Evaluation 2: get table 1

Janne Pott

13/03/2023

Introduction

In our manuscript, we present the summary of the simulation study in Table 1. In part, this was done before, but here I want to present it a bit nicer. This includes for all analyzed n over all tested k with at least one phylogenetic set:

- Prevalence of phylogenetic decisive sets in simulation (overall, min, max)
- True Positive Rate (TPR, sensitivity, power) (overall, min, max)
- Negative Predictive Value (NPV) (overall, min, max)
- Minimal set size satisfying the 4-way partition property (4WPP) in the simulation
- Minimal set size which is fixing taxon traceable (FTT) in the simulation

In addition, I want a second table with more number, not necessarily out of the simulation

- Minimal set size satisfying the 4-way partition property
- Maximal set size not satisfying the 4-way partition property
- Minimal set size which is fixing taxon traceable
- Maximal set size which satisfy the 4-way partition property but is not fixing taxon traceable

Initialize

I use a file names *SourceFile.R* that contains all relevant R packages and user-/server-specific path to the R library. If using this code, you must make all the necessary changes within the template source file.

```
rm(list = ls())
time0<-Sys.time()

source("../SourceFile.R")
source("../helperFunctions/TestHelpRFunction.R")

x_lowerBound = c()
x_upperBound = c()

for(i in 6:10){
  #i=7
  y = i %% 6

  if(y==0){
```

```r
    min_quad = 0.25 * choose(i,3) + i/6
  }else if(y %in% c(2,4)){
    min_quad = 0.25 * choose(i,3)
  }else{
    min_quad = (1/6) * ((i-1)/2) * choose(i,2) + i/12
    min_quad = ceiling(min_quad)
  }
  x_lowerBound = c(x_lowerBound,min_quad)
  max_quad = choose(i,4) - (i-4)
  x_upperBound = c(x_upperBound,max_quad)
}

load("../results/02_SimulationResults_n06.RData")
load("../results/02_SimulationResults_n07.RData")
load("../results/02_SimulationResults_n08.RData")
load("../results/02_SimulationResults_n09.RData")
load("../results/02_SimulationResults_n10.RData")

SimulationResults_n06[,n := 6]
SimulationResults_n07[,n := 7]
SimulationResults_n08[,n := 8]
SimulationResults_n09[,n := 9]
SimulationResults_n10[,n := 10]

sim_n6 = SimulationResults_n06[k>=x_lowerBound[1] & k<=x_upperBound[1]]
sim_n7 = SimulationResults_n07[k>=x_lowerBound[2] & k<=x_upperBound[2]]
sim_n8 = SimulationResults_n08[k>=x_lowerBound[3] & k<=x_upperBound[3]]
sim_n9 = SimulationResults_n09[k>=x_lowerBound[4] & k<=x_upperBound[4]]
sim_n10 = SimulationResults_n10[k>=x_lowerBound[5] & k<=x_upperBound[5]]

sim = rbind(sim_n6,sim_n7,sim_n8,sim_n9,sim_n10)
table(sim$n)
#>
#>   6   7   8   9  10
#>   7  20  52  96 174
```

