

检测基因流示例-应用模拟数据

丁颖，纪繁迪，黄华腾

本示例文档详细介绍了基于位点模式统计量、基于基因树信息的检测方法和 PhyloNet 的步骤和结果。

模拟数据

真实基因树：给定 Network 格式网络

“((((A:4)#H1:2::0.6,B:6):2,(C:4,#H1:0::0.4):4):2,D:10):8,E:18);”，其中(A,C)网络边的遗传比例为 0.4。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
```

序列：在 HKY 模型下，指定群体突变率为 0.05，采用 Seq-gen 生成长度为 1000bp 的序列，基因树包含在 genetree.tre 文件中，命令如下：

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

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，这里我们指定最大线程数为 2。命令如下：

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

至此，我们得到了根据网络树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

1.1 D 统计量

1.2 f 统计量

1.3 D_{FOIL} 统计量

二、基于基因树信息的检验方法

此类方法均采用三个物种的拓扑进行检验，且因模拟基因流发生在 AC 物种间。因此，我们使基因树只包含 ABCE 四个物种，并给定物种树为“((((A,B),C),E);”，其中 E 为外群。

```
library(ape)
gtrees=read.tree("800locus_iqtree_treefile.trees")
gtrees<-lapply(gtrees, function(gt){keep.tip(gt,c("A","B","C","E"))})
stree="(((A,B),C),E);"
sptree<-read.tree(text=stree)
```

2.1 卡方检验并可视化结果

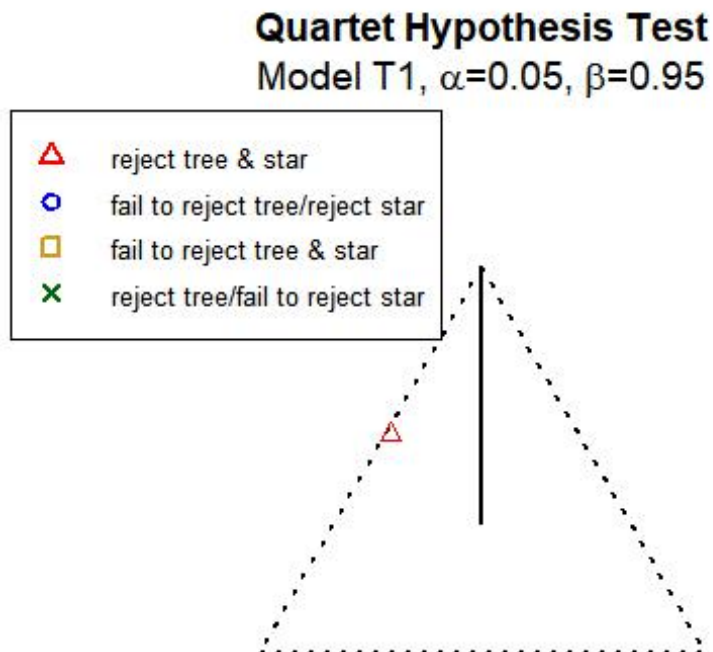
在 MSC 模型下，给定物种树，对所有 quartet 频数进行多重独立假设检验。

```
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，不兼容拓扑频率不符合相等的理论预期。

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



结果显示有一个 quartet 频数拒绝接受此物种树。

2.2 BLT

S1: 读入 run_blt.R 文件，为后续执行 BLT 方法做准备。用户可以从 <https://github.com/YingDings/Introgression-Detection-Methods> 获得；

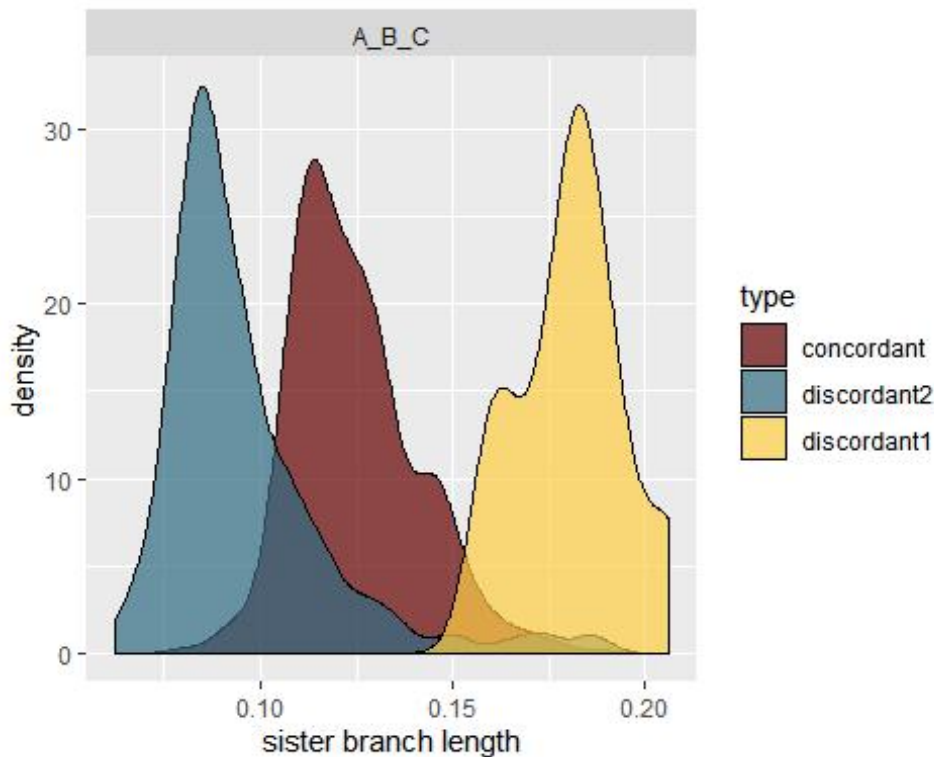
```
source("Introgression-Detection-Methods-main/run_blt.R")
```

