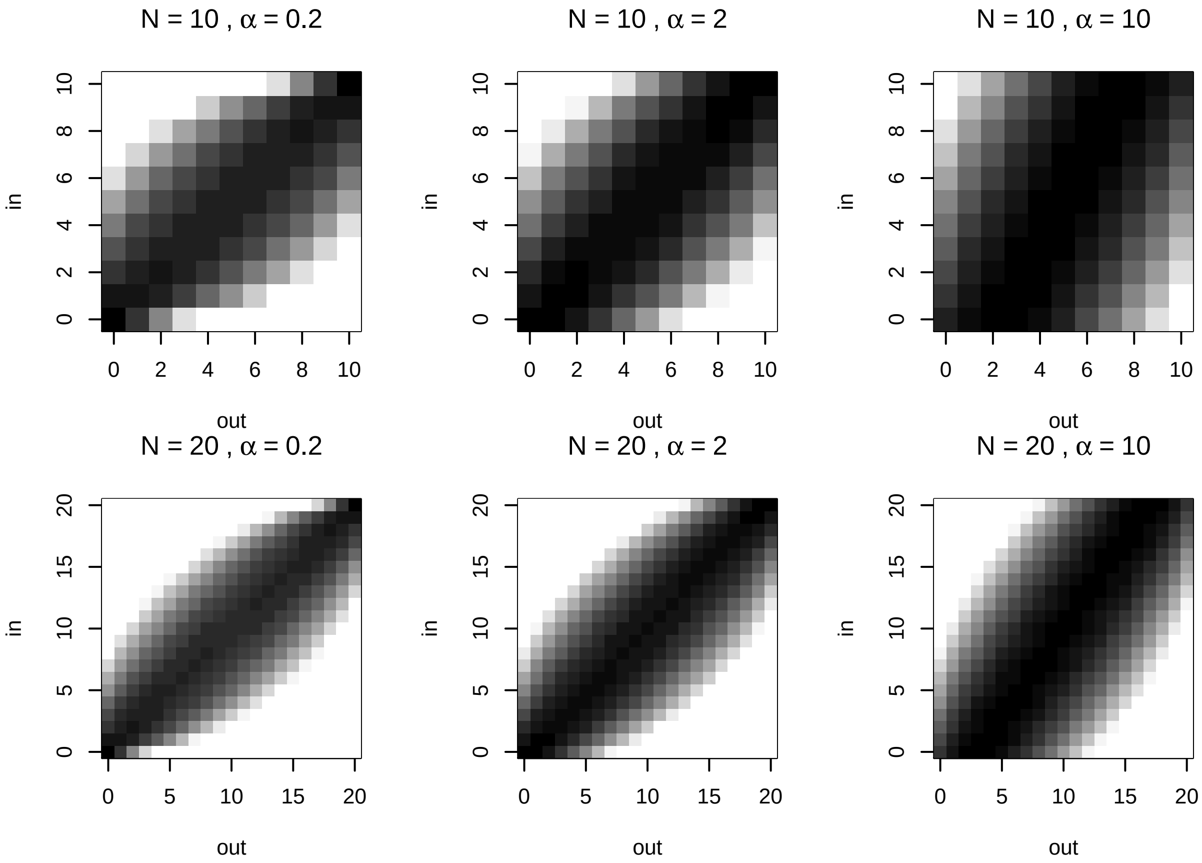
Language change in Bayesian learners does not exhibit s-shaped curves

Kevin Stadler

Thu Aug 13 09:54:52 2015

Replication of Reali and Griffiths (2010)

```
# create the (N+1)x(N+1) Markov chain transition matrix for the given N and alpha
bilm.transition.matrix <- function(N, alpha)
  t(sapply(0:N, function(n)dbinom(0:N, N, (n + alpha / 2) / (N + alpha))))
```



```
# the equivalent Wright-Fisher model transition matrix with mutation rate u
wright.fisher.transition.matrix <- function(N, u)
  t(sapply(0:N, function(x)dbinom(0:N, N, ((1-u)*x + u*(N-x)) / N)))

# calculate the Wright-Fisher model mutation rate equivalent to the BILM's N and
# alpha. the bracketing in the paper itself is garbled and there's a '/' missing
# somewhere, the correct version of the transformation can be found on the top
# of page 5 of the supplementary material
N.alpha.to.u <- function(N, alpha)
```

```
  alpha / (2 * (alpha + N))

# some equivalences between alpha and u for N=10
rbind(alpha=alphas, u=N.alpha.to.u(10, alphas))
```

```
##              [,1]       [,2]  [,3]
## alpha 0.200000000 2.00000000 10.00
## u     0.009803922 0.08333333  0.25
```

```
# the two ways of calculating the matrices are indeed identical (except for rounding errors)
all(bilm.transition.matrix(6, 2) == wright.fisher.transition.matrix(6, N.alpha.to.u(6, 2)))
```

