Simple and efficient R toolkit for ABC inference in population genetics

Martin Petr [1,2]
Fernando Racimo [1,2]

Approximate Bayesian Computation (ABC) is a powerful inference method for evolutionary biology. However, numerous technical challenges and significant amount of coding required often hinder its application to new modeling problems. We present a new R package, *demografr*, designed for building simple, efficient, and reproducible ABC pipelines entirely in R, using tree sequences for fast computation and the *abc* package as an inference engine. Additionally, *demografr* provides individual components for building other kinds of simulation-based inference workflows. Below, we demonstrate the features of *demografr* on a simple (but complete) ABC pipeline.

1. demographic model

Demographic models for ABC inference are defined using the *slendr* R interface [*bodkan.net/slendr*]. Any function producing a *slendr* model can serve as model for a *demografr* inference, with arguments of the function representing model parameters. However, users are not limited by *slendr*: any SLiM or *msprime* script can be used as long as it creates a *tskit* tree sequence.

2. prior distributions

Priors are defined in an R list, using an intuitive domain-specific syntax of `<parameter> ~ <sampling expression>`. In addition to built-in R sampling functions like `runif`, users can define priors using custom-defined distributions.

3. observed summary statistics

Observed summary statistics are inputted as data frames or numeric vectors, and are collected in a named R list (one element of the list for each statistic).

4. tree-sequence summary statistics

The output of each simulation of the model (**1.**) will be a tree sequence. Therefore, simulated summary statistics can be computed very efficiently thanks to *tskit* [Kelleher *et al.*, *PLOS Comp Biol* (2018)], via *slendr*'s intuitive *tskit* R interface: allele frequency spectrum, f_3, f_4, nucleotide diversity, Identity-by-Descent sharing, and more are available. Summary functions are collected in a named R list in a format matching the list of the observed statistics (**3.**). File format conversion for computation in external software is therefore not needed.

5. simulations

Simulations are implicitly parallelized across CPU cores and/or remote servers via the R package *future* [Bengtsson, *The R Journal* (2021)]. During this process, user-defined summary statistic functions (**4.**) are internally applied to the simulated tree sequence produced by each simulation replicate.

6. ABC inference

`perform_abc()` is a wrapper around the inference machinery of the *abc* package [Csilléry *et al.*, *Methods in Eco Evol* (2012)], acting as a bridge feeding simulation results from *demografr* (**5.**) to the *abc* package in the correct format. Support for other methods and packages as inference "engines" is in progress.

7. analysis

The result of ABC inference (**6.**) is compatible with all the validation and diagnostic functions of the *abc* package, for which *demografr* provides convenient wrappers. Additionally, *demografr* provides its own functions for inspection and visualization of posterior distributions.

1.
```
#------------------------------------------------------------------
model <- function(Ne_A, Ne_B, Ne_C, Ne_D, T_AB, T_BC, T_CD, gf_BC) {
  popA <- population("popA", time = 1,    N = Ne_A)
  popB <- population("popB", time = T_AB, N = Ne_B, parent = popA)
  popC <- population("popC", time = T_BC, N = Ne_C, parent = popB)
  popD <- population("popD", time = T_CD, N = Ne_D, parent = popC)

  gf <- gene_flow(from = popB, to = popC,
                  start = 9000, end = 9301, rate = gf_BC)

  compile_model(
    populations = list(popA, popB, popC, popD), gene_flow = gf,
    generation_time = 1, simulation_length = 10000
  )
  return(model)
}
```

2.
```
#------------------------------------------------------------------
priors <- list(
  Ne_A ~ runif(100, 10000), Ne_B ~ runif(100, 10000),
  Ne_C ~ runif(100, 10000), Ne_D ~ runif(100, 10000),
  ... # split time and gene-flow priors omitted for brevity
)
```

3.
```
#------------------------------------------------------------------
observed <- list(diversity  = observed_diversity,
                 divergence = observed_divergence,
                 f4         = observed_f4)
```

4.
```
#------------------------------------------------------------------
compute_diversity <- function(ts) {
  samples <- extract_names(ts, split = "pop")
  ts_diversity(ts, sample_sets = samples)
}
compute_divergence <- function(ts) { ... } # omitted for brevity
compute_f4 <- function(ts) { ... }         # omitted for brevity

functions <- list(diversity  = compute_diversity,
                  divergence = compute_divergence,
                  f4         = compute_f4)
```

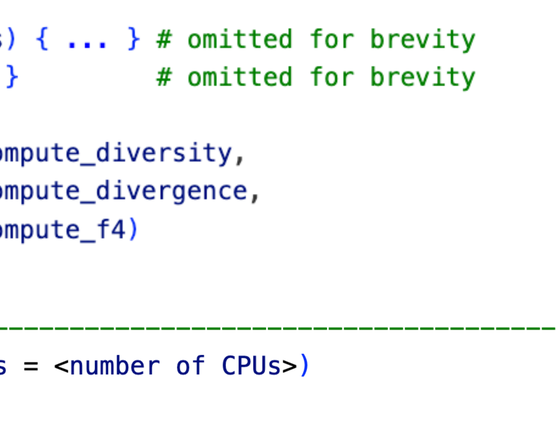
5.
```
#------------------------------------------------------------------
future::plan("multicore", workers = <number of CPUs>)

data <- simulate_abc(
  model, priors, functions, observed, iterations = 100000,
  sequence_length = 10e6, recombination_rate = 1e-8
)
```

6.
```
#------------------------------------------------------------------
abc <- perform_abc(data, engine = "abc",
                   tol = 0.05, method = "neuralnet")
```

7.
```
#------------------------------------------------------------------
extract_summary(abc, "Ne_A"); plot_posterior(abc, "T")
```

```
                                Ne_A
Min.:                     -3.010207
Weighted 2.5 % Perc.:    474.550342
Weighted Median:         1782.484402
Weighted Mean:           1840.627399
Weighted Mode:           1454.772471
Weighted 97.5 % Perc.:   3424.879882
Max.:                    4030.283888
```



Visit bodkan.net/demografr for documentation and extensive tutorials!

[1] Lundbeck Foundation GeoGenetics Centre, Globe Institute, University of Copenhagen | [2] Section for Molecular Ecology and Evolution, Globe Institute, University of Copenhagen

This project is supported by Lundbeck Foundation, Novo Nordisk Foundation and Villum Foundation

Contact: mp@bodkan.net