物种树构建后的基因渗入检验方法示例文档

丁颖，纪繁迪，黄华腾

很多系统发育基因组学分析会在物种树构建之后检验物种间是否存在基因渗入。目前常用的适用于系统发育数据集的几种基因渗入检验方法有：基于位点模式统计量、基于基因树信息的检测方法和 PhyloNet。本示例文档用模拟数据来展示这些方法的使用步骤和结果解读。

模拟数据生成

真实物种分化历史：我们指定了一棵 5 个物种–A、B、C、D、E–的物种树，其中 E 为外群， 物种 AC 之间有基因渗入，用 Newick 格式网络树表示为“((((A:4)#H1:2::0.6,B:6):4,((C:4,#H1:0::0.4):4,D:8):2):8,E:18);”。其中(A,C)网络边的遗传比例为 0.4，枝长单位为 2N（N 是有效群体大小）。

模拟生成基因树：我们采用 ms 生成 800 棵基因树，命令如下：

ms 5 800 -T -I 5 1 1 1 1 1 -es 2.0 3 0.6 -ej 2.0 4 6 -ej 3.0 3 5 -ej 4.0 6 2 -ej 5.0 5 2 -ej 9.0 2 1

模拟生成序列：我们采用 Seq-Gen 生成长度为 1000bp 的序列，碱基替换模型使用 HKY 模型，转换颠换比率设定为 3，群体遗传多样性为 0.05，命令如下：

seq-gen -mHKY -l1000 -s0.05 -t3 -q < genetree.tre > sequence.phy

从模拟序列估计基因树：采用 IQTree,设置参数-m MFP 使其自动测试并选择最优替代模型构建基因树，执行 1000 次超快自展值，这里使用两个线程的命令如下：

iqtree -s sequence.phy -m MFP -bb 1000 -nt AUTO -ntmax 2

至此，我们得到了模拟生成的序列和基因树。

一、基于位点模式统计量的检验方法

此类方法统计不同位点模式的出现频率，用这些频率值进行基因渗入检验。

1.1 *D* 统计量

S1: 准备计算 *D* 统计量的输入文件，包括已标明可变位点和信息位点的基因组序列文件、需要测试的 quartet 组合（格式如下）；

{“p4”: [“E”], “p3”: [“C”],“p2”: [“A”],“p1”: [“B”]}

S2: 从 https://github.com/YingDings/Introgression-Detection-Methods 下载 ABBA-BABA.py，

S3: 执行 ABBA-BABA 检验，命令为：“python ABBA-BABA.py”；

S4: 读取 ABBA-BABA 检验的结果文件“result.abba-baba.csv”和物种组合文件“taxa.abba-baba.csv”。

```
options(stringsAsFactors = F)
outcome_ABBA <- read.csv("result.abba-baba.csv")
tip_ABBA <- read.csv("taxa.abba-baba.csv")
quartet<-apply(tip_ABBA[,2:5], 1, function(x){paste0(x,collapse ="")})
outcome_ABBA$quartet<-quartet
outcome_ABBA
```

```
##   X     dstat  bootmean    bootstd        Z ABBA BABA nloci
## 1 0 0.1040258 0.1041721 0.01017481 10.22385 8899 7222   800
##               quartet
## 1 ['B']['A']['C']['E']
```

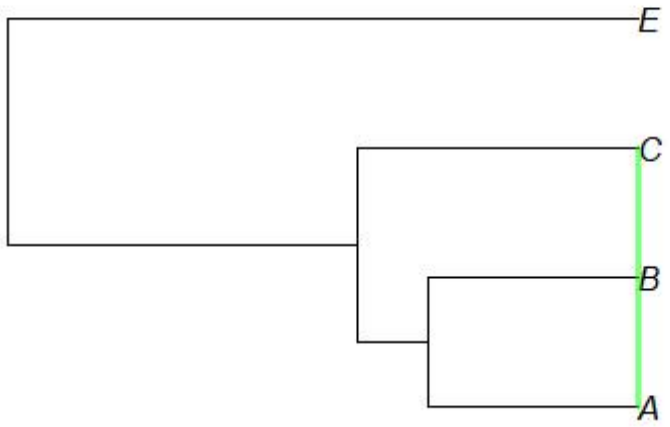
结果显示 ABBA 和 BABA 位点模式的数量分别为 8899 和 7222,且 Z-score 值大于 3，表明物种 A 与 C 之间存在基因渗入。