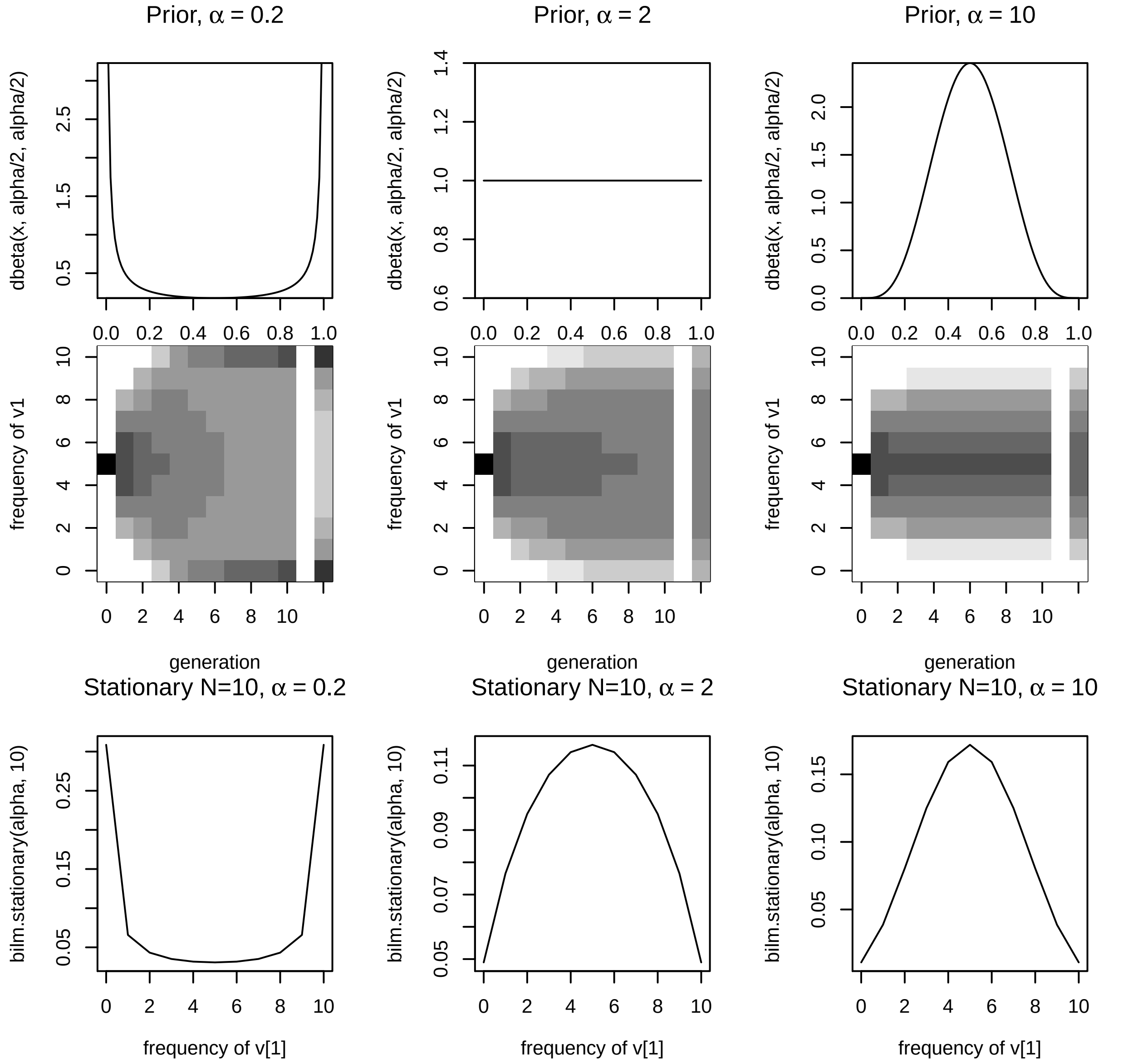
```
## [1] TRUE
```

Going dynamic

```
# compute the development of the Markov chain probability distribution for the
# specified number of iterations. initstate is a vector of length N and
# transitionmatrix an NxN matrix (with rows summing to 1)
markov.chain <- function(transitionmatrix, niterations, initstate=c(1, rep(0, ncol(transitionmatrix)-1)
  out <- matrix(nrow=niterations+1, ncol=length(initstate))
  out[1,] <- initstate
  for (i in 2:(niterations+1)) {
    out[i,] <- t(transitionmatrix) %*% out[i-1,]
  }
  return(out)
}

# full symmetric beta-binomial distribution
beta.stationary <- function(alpha, N)
  suppressWarnings(VGAM::dbetabinom.ab(0:N, N, alpha, alpha))

# stationary distribution of the BILM. bracketing in the paper is again implicit,
# the unambiguous version is on page 5 (bottom) of the supplementary material
bilm.stationary <- function(alpha, N)
  beta.stationary(alpha / (1 + alpha/N), N)

# stationary distribution of the Wright-Fisher model with N genes and mutation
# rate u. as u -> 0 the Beta distribution becomes more strongly bimodal (all
# chains stay in one of the absorbing states)
wright.fisher.stationary <- function(u, N)
  beta.stationary(2*N*u, N)

append.stationary <- function(data, alpha)
  rbind(data, rep(0, ncol(data)), bilm.stationary(alpha, ncol(data)-1))
```

