QuIBL analysis tutorial

Nate Edelman

3/02/2020

This is a tutorial for how to analyze output from the Quantifying Introgression via Branch Lengths software. For more details on QuIBL see http://github.com/michaelmiyagi/QuIBL. To run this tutorial, you will need to have cloned QuIBL and quiblR from github, and intalled their dependencies.

To run QuIBL with a sample dataset, navigate to the directory where quiblR is cloned. We'll start with the data in tutorial/intro_sim_example

```
cd "/Users/nbedelman/Dropbox/quiblR/"
cd tutorial/intro_sim_example
```

Open the file "sampleInputFile.txt" in your text editor of choice

We can see that QuIBL is very simple to run; all we need is a file that has a list of gene trees (one per line), and a taxon that we can use to root the trees ("totalOutgroup"). The rest of the parameters are described at http://github.com/michaelmiyagi/QuIBL, but we don't need to mess with them now. To run QuIBL, supply whatever output name you'd like as the "OutputPath", then return to terminal. From the tutorial/intro_sim_example directory, run

```
#python <path_to_QuIBL>/QuIBL.py sampleInputFile.txt
```

This will take a while, so feel free to run it, but we've also supplied the output here, so you can continue with the tutorial while you wait. Everything that follows should be run in R

Output Data Analysis

First, we'll set up our environment.

```
suppressMessages(library("quiblR"))
suppressMessages(library("ggplot2"))
suppressMessages(library("ape"))
suppressMessages(library("hash"))
suppressMessages(library("ggtree"))
suppressMessages(library("ggpubr"))
suppressMessages(library("dplyr"))
```

Load the data

Now, we'll import the data generated by our 4-taxon QuIBL run from simulated data. The important parameters here are: - We simulated 20% introgression fraction from species 3 into species 2, forward in time - Each population is comprised of 1 million individuals, the introgression time between 3 and 2 is ~2 million

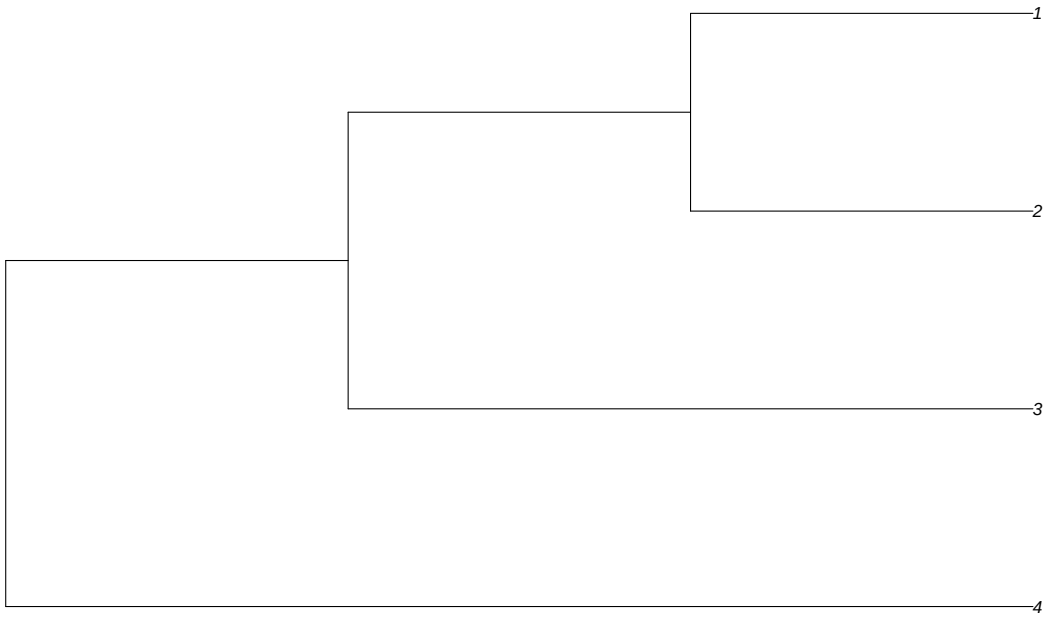
years ago (mya), the common ancestor between 1 and 2 is ~3 mya, and the common ancestor between 1,2, and 3 is ~6 mya. We expect this model to result in an intermediate level of ILS.

```
speciesTree <- read_speciesTree("tutorial/intro_sim_example/sampleSpeciesTree.nwk")
quiblOutput <- read_csv_quibl("tutorial/intro_sim_example/null_1000000_0.2.quibl")
originalTrees <- read.tree("tutorial/intro_sim_example/null_1000000_0.2.trees")
```

Examine the data

What does our data look like? To understand this, let's take a look at what we've simulated as the species tree.

```
plot(speciesTree)
```