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；

```
result_test<-data.frame(triplet=character(),outgroup=character(),frequency=character(),chisq=character(),concor_proxy_t=character(),discor1_proxy_t=character(),discor2_proxy_t=character(),wilcox_text_cd1=character(),wilcox_text_cd2=character(),wilcox_text_d1d2=character())
triplet<-c("A","B","C")
result<-blt(triplet,sptree,gtrees)
```

S3: 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 distcordant1 拓扑中的姐妹物种间遗传距离，因此基因渗入导致不兼容拓扑中 discordant2 有显著更小的平均外部枝长。

2.3 QuIBL

S1: 准备 QuIBL 数据文件,指定输入的基因树文件、测试分支长度分布数量、似然值变化停止阈值、外群、输出文件等参数;

```
sink(file="inputfile.txt")
cat(paste0("[Input]", "\n"))
cat(paste0("treefile: genetree.tres", "\n"))
cat(paste0("numdistributions: 2", "\n"))
cat(paste0("likelihoodthresh: 0.01", "\n"))
cat(paste0("numsteps: 10", "\n"))
cat(paste0("gradascentscalar: 0.5", "\n"))
cat(paste0("totaloutgroup: E", "\n"))
cat(paste0("multiproc: True", "\n"))
cat(paste0("maxcores:1000", "\n"))
cat(paste0("[Output]", "\n"))
cat(paste0("OutputPath: result.csv", "\n"))
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, 根据其结果判断仅存在 ILS 或同时存在 ILS 与 Introgression。

```
result<-read.csv("result.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	lambda1Dist
## 1	A_B_C	A	0	3.144553	0.07295113	0.9270489	0.01447052	0.04523736
## 2	A_B_C	B	0	3.786348	0.03365159	0.9663484	0.07410281	0.27492387

```
## 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 为外群的三元拓扑为 ILS+Introgression。

三、PhyloNet

S1: 准备 PhyloNet 输入文件,包括含有基因树的树模块和执行 PhyloNet 的命令模块;

```
gts<-readLines("800locus_iqtree_treefile.tres")
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"))
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()
```

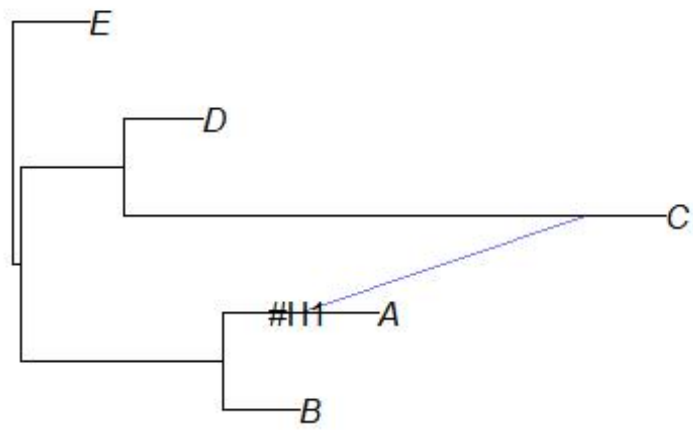
S2: 执行 PhyloNet 命令: “java -jar PhyloNet_3.8.2.jar
phylonet_InferNetwork_MPL_input.nex”

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] "(((B:1.0,(A:1.0)#H1:1.0::0.6011773749687261):2.5688535784776247,
((#H1:1.0::0.3988226250312738,C:1.0):5.927336212055351,D:1.0):1.3220653
51009499):0.12697359216218265,E:1.0);"
```

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



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