

rejection_s2.R

anubhav

2020-09-25

```
# implements rejection algorithm from Handbook of Approximate Bayesian Computation
```

```
rejection_s2 <- function(fw_data, model, model_core_par,
                        model_prior_par, input_parameters, dist_ss,
                        prior_dist, sir_data_main,
                        weight_type, species_ind, real_pred_mat,
                        dist_TSS, n_sir = n_sir)
{
  M <- fw_data$species.sizes      #body size
  tol <- input_parameters$tol
  N <- input_parameters$N
  n_cores <- input_parameters$n_cores
  e <- model_core_par$e
  n_par <- model_core_par$n
  ni <- model_core_par$ni
  r.a <- model_core_par$r.a

  registerDoParallel(cores=n_cores)

  ss_real <- sir_data_main

  post_dists <- data.frame(a=double(), ai=double(), aj=double(), r.b=double())
  n <- N/n_cores

  res_pcores <- foreach(i = 1:n_cores, .combine = rbind) %dopar%
  {
    set.seed(i*n_sir)

    acc_ss <- numeric(n)
    dist <- numeric(n)
    total_sim <- 0

    count <- 1
    while(count <= n)
    {

      #print(ss_real)

      local_count <- 0
      while(local_count == 0){
        ## Generating parameters from sampling

```

```

local_par <- prior_dist(par=model_prior_par, no=1)

## Generating summary stat from likelihood
ss_sim <- model(opt=local_par, x=model_core_par)
num_gen <- runif(1, 0, 1)
K <- weights(0, tol, weight_type)
ss_sim_tl <- trophic_position(web = ss_sim, title = "sim_foodweb")

if(sum(is.na(ss_sim_tl)) == 0){
  dist_temp <- dist_ss(ss_sim = ss_sim_tl, ss_real = ss_real, sp_ind = species_ind)
  #print(paste("dist = ", dist_temp))
  pbly <- weights(dist_temp, tol, weight_type)/K
  #print(paste("pbly = ", pbly))
  #print(paste("count = ", count))
  # if (num_gen<pbly)
  # {
  acc_ss[count] <- dist_TSS(ss_sim = ss_sim, ss_real = real_pred_mat)
  dist[count] <- dist_temp
  post_dists <- rbind(post_dists, local_par)
  count <- count + 1
  local_count <- local_count + 1
  }
  else(print(3))

  # }
}
total_sim <- total_sim + 1

}

list(post_dists_pcore = post_dists, acc_ss_pcore = acc_ss, dist_pcore = dist, total_sim_pcore = t

}

post_dists <- res_pcores[[1]]
acc_ss <- res_pcores[[n_cores+1]]
dist <- res_pcores[[2*n_cores+1]]
total_sim <- res_pcores[[3*n_cores+1]]

if(n_cores > 1){
  for (i in 1:(n_cores-1)){
    post_dists <- rbind(post_dists, res_pcores[[i+1]])
    acc_ss <- c(acc_ss, res_pcores[[n_cores+i+1]])
    dist <- c(dist, res_pcores[[2*n_cores+i+1]])
    total_sim <- total_sim + res_pcores[[3*n_cores+i+1]]
  }
}

list(acc_ss = acc_ss, dist = dist, post_dists = post_dists, total_sim = total_sim)
}

```