Now, let's look at the QuIBL output

```
quiblOutput
```

```
##   triplet outgroup C1       C2  mixprop1  mixprop2 lambda2Dist lambda1Dist
## 1   1_2_3        1  0 2.331515 0.4177218 0.5822782 0.006035603 0.012106278
## 2   1_2_3        2  0 3.525483 0.8163298 0.1836702 0.003567823 0.005789102
## 3   1_2_3        3  0 1.900245 0.2698692 0.7301308 0.005768308 0.011052492
##    BIC2Dist   BIC1Dist count
## 1  -8349.00  -8254.857  1210
## 2  -2976.12  -3008.296   363
## 3 -24712.62 -24015.812  3427
```

To get familiar with our data, we'll walk through each column.

triplet

QuIBL examines each 3-species grouping in our data separately. But we have 4 species, so shouldn't there be (4 choose 3) = 4 different triplets? Why do we only have one (1_2_3)? This is because QuIBL assumes that we have a rooted tree, and in this case we have rooted our tree by fixing "4" as the outgroup. QuIBL discards all triplets that include the overall outgroup, since all loci are forced to have the same topology ("4" as the outgroup).

outgroup

The outgroup dictates the tree topology, as in three-taxon trees there is one outgroup and two ingroups. For example, when outgroup == 1, the tree topology is (1,(2,3)).

C1

The internal branch length in the ILS only model. By definition, this is 0.

C2

The internal branch length in the two-ditribution model. If the topology is concordant with the species tree (In this case, when the outgroup is Htel, see the tree), C2 is the time between the two coalescence events. If this topology is discordant with the species tree, C2 is the time between the introgression event and the common ancestor of all three species.

mixprop2

Under the two-ditribution model, the proportion of loci that have a history of introgression

mixprop1

Under the two-ditribution model, the proportion of loci that have a history of ILS

lambda2Dist

The population size parameter (theta/2) under the two-ditribution model

lambda1Dist

The population size parameter (theta/2) under the ILS only model

BIC2Dist

The Bayesian Information Criteria for the fit of the data to the two-ditribution model

BIC1Dist

The Bayesian Information Criteria for the fit of the data to the ILS only model

We only accept the two-distribution model if BIC2Dist is at least 10 units lower than BIC1Dist

count

The number of loci that have this topology.

Get some big-picture results

In order to interpret the output, we'll do some simple transformaions and report out our top-line results. First, we need to know which of our topologies is concordant with the species tree, and which represent either ILS or introgression. We'll use the isSpeciesTree fuction from quiblR.

```
quiblOutput <- mutate(quiblOutput, isDiscordant=as.integer(! apply(quiblOutput, 1, isSpeciesTree, sTree=
```

Next, we'll mark which of the topologies' 2-distribution model is a significantly better fit than the ILS only model, and we'll calculate the proportion of loci with a history of introgression for the full data

```
quiblOutput <- mutate(quiblOutput, isSignificant = as.numeric(apply(quiblOutput, 1, testSignificance, t
totalTrees <- sum(quiblOutput$count)/length(unique(quiblOutput$triplet))
quiblOutput <- mutate(quiblOutput,totalIntrogressionFraction=(mixprop2*count*isDiscordant)/totalTrees)
quiblOutput
```

```
##   triplet outgroup C1       C2  mixprop1  mixprop2 lambda2Dist lambda1Dist
## 1   1_2_3        1  0 2.331515 0.4177218 0.5822782 0.006035603 0.012106278
## 2   1_2_3        2  0 3.525483 0.8163298 0.1836702 0.003567823 0.005789102
## 3   1_2_3        3  0 1.900245 0.2698692 0.7301308 0.005768308 0.011052492
##    BIC2Dist   BIC1Dist count isDiscordant isSignificant
## 1  -8349.00  -8254.857  1210            1             1
## 2  -2976.12  -3008.296   363            1             0
## 3 -24712.62 -24015.812  3427            0             1
##   totalIntrogressionFraction
## 1                 0.14091133
## 2                 0.01333446
## 3                 0.00000000
```

So, what have we learned? For the big picture question of introgression, we're interested in whether there is significantly more support for the two-distribution (ILS+introgression) than the one-distribution model (ILS only). We see that one of our discordant topologies - outgroup=1, is both discordant and significant. We can then look at the inferred introgression fraction, which is ~14%. This is conservative compared to the simulated value of 20%.

