06 - Testing phyloflows sampling adjustments

Xiaoyue Xi and Oliver Ratmann

2019-04-30

To test the performance of **phyloflows** model for estimating transmission flows under biased sampling, we set up the following simulation exercise.

phyloflows MCMC on 100 simulated data sets

First, we simulated several data sets as "Data Set 1 (simple SARWS)". They can be loaded through:

```r
library(ggplot2)
library(data.table)
library(phyloflows)
data(twoGroupFlows100)
```

We then ran our MCMC algorithm on these 100 data set as follows:

```r
for (i in 1:100){
  dobs  <-  twoGroupFlows100[[i]][[1]]
  dprior_sarws  <-  twoGroupFlows100[[i]][[2]]
  dprior_sar  <-  twoGroupFlows100[[i]][[3]]
  randomnumber  <-  twoGroupFlows100[[i]][[4]]

  # number of iterations
  tmp  <- subset(dobs, select=c(TRM_CAT_PAIR_ID, TR_SAMPLING_CATEGORY, REC_SAMPLING_CATEGORY))
  tmp  <- melt(tmp, id.vars='TRM_CAT_PAIR_ID', value.name='SAMPLING_CATEGORY', variable.name='WHO')
  dlu            <- unique(subset(tmp, select=c(WHO, SAMPLING_CATEGORY)))
  dlu[, UPDATE_ID:= seq_len(nrow(dlu))]

  # mcmc
  mcmc.file  <-  paste0("sarws_mcmc",i,".rda")
  control  <-  list(seed=randomnumber, mcmc.n=nrow(dlu)*1e5, verbose=0, outfile=mcmc.file)
  source.attribution.mcmc(dobs, dprior_sarws, control)

  mcmc.file<-paste0("sar_mcmc",i,".rda")
  control<-list(seed=randomnumber, mcmc.n=nrow(dlu)*1e5, verbose=0, outfile=mcmc.file)
  source.attribution.mcmc(dobs, dprior_sar, control)
}
```

Mean absolute error and worst case error

After this, we calculated the mean absolute errors and worst case errors between the true transmission flows (`TRUE_PI`) and the corresponding mean estimates of the posterior distribution under **phyloflows** Bayesian multi-level model with adjustment for sampling heterogeneity.

```r
# true PI
TRUE_PI  <-  c(0.36,0.04,0.06,0.54)
```

```r
# record median of PI for each mcmc outputs
PI_M_SARWS_M <-  matrix(NA_real_,nrow = 100, ncol=4)
PI_M_SAR_M  <-  matrix(NA_real_,nrow = 100, ncol=4)

for (i in 1:100){
  load(paste0("sarws_mcmc",i,".rda"))
  # remove burnin
  burnin  <-  round(nrow(mc$pars$S)*0.2)
  tmp.pi  <-  1:nrow(mc$pars$PI)
  id.pi  <-  tmp.pi[tmp.pi>burnin]
  PI  <-  mc$pars$PI[id.pi,]
  # estimated PI
  PI_M_SARWS  <-  apply(PI, 2, mean)
  PI_M_SARWS_M[i,]  <-  PI_M_SARWS
}

# calculate the mean absolute error and worst case error
PI_TRUE_M  <-  t(replicate(100, TRUE_PI))
PI_ABS_ERROR_SARWS  <-  abs(PI_M_SARWS_M-PI_TRUE_M)
PI_MAE_SARWS  <-  apply(PI_ABS_ERROR_SARWS,1,mean)
PI_WCE_SARWS  <-  apply(PI_ABS_ERROR_SARWS,1,max)
PI_MAE_SARWS.df  <-  data.table(PI_MAE=PI_MAE_SARWS, REPLICATE=1:100,
                                SCENARIO=rep(1,100), N=rep(300,100), SAMP_DIFF=rep(0.15,100))
PI_WCE_SARWS.df  <-  data.table(PI_WCE=PI_WCE_SARWS, REPLICATE=1:100,
                                SCENARIO=rep(1,100), N=rep(300,100), SAMP_DIFF=rep(0.15,100))
```

Similarly, errors were calculated for the scenario where no differences appear in sampling.

```r
for (i in 1:100){
  load(paste0("sar_mcmc",i,".rda"))
  # remove burnin
  burnin  <-  round(nrow(mc$pars$S)*0.2)
  tmp.pi<-1:nrow(mc$pars$PI)
  id.pi<-tmp.pi[tmp.pi>burnin]
  PI<-mc$pars$PI[id.pi,]
  # estimated PI
  PI_M_SAR<-apply(PI, 2, median)
  PI_M_SAR_M[i,]<-PI_M_SAR
}

# calculate the mean absolute error and worst case error of estimation
PI_ABS_ERROR_SAR  <-  abs(PI_M_SAR_M-PI_TRUE_M)
PI_MAE_SAR  <-  apply(PI_ABS_ERROR_SAR,1,mean)
PI_WCE_SAR  <-  apply(PI_ABS_ERROR_SAR,1,max)

PI_MAE_SAR.df  <-  data.table(PI_MAE=PI_MAE_SAR, REPLICATE=1:100,
                                       SCENARIO=rep(1,100), N=rep(300,100), SAMP_DIFF=rep(0.15,100))
PI_WCE_SAR.df  <-  data.table(PI_WCE=PI_WCE_SAR, REPLICATE=1:100,
                             SCENARIO=rep(1,100), N=rep(300,100), SAMP_DIFF=rep(0.15,100))
```

