phyloflows: Calculating sources, onward transmissions and flow ratios

Xiaoyue Xi and Oliver Ratmann

2019-09-10

This vignette describes how a number of key summary statistics of inferred transmission flows can be easily calculated with **phyloflows** `source.attribution.mcmc.getKeyQuantities` function. 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.

```r
require(data.table)
require(phyloflows)
data(twoGroupFlows1, package="phyloflows")
dobs <- twoGroupFlows1$dobs
dprior <- twoGroupFlows1$dprior
mcmc.file <- file.path(getwd(),'twoGroupFlows1_mcmc.RData')
control <- list(seed=42, mcmc.n=5e4, verbose=0)
mc <- phyloflows:::source.attribution.mcmc(dobs, dprior, control)
```

Sources of transmission for each group

OK, so now we have samples from posterior distribution of transmission flows within and between the two population groups,

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

One important summary statistic are the sources of transmissions into each recipient group, defined by

$$\eta = (\eta_{11}, \eta_{21}, \eta_{12}, \eta_{22})$$

where $\eta_{ij} = \pi_{ij}/\sum_s \pi_{sj}$.

Onward transmissions from each group

Another important summary statistic are the proportions of transmissions that originate from each group, defined by

$$\nu = (\nu_{11}, \nu_{21}, \nu_{12}, \nu_{22})$$

where $\nu_{ij} = \pi_{ij}/\sum_s \pi_{is}$.

Transmission flow ratios

Yet another important summary statistic are ratios of transmission flows, defined by

$$\rho_{ij} = \pi_{ij}/\pi_{ji}.$$

Calculating key quantities

phyloflow has a function to calculate the above summary statistics. The basic syntax is as follows:

```
#   specify list of user options
#
#   burnin.p: proportion of samples to discard as burn-in
#             (only needed when the burn-in was not already removed)
#
#   thin: keep every thin-nth iteration
#         (only needed when thinning was not already performed)
#
#   quantiles: quantiles of the marginal posterior distributions
#              that will be computed
#
#   flowratios: list of vectors of 3 elements. The 3 elements
#               specify the name of the flow ratio (first element),
#               the enumerator of the flow ratio (second element),
#               and the denominator of the flow ratio (third element)
control <- list(  burnin.p=0.05,
                  thin=NA_integer_,
                  quantiles= c('CL'=0.025,'IL'=0.25,'M'=0.5,'IU'=0.75,'CU'=0.975),
                  flowratios= list( c('1/2', '1 2', '2 1'), c('2/1', '2 1', '1 2'))
                  )
ans <- source.attribution.mcmc.getKeyQuantities(mc=mc,
        dobs=dobs, control=control)
#>
#> Removing burnin in set to  5 % of chain, total iterations= 312
#> Computing flows...
#> Computing WAIFM...
#> Computing sources...
#> Computing flow ratios...
ans
#>     TR_TARGETCAT REC_TARGETCAT         CL         CU         IL         IU          M
#>  1:            1            1 0.28456473 0.38958457 0.31700506 0.35360498 0.33477632    33.5%\n[28.5
#>  2:            1            2 0.02761753 0.07587423 0.03965832 0.05634378 0.04756175    4.8%\n[2.8%
#>  3:            2            1 0.04038266 0.09469313 0.05451156 0.07383531 0.06399857     6.4%\n[4%
#>  4:            2            2 0.49236956 0.60779904 0.53073874 0.57183554 0.55152806 55.2%\n[49.2%
#>  5:            1            1 0.80892729 0.92597288 0.85423743 0.89498994 0.87594336 87.6%\n[80.9%
#>  6:            1            2 0.07402712 0.19107271 0.10501006 0.14576257 0.12405664   12.4%\n[7.4%
#>  7:            2            1 0.06640809 0.15340581 0.08886043 0.11994677 0.10402981   10.4%\n[6.6%
#>  8:            2            2 0.84659419 0.93359191 0.88005323 0.91113957 0.89597019 89.6%\n[84.7%
#>  9:            1            1 0.76945781 0.89709465 0.81630088 0.86138611 0.83995630    84%\n[76.9%
#> 10:            2            1 0.10290535 0.23054219 0.13861389 0.18369912 0.16004370    16%\n[10.3%
#> 11:            1            2 0.04597194 0.12694009 0.06630491 0.09381147 0.07948609   7.9%\n[4.6%
#> 12:            2            2 0.87305991 0.95402806 0.90618853 0.93369509 0.92051391 92.1%\n[87.3%
#> 13:           NA           NA 0.36586230 1.46188507 0.58874012 0.94076841 0.74408828    0.74\n[0.37
#> 14:           NA           NA 0.68404830 2.73327034 1.06296087 1.69854232 1.34392656    1.34\n[0.68
#>                  LABEL2        STAT FLOWRATIO_CAT
#>  1:    33.5% (28.5%-39%)     flows          <NA>
#>  2:    4.8% (2.8%-7.6%)     flows          <NA>
#>  3:      6.4% (4%-9.5%)     flows          <NA>
#>  4: 55.2% (49.2%-60.8%)     flows          <NA>
#>  5: 87.6% (80.9%-92.6%)     waifm          <NA>
#>  6:   12.4% (7.4%-19.1%)     waifm          <NA>
```

```
#>  7:   10.4% (6.6%-15.3%)       waifm           <NA>
#>  8: 89.6% (84.7%-93.4%)        waifm           <NA>
#>  9:   84% (76.9%-89.7%)       sources          <NA>
#> 10:   16% (10.3%-23.1%)       sources          <NA>
#> 11:   7.9% (4.6%-12.7%)       sources          <NA>
#> 12: 92.1% (87.3%-95.4%)       sources          <NA>
#> 13:     0.74 (0.37-1.46)  flow_ratio           1/2
#> 14:     1.34 (0.68-2.73)  flow_ratio           2/1
```

Note

Note it is also possible to specify a file name to MCMC output or aggregated MCMC output, and it is also possible to input aggregated MCMC output. Please look up the package help for further instructions.

```
?phyloflows:::source.attribution.mcmc.getKeyQuantities
```