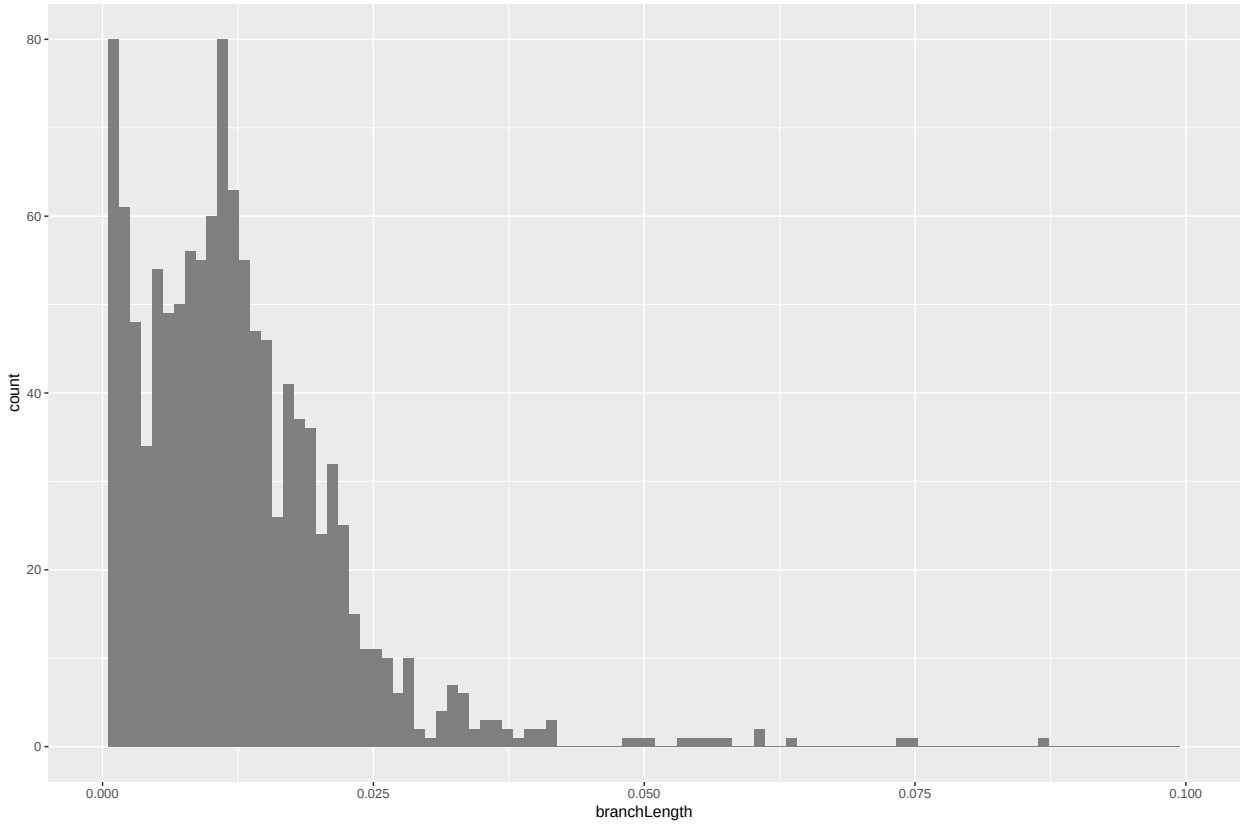
Evaluating model fit - does it pass the smell test

OK, let's take a quick look under the hood at the data that QuIBL is using to generate this output. The only data we're actually using is the internal branch length of triplet gene tress, grouped by topology. We can extract this information from the original tree file with the function getPerLocusStats. This will take a few moments.

```
perLocusStats <- getPerLocusStats(quiblOutput = quiblOutput, trip = "1_2_3", treeList = originalTrees, 
head(perLocusStats)
```

```
##                                                  tree out branchLength
## 1  ((1:0.02097764,2:0.0077361):0.01130941,3:0.03255054);   3   0.01130941
## 2  ((1:0.02778187,2:0.01743027):0.00552635,3:0.0253041);   3   0.00552635
## 3 ((1:0.01004192,2:0.01431998):0.02699448,3:0.03326814);   3   0.02699448
## 4 (2:0.02422109,(3:0.01742974,1:0.01854052):0.00032585);   2   0.00032585
## 5  ((1:0.01493838,2:0.01452438):0.00693963,3:0.0168215);   3   0.00693963
## 6 ((1:0.01332762,2:0.01222428):0.01125371,3:0.02874083);   3   0.01125371
##      introProb
## 1 0.9122753719
## 2 0.5795106366
## 3 0.9122753719
## 4 0.0006833402
## 5 0.7058876849
## 6 0.9122753719
```

