

# Compare approaches to ‘pre-simulate’

While thinking about how to speed up our simulations, I inevitably returned to the basic approach that *splatche* is supposed to take. I asked myself how hard it would be to do the “pre-simulation” in forward time and then use that output to parameterize a *fastsimcoal* model.

I tried two separate approaches. The first is more demographically realistic and runs in *rmetasim*. Basically, I set up a refugium (or multiples) and run the simulation with a particular dispersal kernel among populations. Each colonization event is recorded (both the source and sink as well as the time that it occurred). The colonization process works pretty much like you’d expect, very few populations colonized at first, but as some get colonized and can serve as sources, there is a burst of colonization followed by a tail that occurs when a few single uncolonized sites hang on for a while.

To speed this traditional demographic approach up, I run it with small population sizes. It runs acceptable fast for 100 populations, but takes around 20 sec for 196 populations and much slower for bigger numbers. Dispersal characteristics can change this dramatically, of course.

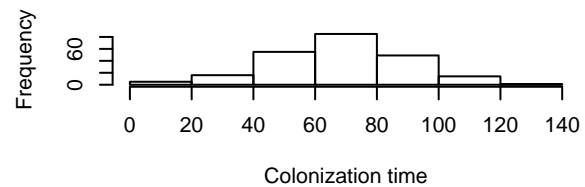
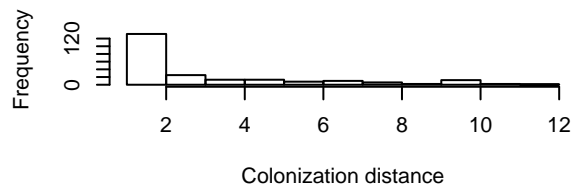
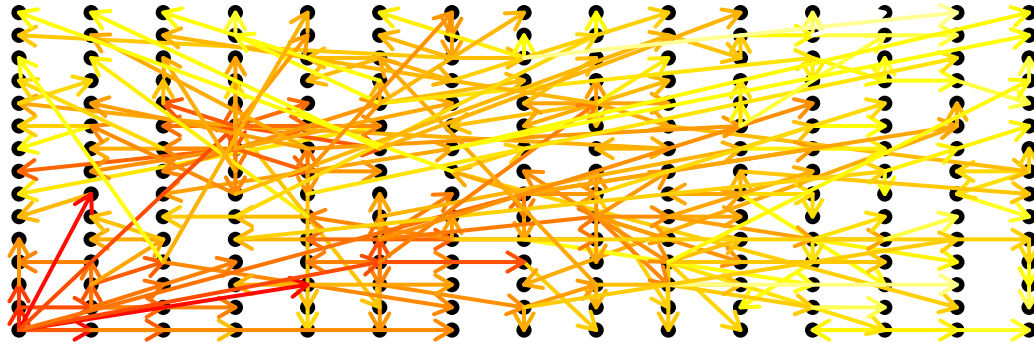
Here is an example with 225 populations: The arrows correspond to colonization events. I consider colonization to have occurred when a single individual reaches a site. The redder colors are early colonizations, the yellow ones are later (the lighter the yellow the later). The refugium is in the lower left corner.

```
library(rmetasim)
source("make-landscape.R")
source("plothist.R")
source("getpophist.R")

l <- recolonizeLandscape(dens.scale=0.05,
                        h=225,
                        refs=c(1),
                        sizeref=c(25),
                        mix=0.01,
                        longmean=3,
                        shortscale=0.35
                        )
system.time({pops1 <- getpophist(l)})
```

```
##      user  system elapsed
## 27.336   0.080   27.397
```

```
plothist(pops1)
```