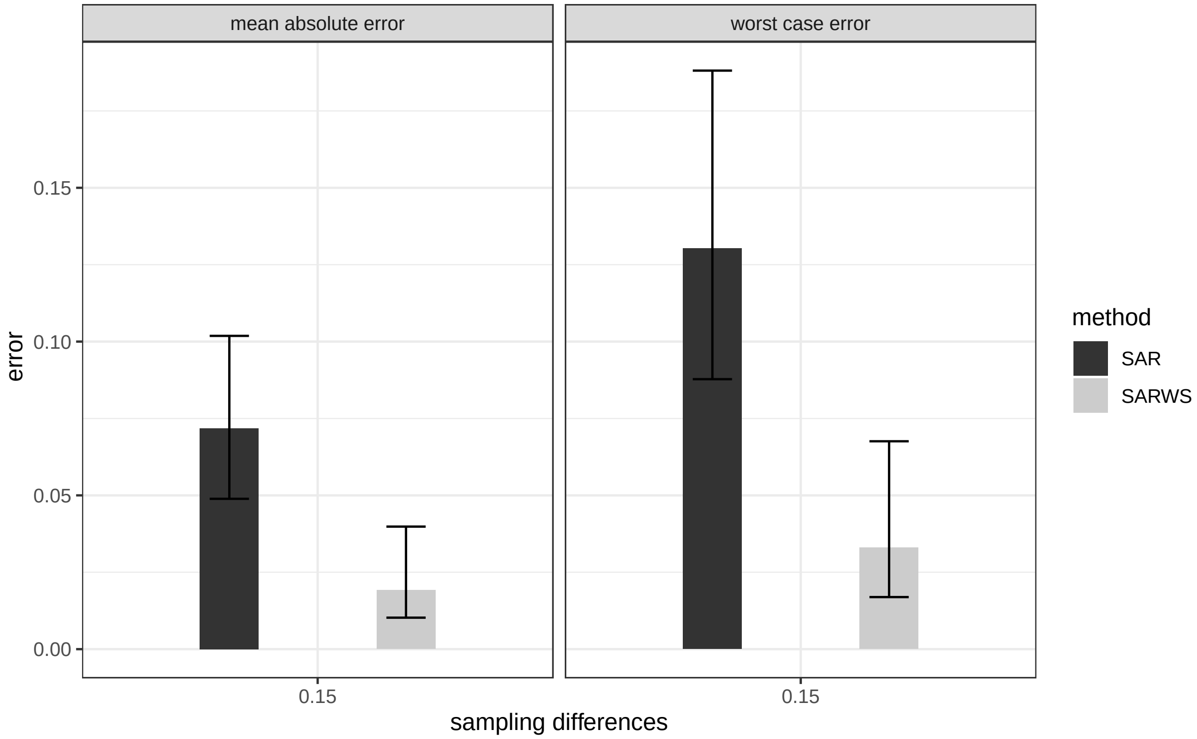
Gathering results together gives data *twoGroupFlows100_mcmcError*.

```r
# combine data tables
PI_MAE_SARWS.df[,METHOD:='SARWS']
```

```
PI_MAE_SAR.df[,METHOD:='SAR']
PI_MAE.df  <-  rbind(PI_MAE_SARWS.df,PI_MAE_SAR.df)

PI_WCE_SARWS.df[,METHOD:='SARWS']
PI_WCE_SAR.df[,METHOD:='SAR']
PI_WCE.df  <-  rbind(PI_WCE_SARWS.df,PI_WCE_SAR.df)

PI_MAE.df[,ERROR_TYPE:='MAE']
setnames(PI_MAE.df,'PI_MAE','ERROR')
PI_WCE.df[,ERROR_TYPE:='WCE']
setnames(PI_WCE.df,'PI_WCE','ERROR')
PI_ERROR.df <- rbind(PI_MAE.df,PI_WCE.df)
```

Then we load error data, make plots and compare errors with and without adjustment for sampling heterogeneity. It shows 5% reduction in mean absolute errors and 10% reduction in worst case errors on average by adjusting for sampling differences, when it exists (15%).

```
data(twoGroupFlows100_mcmcError)
de <-  twoGroupFlows100_mcmcError[, list( CL = quantile(ERROR,0.025),
                                          CU = quantile(ERROR,0.975),
                                          M = quantile(ERROR,0.5)),
                                     by = c('SAMP_DIFF','SCENARIO','N','METHOD','ERROR_TYPE')]

de$ERROR_TYPE <- factor(de$ERROR_TYPE,levels = c('MAE', 'WCE'), labels = c('mean absolute error','worst
ggplot(de, aes(x=as.factor(SAMP_DIFF), y=M, fill=METHOD)) +
   geom_bar(stat="identity",position=position_dodge(.9),width = 0.3)+
   geom_errorbar(aes(ymin=CL, ymax=CU),width=.2,position=position_dodge(.9))+
   scale_fill_grey(start = 0.2, end = 0.8)+
   theme_bw() +
   facet_grid(.~ERROR_TYPE)+
   labs(x=' sampling differences \n', y='\n error',fill='method')
```


mean absolute error
worst case error
error
0.00
0.05
0.10
0.15
0.15
0.15
sampling differences
method
SAR
SARWS