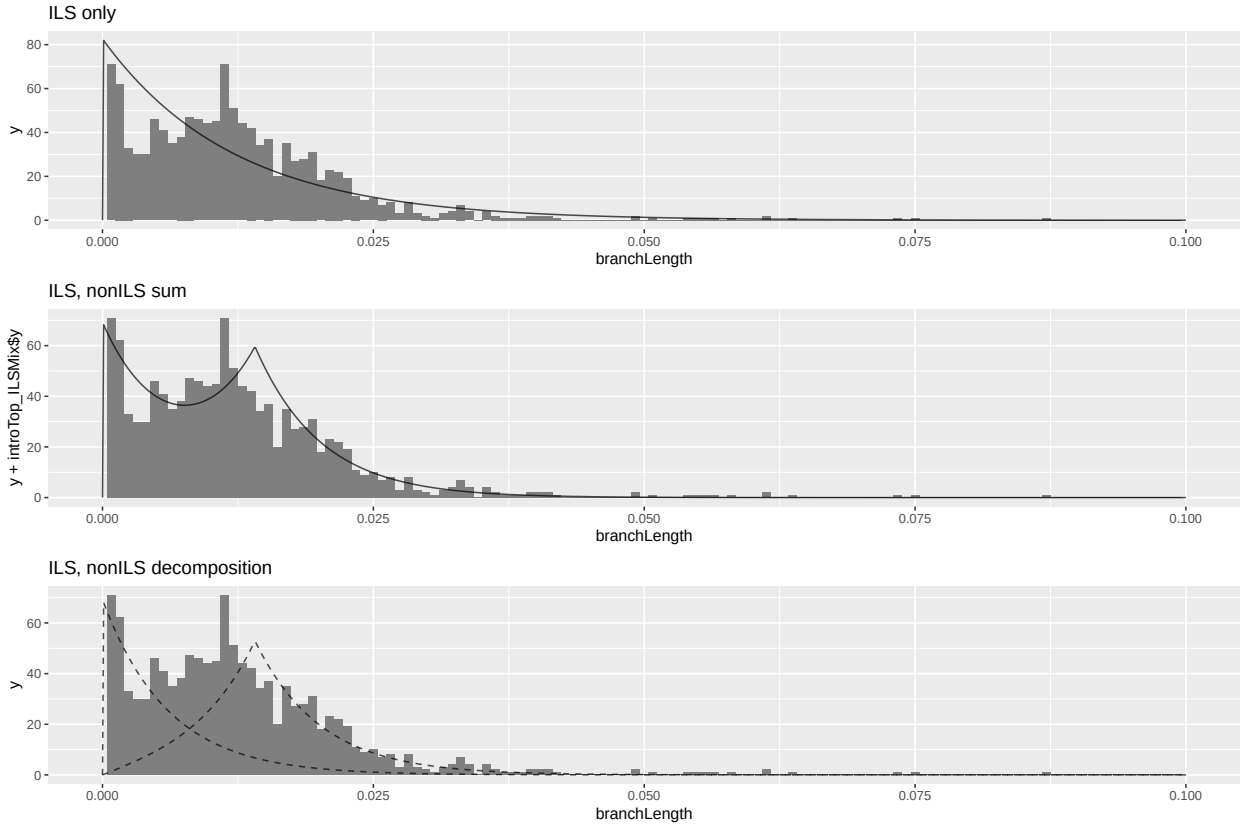
As we can see, perLocusStats is now a data frame that includes the triplet sub-tree, the outgroup, internal branch length, and introgression probability. Let's look at the internal branch lengths for our putative introgression topology, where "1" is the outgroup.



Next, we'll generate the distributions that QuIBL calculated based on this data. We'll get three distributions: The ILS only, ILS as a part of the 2-distribution mixture, and non-ILS as a part of the 2-distribution mixture.

```
introTop_ILSOnly <- getILSOnlyDist(0,0.1,subset(quiblOutput, outgroup=="1" & triplet=="1_2_3"))
introTop_ILSMix <- getILSMixtureDist(0,0.1,subset(quiblOutput, outgroup=="1" & triplet=="1_2_3"))
introTop_nonILSMix <- getNonILSMixtureDist(0,0.1,subset(quiblOutput, outgroup=="1" & triplet=="1_2_3"))
```