S5: 这里我们提供一个 Dplot 函数进行结果的可视化。用户可以从 https://github.com/YingDings/Introgression-Detection-Methods 下载该函数（D-plot.R），并载入 R 环境；

```
source("Introgression-Detection-Methods-main/D-plot.R")
```

S6: 指定物种树“(((A:6,B:6):4,C:10):8,E:18);”，调用 D-plot.R 中的 Dplot 函数可视化结果：

```
tree="(((A:6,B:6):2,C:8):10,E:18);"
network <- Dplot(tree,Doutcome=outcome_ABBA,taxa=tip_ABBA)
plot(network,col = "green",lty = 1,lwd = 3)
```



可视化物种 A 和 C 之间的基因渗入。

1.2 f 统计量

S1: 通过如下公式计算 f 统计量,同时由 1.2 节 ABBA-BABA 检验结果已知 ABBA 和 BABA 位点模式的数量分别为 8899 和 7222；

$$f_{hom} = \frac{S(S_1,S_2,S_3,O)}{S(S_1,S_3,S_3,O)}$$

S2: 计算公式分母，即符合 AABA 和 ABBA 位点模式的数量；依次读入 800 个只包含 ABCE 四个物种的基因序列文件，并计算符合 AABA 位点模式的位点数量。

```
library(stringr)
files<-dir("./data_modify/")
id<-which(str_detect(files,"phy"))
locus_num <- 0#AABA 位点模式的数量
for(i in 1:length(id)){
  data <- read.table(paste0("./data_modify/",files[id[i]]))
  sites <- data$V2[2:length(data$V2)]
  sites <- strsplit(as.character(sites),"")#sites 为各个位点的碱基
  for (j in 1:length(sites[[1]])){
    site <- vector()
    for (m in 1:length(sites)) {
      site[m] <- sites[[m]][j]
    }
```

```
    if(length(unique(site[-4]))==1&length(unique(site))==2){
      locus_num <- locus_num+1
    }
  }
}
print(paste("AABA:",locus_num))
```

```
## [1] "AABA: 60353"
```

结果显示 AABA 位点模式的数量为 60353.

S3: 结合 S1 和 S2 的结果，计算 f 统计量；

```
f_statistic <- (8899-7222)/(60353+8899)
print(f_statistic)
```

```
## [1] 0.02421591
```

f 统计量为 0.02421591，不等于零，证明物种 A 与 C 之间存在基因渗入，但是和其它研究模拟结果一样，如果不是近期的基因渗入，f 统计量存在低估渗入比例的现象。

1.3 D_{FOIL} 统计量

S1: 生成D_{FOIL}检验可识别的计数文件,命令为：“python3 fasta2dfoil.py 800locus_combine.fasta –out 800locus_combine.txt –names A,B,C,D,E”；

S2: 将 S1 生成的 800locus_combine.txt 作为输入文件，计算D_{FOIL}，命令为：“python dfoil.py –infile 800locus_combine.txt –out Dfoil.txt”；

S3: 分析D_{FOIL}检验结果。读入结果文件“Dfoil.txt”，通过 Pvalue 值判断结果是否具有显著性。

```
result<-scan("Dfoil.txt",what=character(),sep="\n")
ele <- unlist(str_split(result,pattern = "\t"))
result<-matrix(ele,nrow=2,byrow=T)
colnames(result)<-result[1,]
result<-as.data.frame(result)
result<-result[-1,]
Pvalue <- result[,c("DFO_Pvalue","DIL_Pvalue","DFI_Pvalue")]
Pvalue
```

```
##   DFO_Pvalue DIL_Pvalue DFI_Pvalue
## 2        0.0        0.0        0.0
```

Pvalue 为 0 这一结果显著支持五分类单元物种树中存在基因渗入。

二、基于基因树信息的检验方法

此类方法均针对三个物种的基因树进行检验。为了方便起见，我们这里将数据集中的物种 D 删除，只保留物种 E 作为外群。

```
library(ape)
gtrees=read.tree("800locus_iqtree_treefile_root.tres")#读入有根基因树
gtrees<-lapply(gtrees, function(gt){keep.tip(gt,c("A","B","C","E"))})#删除所有基因树中的物种D
class(gtrees)<-"multiPhylo"
gtrees<-read.tree(text=write.tree(gtrees))
stree="(((A,B),C),E);"
sptree<-read.tree(text=stree)
```