Get Table 1

```r
dumTab2 = foreach(i = 6:10)%do%{
  # i=6
  mySim = copy(sim)
  mySim = mySim[n == i,]
  stats = TestHelpRFunction(P = sum(mySim$NR_PhyloDec),
                            N = sum(mySim$NR_NotPhyloDec),
                            PP = sum(mySim$NR_FTT))
  stats[,n:=i]
  stats
}
myStats = rbindlist(dumTab2)
myStats
#>    Prevalence PPV       NPV       TPR TNR  n
```

```r
#> 1:  0.1192290   1 0.9802346 0.8510445   1  6
#> 2:  0.4541400   1 0.9728737 0.9664861   1  7
#> 3:  0.4524173   1 0.9833916 0.9795585   1  8
#> 4:  0.5146885   1 0.9877019 0.9882595   1  9
#> 5:  0.5270874   1 0.9934300 0.9940663   1 10

dumTab3 = foreach(i = 6:10)%do%{
  # i=6
  mySim = copy(sim)
  mySim = mySim[n == i,]

  dumTab4 = foreach(k = 1:dim(mySim)[1])%do%{
    # k=1
    myRow = copy(mySim)
    myRow = myRow[k,]

    stats_k = TestHelpRFunction(P = myRow$NR_PhyloDec,
                                N = myRow$NR_NotPhyloDec,
                                PP = myRow$NR_FTT)
    stats_k[,n :=i]
    stats_k[,k := myRow$k]
    stats_k
  }
  myStats_k = rbindlist(dumTab4)
  myStats_k
}
myStats_k = rbindlist(dumTab3)

x1 = myStats_k[Prevalence!=0,min(k),by = n]
x2 = myStats_k[Prevalence!=0 & TPR>0,min(k),by = n]

tab1 = copy(x1)
setnames(tab1,"V1","k_min_4WPP")
tab1[,k_min_FTT := x2$V1]

filt_TPR = !is.na(myStats_k$TPR)
tab6 = myStats_k[n==6 & filt_TPR,summary(TPR)]
tab7 = myStats_k[n==7 & filt_TPR,summary(TPR)]
tab8 = myStats_k[n==8 & filt_TPR,summary(TPR)]
tab9 = myStats_k[n==9 & filt_TPR,summary(TPR)]
tab10 = myStats_k[n==10 & filt_TPR,summary(TPR)]
tab_TPR = rbind(tab6,tab7,tab8,tab9,tab10)
tab1[,TPR := signif(tab_TPR[,3],2)]
tab1[,TPR_IQR := paste0("[",signif(tab_TPR[,2],2),",",signif(tab_TPR[,5],2),"]")]

filt_NPV = !is.na(myStats_k$NPV)
tab6 = myStats_k[n==6 & filt_NPV,summary(NPV)]
tab7 = myStats_k[n==7 & filt_NPV,summary(NPV)]
tab8 = myStats_k[n==8 & filt_NPV,summary(NPV)]
tab9 = myStats_k[n==9 & filt_NPV,summary(NPV)]
tab10 = myStats_k[n==10 & filt_NPV,summary(NPV)]
tab_NPV = rbind(tab6,tab7,tab8,tab9,tab10)
tab1[,NPV := signif(tab_NPV[,3],2)]
```

```r
tab1[,NPV_IQR := paste0("[",signif(tab_NPV[,2],2),",",signif(tab_NPV[,5],2),"]")]

tab6 = myStats_k[n==6 & Prevalence!=0,summary(Prevalence)]
tab7 = myStats_k[n==7 & Prevalence!=0,summary(Prevalence)]
tab8 = myStats_k[n==8 & Prevalence!=0,summary(Prevalence)]
tab9 = myStats_k[n==9 & Prevalence!=0,summary(Prevalence)]
tab10 = myStats_k[n==10 & Prevalence!=0,summary(Prevalence)]
tab_Prev = rbind(tab6,tab7,tab8,tab9,tab10)
tab1[,Prev := signif(tab_Prev[,3],2)]
tab1[,Prev_IQR := paste0("[",signif(tab_Prev[,2],2),",",signif(tab_Prev[,5],2),"]")]
tab1
#>     n k_min_4WPP k_min_FTT  TPR   TPR_IQR NPV   NPV_IQR Prev     Prev_IQR
#> 1:  6          9        10 0.97 [0.71,1]    1 [0.98,1] 0.62 [0.36,0.83]
#> 2:  7         17        20 0.99 [0.65,1]    1 [0.97,1] 0.74 [0.25,0.96]
#> 3:  8         30        35 1.00   [0.8,1]    1 [0.98,1] 0.85 [0.29,0.99]
#> 4:  9         46        56 1.00 [0.91,1]    1 [0.99,1] 0.91     [0.32,1]
#> 5: 10         72        84 1.00 [0.97,1]    1     [1,1] 0.95     [0.44,1]
knitr::kable(tab1)
```