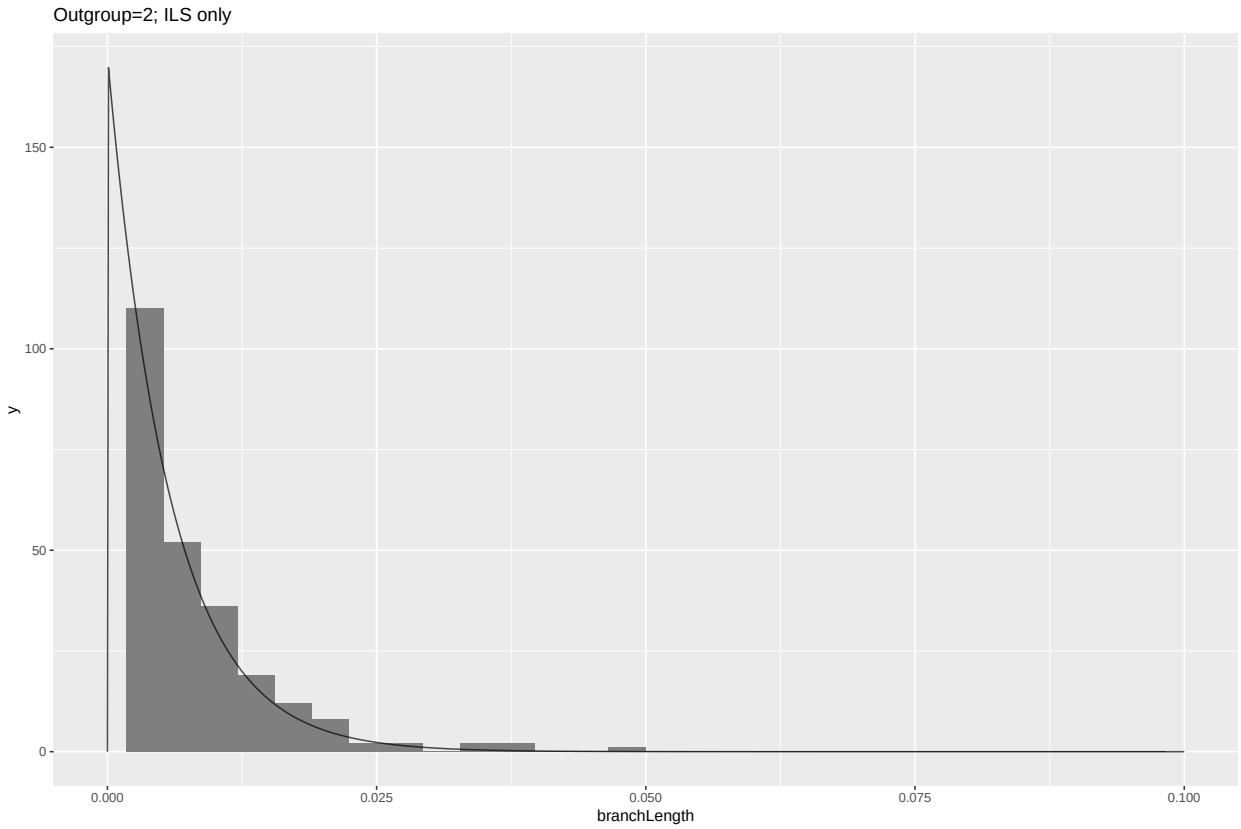
Now, we'll plot them over our data and see how they look. For the 2-distribution mixture, we'll plot each individual distribution as a dashed line, and the sum of the two as a solid line. The sum should hopefully match the data.



Not perfect, but not bad! Clearly, the ILS only model is a poor fit for our data. We can tell there are two peaks in our data, not a simple exponential decay. Looking at the sum of the mixture model (which is what we expect to observe), we can see that the introgression distribution nicely fits the right side of our data, though the peak is slightly overestimated.

So, why are we off, even if the 2-distribution model is clearly better than the ILS only? It could be a number of reasons. First, we have a finite number of datapoints here, but we're fitting a continuous model. We've tried to illustrate this limitation by plotting our data as a histogram instead of a smoothed density distribution so you can see how blocky it is. Second, though the sequences were simulated, we did then estimate the gene tree branch lengths with PhyML. That introduces some error as well.

We can follow this up by looking at the distribution of branch lengths that we expect to only arise due to ILS. There are many fewer of these, but we can still see if the distribution looks reasonable.



Larger datasets

This was a relatively simple, compact example, but what happens when we are analyzing a broader clade? Let's load in some real data from Edelman et al 2019.

```
largeSpeciesTree <- read_speciesTree("tutorial/heliconius_full_example/fullHeliconiusSpeciesTree.txt")
largeQuiblOutput <- read_csv_quibl("tutorial/heliconius_full_example/heliconius_5kTrees.quibl.csv")
largeOriginalTrees <- read.tree("tutorial/heliconius_full_example/heliconius_5kTrees.nwk")
```