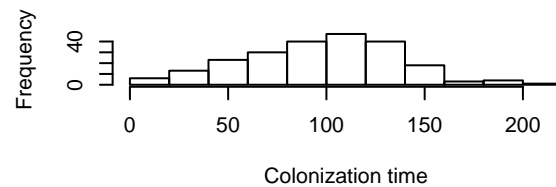
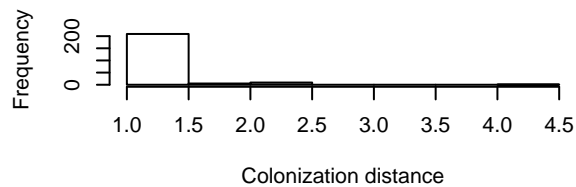
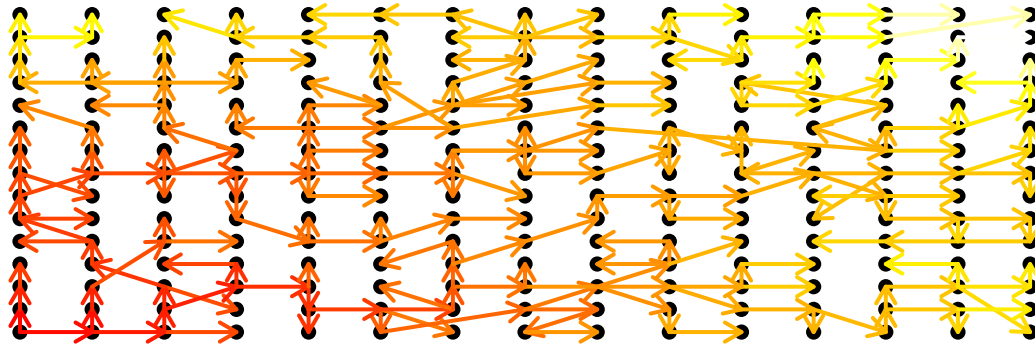
The other approach I tried is to run the demography “backwards” in each generation. Basically, each time click, I give each site a single chance to be colonized. I choose a direction and distance from the dispersal kernel (same pdf as above). If that direction and distance falls in a grid cell associated with a site that *has already been colonized* then I consider that a colonization event. If not then the site remains uncolonized until the next time click. So this approach is based on the perspective of the colonized site.

This approach is really fast. Here is the same simulation conducted above using this approach

```
source("colonization-model.R")
system.time({
  pops2 <- run.sim(h=225,
    extent=c(x0=0,y0=0,x1=15,y1=15),
    shortscale=0.35,
    longmean=3,
    mix=0.01)
})
```

```
##    user  system elapsed
##   1.216   0.012   1.226
```

```
plothist(pops2)
```



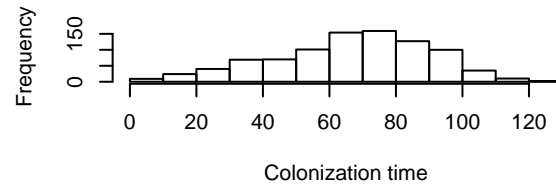
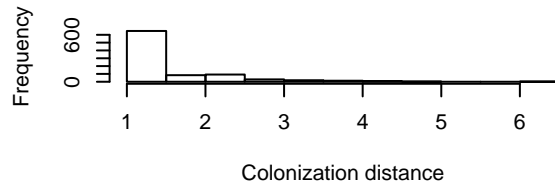
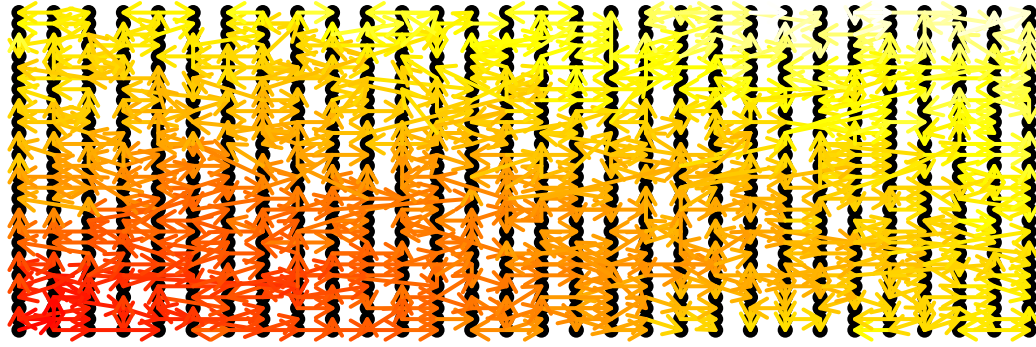
You can see in the second model that there colonization distances are smaller, but the colonization times are similar than the first. *It's really fast, though* and makes ABC more palatable.

In fact it is so fast that enormous landscapes are possible in reasonable time:

```
system.time({
  pops3 <- run.sim(h=900,
    extent=c(x0=0,y0=0,x1=15,y1=15),
    shortscale=0.35,
    longmean=3,
    mix=0.01)
})
```

```
##    user  system elapsed
##  4.608   0.016   4.624
```

```
plothist(pops3)
```



Anyway, it is relatively straightforward to convert these colonization histories to simcoal histories. Using the second approach it is conceivable that we can run into simcoal limitations on number of demes (I know it can run with 100). I'm thinking this is less effort than hacking Katies code or messing with an unreleased splatche.

There are many types of 'unrealism' here. In the first approach, colonization is premature. Often there are individuals that reach a site and then die off before population establishment (in the sim as well in real life). In the second, colonization does not actually work the way I am specifying, but I think both may be a good enough approximation for simcoal histories.

Ideas to contemplate, things to do: - How to represent cpDNA inheritance and dispersal versus nuclear using coalescent simulation - Figure out limitations of fastsimcoal with respect to deme number - improve simcoal parser to be more flexible with marker types