n	k_min_4WPP	k_min_FTT	TPR	TPR_IQR	NPV	NPV_IQR	Prev	Prev_IQR
6	9	10	0.97	[0.71,1]	1	[0.98,1]	0.62	[0.36,0.83]
7	17	20	0.99	[0.65,1]	1	[0.97,1]	0.74	[0.25,0.96]
8	30	35	1.00	[0.8,1]	1	[0.98,1]	0.85	[0.29,0.99]
9	46	56	1.00	[0.91,1]	1	[0.99,1]	0.91	[0.32,1]
10	72	84	1.00	[0.97,1]	1	[1,1]	0.95	[0.44,1]

Get Table 2

- Minimal set size satisfying the 4-way partition property
- Maximal set size not satisfying the 4-way partition property
- Minimal set size which is fixing taxon traceable
- Maximal set size which satisfy the 4-way partition property but is not fixing taxon traceable

```r
x1 = myStats_k[Prevalence!=0,min(k),by = n]
y1 = myStats_k[Prevalence<1,max(k),by = n]
x2 = myStats_k[Prevalence!=0 & TPR>0,min(k),by = n]
y2 = sim[posRate<1,max(k),by = n]

y3 = c()
for(i in 6:10){
  #i=6
  dum = choose(i,4) - 3*i +13
  y3[i-5]=dum
}

tab2 = copy(x1)
setnames(tab2,"V1","k_min_4WPP")
tab2[,k_max_4WPP_sim := y1$V1]
tab2[,k_max_4WPP_theo := x_upperBound-1]
tab2[,k_min_FTT := x2$V1]
```

```r
tab2[,k_max_diff_sim := y2$V1]
tab2[,k_max_diff_theo := y3]

tab2 = t(tab2)
knitr::kable(tab2)
```

n	6	7	8	9	10
k_min_4WPP	9	17	30	46	72
k_max_4WPP_sim	12	31	62	113	183
k_max_4WPP_theo	12	31	65	120	203
k_min_FTT	10	20	35	56	84
k_max_diff_sim	10	26	50	89	146
k_max_diff_theo	10	27	59	112	193

Session Info

```r
sessionInfo()
#> R version 4.2.2 (2022-10-31 ucrt)
#> Platform: x86_64-w64-mingw32/x64 (64-bit)
#> Running under: Windows 10 x64 (build 22621)
#>
#> Matrix products: default
#>
#> locale:
#> [1] LC_COLLATE=German_Germany.utf8  LC_CTYPE=German_Germany.utf8
#> [3] LC_MONETARY=German_Germany.utf8 LC_NUMERIC=C
#> [5] LC_TIME=German_Germany.utf8
#>
#> attached base packages:
#> [1] grid      stats     graphics  grDevices utils     datasets  methods
#> [8] base
#>
#> other attached packages:
#> [1] cowplot_1.1.1           gtable_0.3.1            ggplot2_3.4.1
#> [4] FixingTaxonTraceR_0.0.1 foreach_1.5.2           data.table_1.14.8
#>
#> loaded via a namespace (and not attached):
#>  [1] rstudioapi_0.14  knitr_1.42       magrittr_2.0.3   munsell_0.5.0
#>  [5] colorspace_2.1-0 R6_2.5.1         rlang_1.0.6      fastmap_1.1.1
#>  [9] fansi_1.0.4      tools_4.2.2      xfun_0.37        utf8_1.2.3
#> [13] cli_3.6.0        withr_2.5.0      htmltools_0.5.4  iterators_1.0.14
#> [17] yaml_2.3.7       digest_0.6.31    tibble_3.2.0     lifecycle_1.0.3
#> [21] vctrs_0.5.2      codetools_0.2-18 glue_1.6.2       evaluate_0.20
#> [25] rmarkdown_2.20   compiler_4.2.2   pillar_1.8.1     scales_1.2.1
#> [29] pkgconfig_2.0.3
message("\nTOTAL TIME : " ,round(difftime(Sys.time(),time0,units = "mins"),3)," minutes")
#>
#> TOTAL TIME : 0.024 minutes
```