Again, we'll add our useful columns as above.

```
totalTrees <- sum(largeQuiblOutput$count)/length(unique(largeQuiblOutput$triplet))

largeQuiblOutput <- mutate(largeQuiblOutput,
                           isDiscordant = as.integer(! apply(largeQuiblOutput, 1, isSpeciesTree, sTree=l
                           isSignificant = as.integer(apply(largeQuiblOutput, 1, testSignificance, thres
                           totalIntrogressionFraction=(mixprop2*count*isDiscordant)/totalTrees)
head(largeQuiblOutput)
```

```
##             triplet outgroup C1        C2    mixprop1   mixprop2 lambda2Dist
## 1 HeraRef_Hhim_Hhsa  HeraRef  0  5.752658 0.929079196 0.07092080  0.01783054
## 2 HeraRef_Hhim_Hhsa     Hhim  0 28.078017 0.989583332 0.01041667  0.01712382
## 3 HeraRef_Hhim_Hhsa     Hhsa  0  1.651121 0.006806734 0.99319327  0.01234374
## 4 HeraRef_Hhim_Htel  HeraRef  0 21.302267 0.981818157 0.01818184  0.02883111
## 5 HeraRef_Hhim_Htel     Hhim  0 27.659891 0.986111107 0.01388889  0.01648965
```

```
## 6 HeraRef_Hhim_Htel     Htel  0  1.724460 0.008329904 0.99167010  0.01472461
##   lambda1Dist BIC2Dist   BIC1Dist count isDiscordant isSignificant
## 1  0.02505619 2.417914  -6.395419    83            1             0
## 2  0.02213219 2.365686  -6.631231    96            1             0
## 3  0.02561019 6.256734 -10.440968  4821            0             0
## 4  0.03999780 2.221026  -5.601384    55            1             0
## 5  0.02282440 2.064693  -6.321514    72            1             0
## 6  0.03142656 6.398756 -10.291871  4873            0             0
##   totalIntrogressionFraction
## 1               0.0011772971
## 2               0.0002000020
## 3               0.0000000000
## 4               0.0002000023
## 5               0.0002000021
## 6               0.0000000000
```

Now, we have lots of results...60 rows in all. That's not quite as easy to make sense of as a data frame, so let's try to summarize the data. We'll use quiblR's getIntrogressionSummary function. This will return a data frame with the average introgression fraction for each pair of taxa. Specifically, it goes triplet by triplet, ignores topologies concordant with the species tree, and records mixprop2*count/total trees, and averages that value across all tests that include each pair of species.

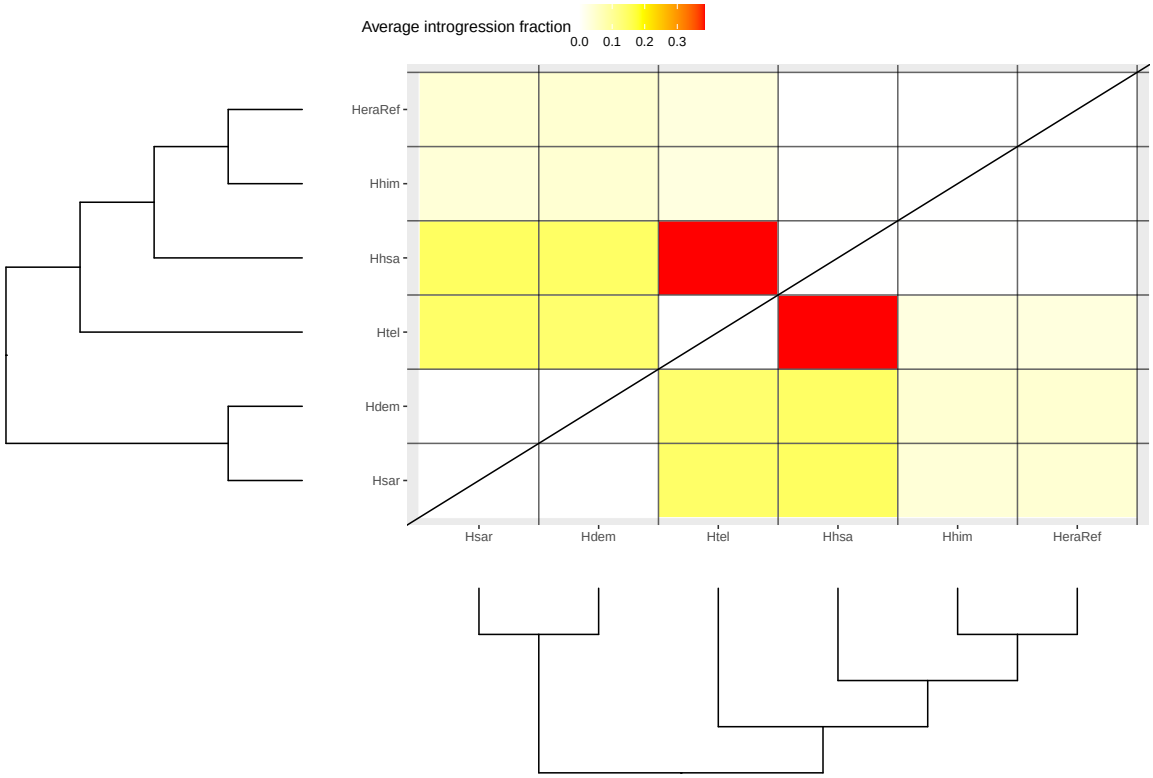
```
introgressionSummary <- getIntrogressionSummary(largeQuiblOutput,largeSpeciesTree)
#introgressionSummary$value <- as.numeric(ifelse(introgressionSummary$tax1==introgressionSummary$tax2,N
head(introgressionSummary)
```

```
##    tax1 tax2      value
## 35 Hsar Hsar 0.00000000
## 36 Hsar Hsar 0.00000000
## 4  Hsar Hdem 0.00000000
## 40 Hsar Htel 0.14345883
## 32 Hsar Hhsa 0.14923579
## 24 Hsar Hhim 0.04008875
```