2.1 卡方检验并可视化结果

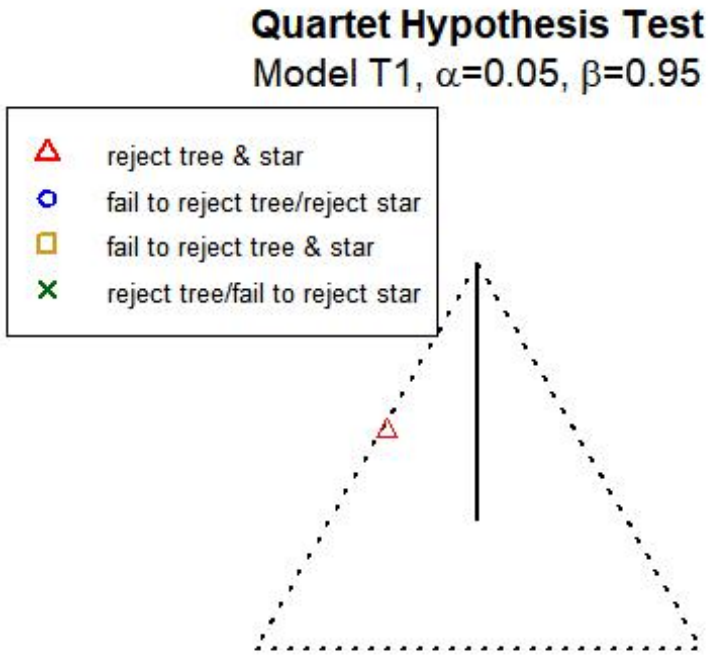
在 MSC 模型下，给定物种树，对所有 quartets 频数进行多重独立假设检验。

```
library(MSCquartets)
tnames=c("A","B","C","E")
QT=quartetTable(gtrees,tnames)
RQT=quartetTableResolved(QT)
pTable=quartetTreeTestInd(RQT,"T1",speciestree=stree)
pTable=quartetStarTestInd(pTable)
pTable
```

```
##      A B C E 12|34 13|24 14|23        p_T1 qindex       p_star
## [1,] 1 1 1 1   451   339    10 1.788626e-87      1 2.667303e-86
```

结果 qindex=1 表明与物种树拓扑一致的拓扑为 12|34，其频率为 451；另外两种与物种树不兼容的拓扑频率分别为 339 和 10，拓扑频率相差很大，不符合相等的理论预期。

```
quartetTestPlot(pTable, "T1", alpha=.05, beta=.95)
```



可视化结果显示有一个 quartet 频数不符合预期。

2.2 BLT

S1: BLT 方法目前没有现成的 R 包，这里我们提供一个 blt 函数进行该检验。用户可以从 https://github.com/YingDings/Introgression-Detection-Methods 下载该函数（run_blt.R），并载入 R 环境；

```
source("Introgression-Detection-Methods-main/run_blt.R")
```

S2: 调用 run_blt.R 中的 blt 函数计算拓扑中两个姐妹物种间的分支长度；

```
triplet<-c("A","B","C")
result<-blt(triplet,sptree,gtrees)
```

