x in GeneTrees/*;do pxrr -t $x -g Beet,Spinach -o $x.rr;done  
mv GeneTrees/*.rr GeneTreesForAnalysis/
```

### **Running Phyparts**

Now that both the gene trees and the species tree are rooted and the outgroups removed we can go ahead and conduct the conflict analysis using phyparts.

**Example Command:** `java -jar phyparts -a <analysis type> -d <directory with trees> -m <mapping tree> -o <output> -v`

```
java -jar phyparts/phyparts-0.0.1-SNAPSHOT-jar-with-dependencies.jar -a 1 -d
GeneTreesForAnalysis/ -m RAxML_bestTree.SpeciesTree.rr -o ConflictResults/WorkShop -v
```

These results will be a mixture of the genes that are concordant and the ones that are conflicting. The file Workshop.concon.tre contains the trees with the concordant gene trees, the conflicting gene trees and the ICA scores mapped on the tree. You can separate them out with the following commands.

```
cd ConflictResults
head -n 1 WorkShop.concon.tre > Concordant.tre
head -n 2 WorkShop.concon.tre | tail -n 1 > Conflict.tre
```

We can then open the Concordant.tre and Conflict.tre files using figtree and the node labels on the trees will represent the number of gene trees with concordant signal and the number of gene trees with conflicting signal. For more information on what the other files are you can visit:

<https://bitbucket.org/blackrim/phyparts>

In figtree be sure to check the conflict that occurred at the node which changed places between Astral and the Supermatrix.

## Quartet support

Next we will add some support to the tree using the quartet method.

First move back a directory and clone the quartet directory

```
cd ..
git clone https://github.com/FePhyFoFum/quartetsampling.git
```

Convert your matrix to phylip for the quartet sampler

```
pxs2phy -s Supermatrix/Concatenated.aln -o Concatenated.phy
```

Now run the quartet sampling program

**Example Command:** `python3 quartetsampling/pysrc/quartet_sampling.py -t <SpeciesTree> -a <Supermatrix in phylip> -N <replicates> -L <likelihood difference> -T <threads> -d <molecule type> -X <raxml location> -o <Result Folder>`

```
python3 quartetsampling/pysrc/quartet_sampling.py -t RAxML_bestTree.SpeciesTree -a  
Concatenated.phy -N 200 -L 2.0 -T 2 -d amino -X raxml -o QuartetResults
```

As a group we'll go through what these results mean in more details but for info about interpretation please see:

Pease, James B., et al. "Quartet Sampling distinguishes lack of support from conflicting support in the green plant tree of life." *American journal of botany* 105.3 (2018): 385-403.