We can now make a heatmap of these relationships.

```
summaryMatrix <- ggplot(data = introgressionSummary, aes(tax1, tax2, fill = value))+
 geom_tile(color = "white")+
 scale_fill_gradient2(low = "white", high = "red", mid = "yellow",na.value = "grey50",
   midpoint = max(introgressionSummary$value)/2, limit = c(0,max(introgressionSummary$value)),
   name="Average introgression fraction") +
geom_abline(slope = 1, intercept=0)+
  geom_vline(xintercept=seq(1.5,nrow(introgressionSummary)+0.5,1), alpha=0.6)+
  geom_hline(yintercept=seq(1.5,nrow(introgressionSummary)+0.5,1), alpha=0.6)+
  labs(x="", y="")+
  #scale_x_discrete(position = "top") +
 theme(panel.grid = element_blank(),legend.position = "none")


#summaryMatrix

speciesTreeSubset <- ggtree(extractTripletTree(largeSpeciesTree, unique(introgressionSummary$tax1)))
```

```
speciesTreeSubset_down <- ggtree(extractTripletTree(largeSpeciesTree, unique(introgressionSummary$tax1))

#speciesTreeSubset

ggarrange(  speciesTreeSubset, summaryMatrix,NULL, speciesTreeSubset_down,
          ncol = 2,nrow=2, heights=c(2,1), widths=c(1,2), align="hv",common.legend = TRUE)
```



It looks like we have the most evidence for introgression involving *H. hecalesia* (Hhsa) and *H. telesiphe* (Htel). Let's check out a triplet that would be informative for checking for introgression between these two. We'll use HeraRef_Hhsa_Htel, which estimates that when *H. erato* (HeraRef) is the outgroup, ~94% of all topologies arose via introgression (mixprop2), corresponding to an overall introgression fraction in the genome of ~38% (totalIntrogressionFraction).

```
subset(largeQuiblOutput, triplet=="HeraRef_Hhsa_Htel")
```

```
##              triplet outgroup C1        C2   mixprop1  mixprop2 lambda2Dist
## 13 HeraRef_Hhsa_Htel  HeraRef  0 0.8290148 0.05924158 0.9407584 0.009259183
## 14 HeraRef_Hhsa_Htel     Hhsa  0 0.2872300 0.32633967 0.6736603 0.011872018
## 15 HeraRef_Hhsa_Htel     Htel  0 0.6830959 0.05569463 0.9443054 0.010743734
##    lambda1Dist BIC2Dist   BIC1Dist count isDiscordant isSignificant
## 13  0.01350030 5.127408 -10.007675  2032            1             0
## 14  0.01316693 3.759472  -8.592181   486            1             0
## 15  0.01472817 5.390597 -10.154336  2482            0             0
##    totalIntrogressionFraction
## 13                 0.38232805
## 14                 0.06548044
## 15                 0.00000000
```

