phyloflows: Aggregating MCMC output to a new set of variables

Xiaoyue Xi and Oliver Ratmann

2019-09-10

This vignette describes how to aggregate estimated transmission flows to those of other (broader) population groups. Please work through the vignette *phyloflows: Estimating transmission flows under heterogeneous sampling - a first example* before you go ahead here.

Getting started

We continue our "First_Example". The following code chunk contains all code needed, up to running **phyloflows** MCMC routine. The only change is that the number of iterations is now 50,000. The MCMC should take about 2 minutes to run.

```
require(data.table)
require(phyloflows)
data(twoGroupFlows1, package="phyloflows")
dobs <- twoGroupFlows1$dobs
dprior <- twoGroupFlows1$dprior
control <- list(seed=42, mcmc.n=5e4, verbose=0)
mc <- phyloflows:::source.attribution.mcmc(dobs, dprior, control)
```

Aggregating flows

Why would it be useful to aggregated the estimated transmission flows? Let us suppose that group "1" are the individuals aged 15-24 and group "2" are the individuals aged 25 or older in a population. The estimated flow vector

$$\pi = (\pi_{11}, \pi_{12}, \pi_{21}, \pi_{22})$$

describes the transsmission flow within and between the two age categories. But what is the overall contribution of transmissions from individuals aged 15-24, and the overall contribution of transmissions from individuals aged 25+? We want to estimate

$$\eta = (\eta_1, \eta_2)$$

where $\eta_1 = \pi_{11} + \pi_{12}$ and $\eta_2 = \pi_{21} + \pi_{22}$. There are many similar scenarios like that, and **phyloflows** has a little function to help you with that task. The syntax is as follows.

```
daggregateTo <- subset(dobs, select=c(TRM_CAT_PAIR_ID, TR_TRM_CATEGORY, REC_TRM_CATEGORY))
daggregateTo[, TR_TARGETCAT:= TR_TRM_CATEGORY]
daggregateTo[, REC_TARGETCAT:= 'Any']
set(daggregateTo, NULL, c('TR_TRM_CATEGORY','REC_TRM_CATEGORY'), NULL)
control <- list( burnin.p=0.05,
                 thin=NA_integer_,
                 regex_pars='PI')
mca <- phyloflows:::source.attribution.mcmc.aggregateToTarget(mc=mc,
        daggregateTo=daggregateTo,
        control=control)
#>
#> Using MCMC output specified as input...
```

```
#> Collecting parameters...
#> Removing burnin in set to  5 % of chain, total iterations= 625
#> Making aggregated MCMC output...
mca
#>         VARIABLE TR_TARGETCAT REC_TARGETCAT SAMPLE     VALUE
#>     1:       PI            1           Any      1 0.3590633
#>     2:       PI            2           Any      1 0.6409367
#>     3:       PI            1           Any      2 0.4151095
#>     4:       PI            2           Any      2 0.5848905
#>     5:       PI            1           Any      3 0.3511756
#>    ---
#> 23750:       PI            2           Any  11875 0.5775169
#> 23751:       PI            1           Any  11876 0.3961713
#> 23752:       PI            2           Any  11876 0.6038287
#> 23753:       PI            1           Any  11877 0.3923836
#> 23754:       PI            2           Any  11877 0.6076164
```

The output is a data.table that contains the aggregated transmission flows, and other aggregated variables depending on the value of `control[['regex_pars]]`. In our case, we removed a burnin-period of 5% of the MCMC chain, and did not thin the remaining iterations, yielding about 12,000 MCMC samples of the aggregated flows.

That's it for now. Use your usual R wizadry to process the output further, and have a look at the other vignettes.