S3: 调用 run_blt.R 中的 wilcox.test 函数对 result 结果进行 Wilcoxon 秩和检验；

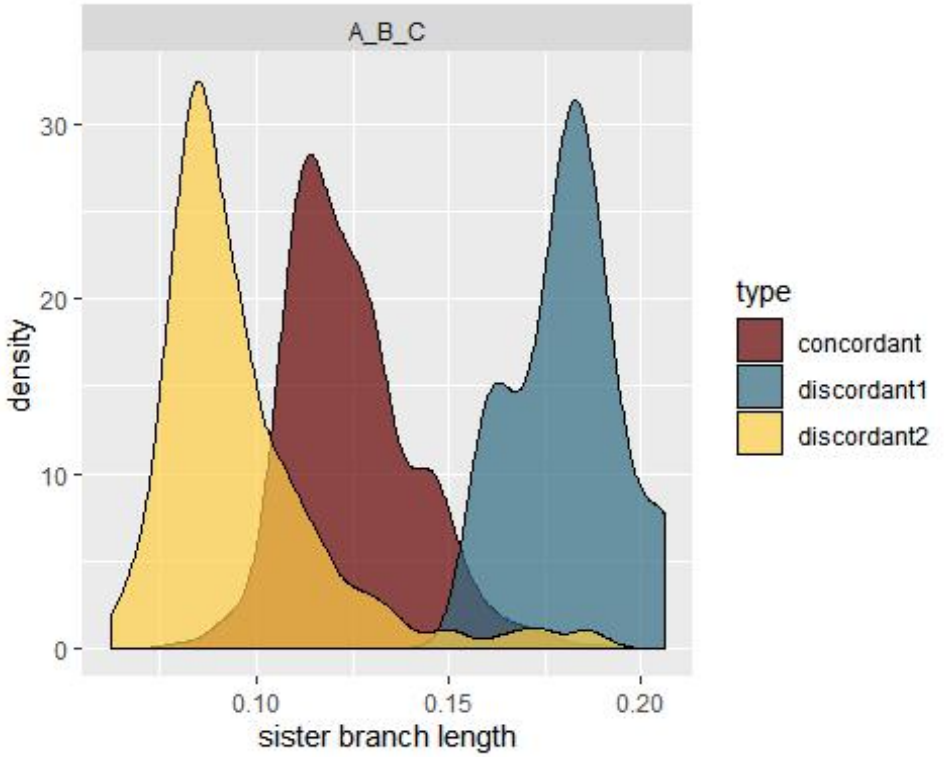
```
result_test<-wilcox_test(result,triplet,sptree)#指定只包含物种ABC的物种树上的外群
result_test

##   triplet Concor_sisBL Discor1_sisBL Discor2_sisBL wilcox_ConcorDisco
r1
## 1   A_B_C    0.2485182     0.3613987     0.1915359        9.754804e-0
8
##   wilcox_ConcorDiscor2 wilcox_Discor1Discor2
## 1        1.477765e-07         2.680235e-76
```

结果中 X_sisBL 表示所有三元拓扑中姐妹物种分支长度的平均值；wilcox_XX 表示两种类型的拓扑中姐妹物种分支长度的秩和检验的 Pvalue，其中 Concor 表示与物种树一致的拓扑；Discor1 表示与物种树拓扑不兼容的拓扑 1；Discor2 表示与物种树拓扑不兼容的拓扑 2。

结果显示 wilcox_Discor1Discor2 值远小于 0.05，表明 discordant1 中的姐妹物种分支长度与 discordant2 的姐妹物种分支长度有显著差别。

S4：可视化结果。

```
library(ggplot2)
ggplot(result, aes(x=((branchlength1/treelength)+(branchlength2/treelength))/2, fill=type))+geom_density(alpha=0.7)+scale_fill_manual(values=c("#630000", "#316B83","#FFCE45"))+facet_wrap(~triplet)+xlab("sister branch length")
```



结果显示 discordant2 拓扑中姐妹物种间遗传距离小于 discordant1 拓扑中的姐妹物种间遗传距离，因此 discordant2 拓扑中姐妹物种间存在基因渗入。

2.3 QuIBL

S1：准备 QuIBL 输入文件 inputfile.txt;

```
sink(file="inputfile.txt")
cat(paste0("[Input]","\n"))
cat(paste0("treefile: genetree.tres","\n"))#genetree.tres 为基因树文件
```

```
（不同文件夹须加路径）
cat(paste0("numdistributions: 2","\n"))
cat(paste0("likelihoodthresh: 0.01","\n"))#似然值变化阈值
cat(paste0("numsteps: 10","\n"))
cat(paste0("gradascentscalar: 0.5","\n"))
cat(paste0("totaloutgroup: E","\n"))#指定外群为E
cat(paste0("multiproc: True","\n"))
cat(paste0("maxcores:1000","\n"))
cat(paste0("[Output]","\n"))
cat(paste0("OutputPath: result.csv","\n"))#result.csv 为输出文件名
sink()
```