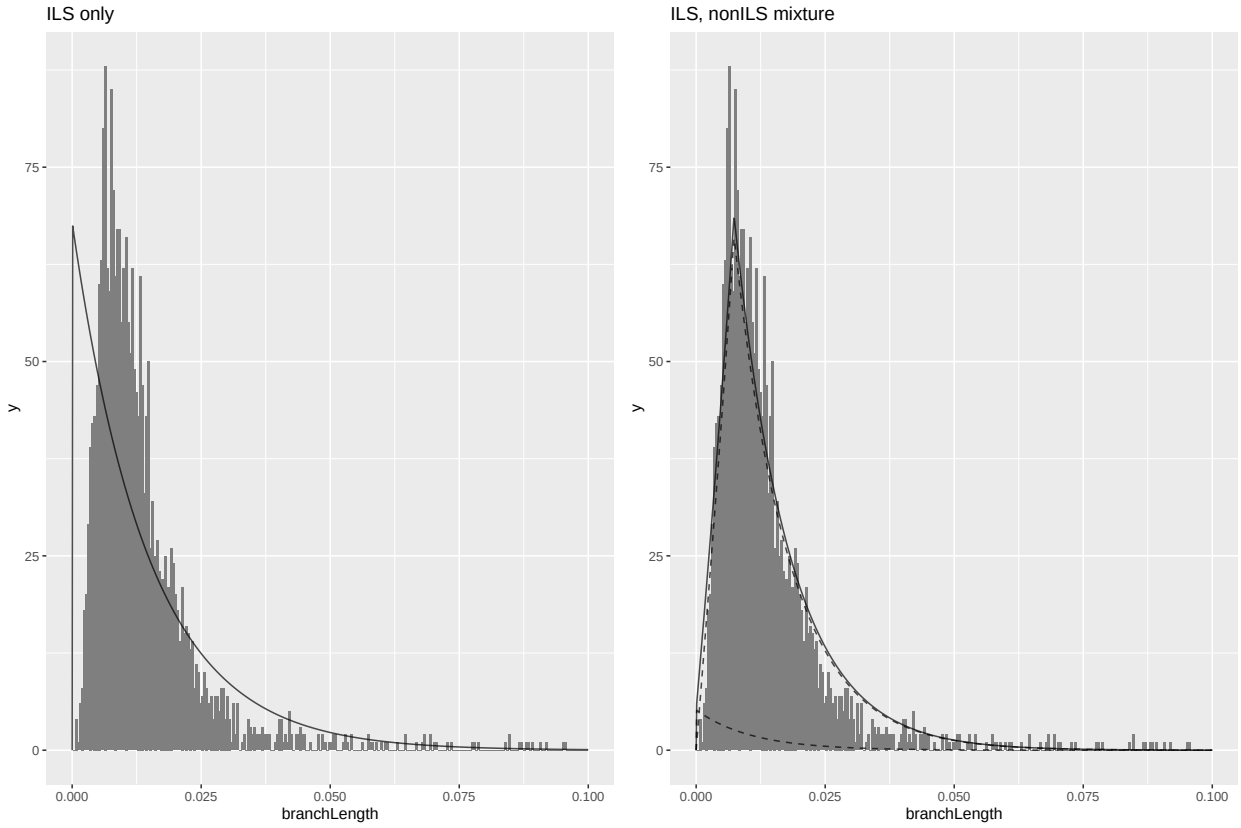
```
Hhsa_Htel_Hera_LocusStats <- getPerLocusStats(quiblOutput = largeQuiblOutput, trip = "HeraRef_Hhsa_Htel
head(Hhsa_Htel_Hera_LocusStats)
```

```
##                                                                    tree
## 1    (HeraRef:0.0593257662,(Hhsa:0.033715788,Htel:0.040142383)94:0.008386)55;
## 2    (HeraRef:0.0485842598,(Hhsa:0.0446369106,Htel:0.0459084911)97:0.007069);
## 3 (HeraRef:0.0564384558,(Hhsa:0.0306989322,Htel:0.0232391863)100:0.011257)54;
## 4  (HeraRef:0.0549916381,(Hhsa:0.035454053,Htel:0.0254074603)100:0.012668)48;
## 5   (HeraRef:0.049050774,(Hhsa:0.0239087842,Htel:0.018350125)100:0.033679)72;
## 6  ((HeraRef:0.0295762653,Hhsa:0.0272504644)78:0.003553,Htel:0.0288282716)74;
##       out branchLength introProb
## 1 HeraRef     0.008386 0.9631419
## 2 HeraRef     0.007069 0.9568302
## 3 HeraRef     0.011257 0.9631419
## 4 HeraRef     0.012668 0.9631419
## 5 HeraRef     0.033679 0.9631419
## 6    Htel     0.003553 0.8902313
```

Next, we'll generate the distributions

```
Hhsa_Htel_Hera_ILSOnly <- getILSOnlyDist(0,0.1,subset(largeQuiblOutput, outgroup=="Htel" & triplet=="He
Hhsa_Htel_Hera_ILSMix <- getILSMixtureDist(0,0.1,subset(largeQuiblOutput, outgroup=="Htel" & triplet=="
Hhsa_Htel_Hera_nonILSMix <- getNonILSMixtureDist(0,0.1,subset(largeQuiblOutput, outgroup=="Htel" & trip
```

Now, we'll plot them over our data and see how they look



OK, that looks pretty good! We can be confident in our estimated introgression fraction.