S2: 执行 QuIBL 方法，命令为："python QuIBL.py inputfile.txt"；

S3: 分析 QuIBL 结果。读入结果文件，计算两种分布模型的 BIC 值差 deltaBIC，根据其结果判断是否仅存在 ILS 或同时存在 ILS 与 Introgression。

```
result<-read.csv("result_quibl.csv")
result$deltaBIC<-result$BIC2Dist-result$BIC1Dist
result$type<-""
type=c("concordant","discordant1","discordant2")
t<-drop.tip(sptree,"E")
out<-t$tip.label[min(t$edge[t$edge[,1]==length(t$tip.label)+1,2])]
temp<-seq(from=1,to=nrow(result),by=3)
w<-which(result$outgroup[temp[1]:(temp[1]+2)]==out)
result$type[temp[1]:(temp[1]+2)][w]<-type[1]
result$type[temp[1]:(temp[1]+2)][-w]<-type[2:3]
result$result<-ifelse(result$type=="concordant" & result$deltaBIC < -30
 ,"Concordant",ifelse(result$deltaBIC< -30 & result$type!="concordant",
"ILS+Introgression",ifelse(result$type=="concordant" & result$deltaBIC
> -30,"Extreme ILS","ILS")))
result

##   triplet outgroup C1      C2   mixprop1  mixprop2 lambda2Dist lambda
1Dist
## 1   A_B_C        A  0 3.144553 0.07295113 0.9270489  0.01447052  0.045
23736
## 2   A_B_C        B  0 3.786348 0.03365159 0.9663484  0.07410281  0.274
92387
## 3   A_B_C        C  0 2.218411 0.01364282 0.9863572  0.08399392  0.208
13639
##     BIC2Dist   BIC1Dist count    deltaBIC       type           result
## 1  -43.35183  -39.61406    10   -3.737773 discordant1              IL
S
## 2 -588.02144 -191.64900   339 -396.372444 discordant2 ILS+Introgressi
on
## 3 -893.33633 -507.63317   451 -385.703165  concordant       Concorda
nt
```

结果显示 B 为外群时的三元拓扑为 ILS+Introgression。

三、PhyloNet

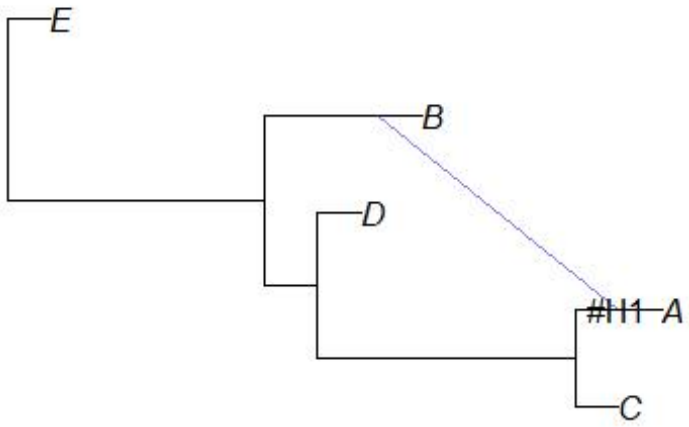
S1: 准备 PhyloNet 输入文件。文件格式为 Nexus，里面包括两个模块：含基因树的树模块和执行 PhyloNet 的命令模块；

```
gts<-readLines("800locus_iqtree_treefile_root.tres")#读入所有有根基因树
gts<-paste0("Tree gt",1:800,"=",gts)
#write file
sink("phylonet_InferNetwork_MPL_input.nex")
cat(paste0("#NEXUS","\n","\n"))
cat(paste0("BEGIN TREES;","\n","\n"))
cat(paste0(gts,"\n"))#写入基因树
cat(paste0("\n","END;","\n"))
cat(paste0("\n", "BEGIN PHYLONET;","\n","\n"))
cat("InferNetwork_MPL (all) 1 -pl 20 -di resultOutputFile phylonet_Infer
Network_MPL_out.tres;")
cat(paste0("\n","\n","END;","\n"))
sink()
```

S2: 执行 PhyloNet 命令："java -jar PhyloNet_3.8.2.jar phylonet_InferNetwork_MPL_input.nex"

S3: 读取 PhyloNet 结果，用 R 可视化似然值最大的网络树。

```
result<-readLines("phylonet_InferNetwork_MPL_out.tres")
network<-result[3]
network

## [1] "((((C:1.0,(A:1.0)#H1:1.0::0.4012780851740897):5.935140856165301,
D:1.0):1.2297544497301343,(#H1:1.0::0.5987219148259103,B:1.0):2.6232814
28512928):5.910181964983381,E:1.0);"

net<-read.evonet(text=network)
plot(net)
nodelabels(text=net$node.label,frame = "none")
```



结果显示物种 A 与 C 存在基因渗入，通过网络树 Newick 格式可以看出两者间的遗传比例为 40%。