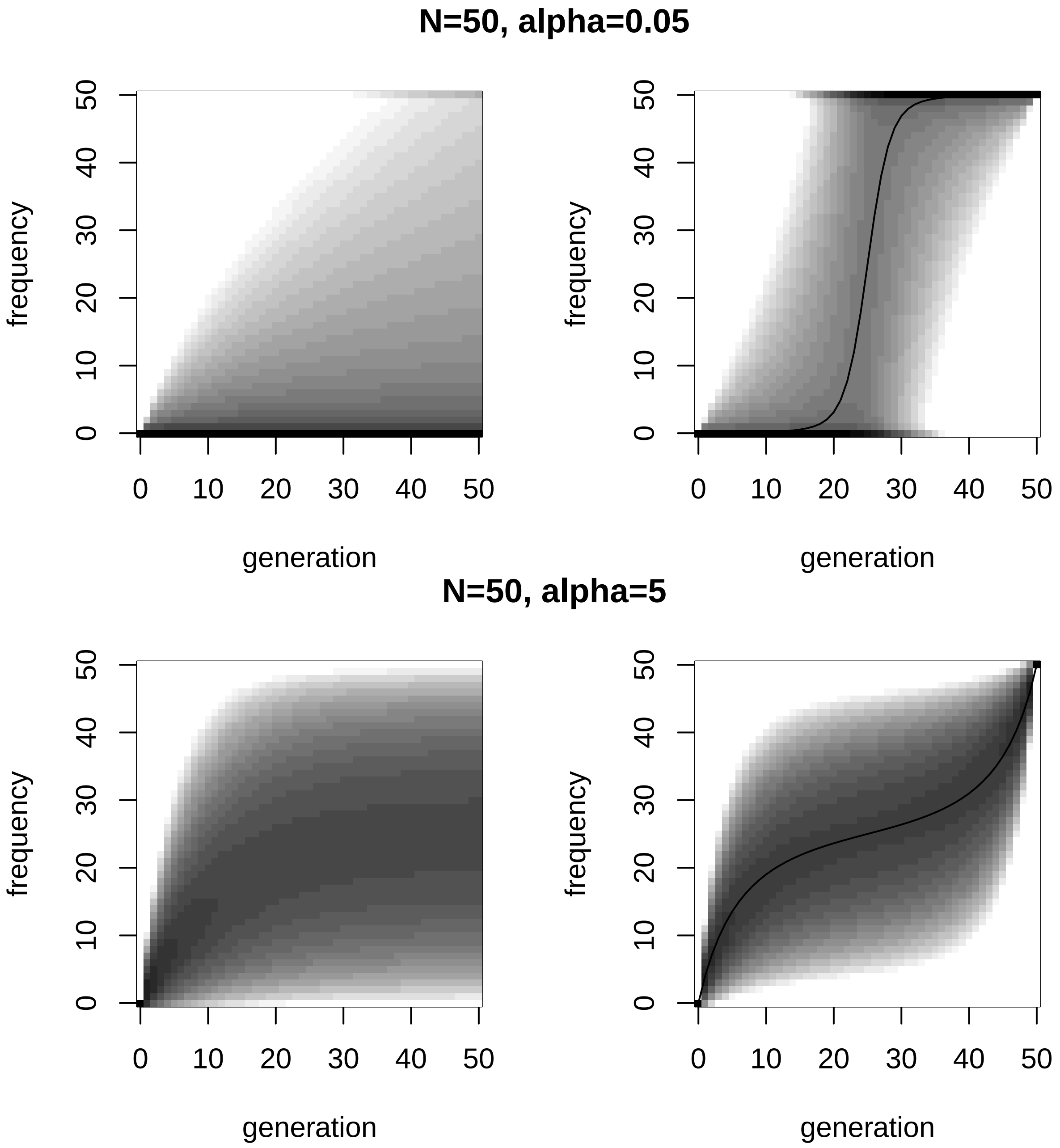
Figure 1

Row 1: prior distributions for three different values of α. Row 2: Overview of the Markov chain developments for $N = 10$ from initial state 5/10. The respective stationary distribution is also shown in the last column of the heatmap. Row 3: stationary distribution for $N = 10$.



S-shaped curves in language change (Figure 2)

Probability distribution of Markov chains of length N where the population starts in one homogeneous state and finishes in the other. Left figures: conditioning on the initial state only, right figures: conditioning on both the initial and final fixation states.



N=50, alpha=0.05

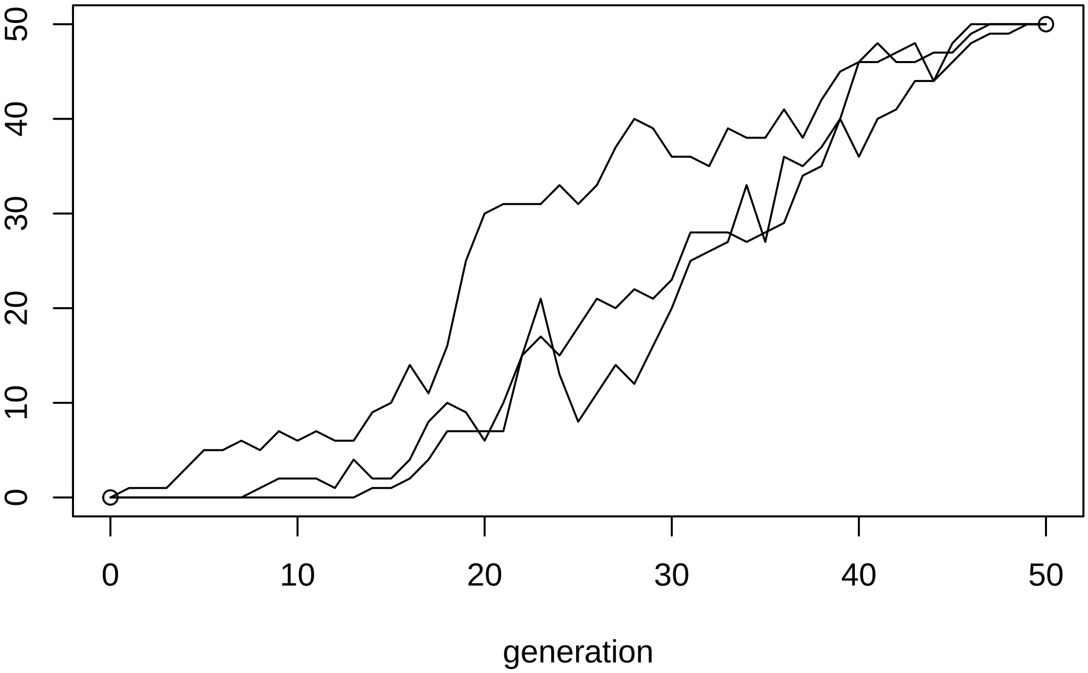
N=50, alpha=5

Critique

The trajectory for regularising $\alpha = 0.05$ shown above is the *average* of all transitions produced by the model, but this is not necessarily representative of a *typical* trajectory. Before taking a closer look at typical trajectories, there are a number of concrete predictions made by the model which are not fleshed out:

- Reali & Griffiths make the valid point that linguists only report those trajectories where a change actually took place, so a lot of the low-level variation walks generated by their model would simply never be documented. But because the transition matrix under neutral evolution is completely symmetric, a trajectory reaching the 50% mark is just as likely to go to fixation as it is likely to revert back to the initial state. This would predict that for any completed language change we should find another change that went to the halfway mark (in an individual? in a population?) before being interrupted, which does not seem to be the case.
- The probability that a chain that was initialised at 0/50 is also at 0/50 after 50 generations is 0.838508, while the probability that it has gone all the way to 50/50 is 6.8996171×10^{-4}. That transitions are rare is fine, the question is what the actual trajectories of these rare transitions look like. Let's generate some chains that fulfill the start and end condition, i.e. let's do some actual iterated learning!

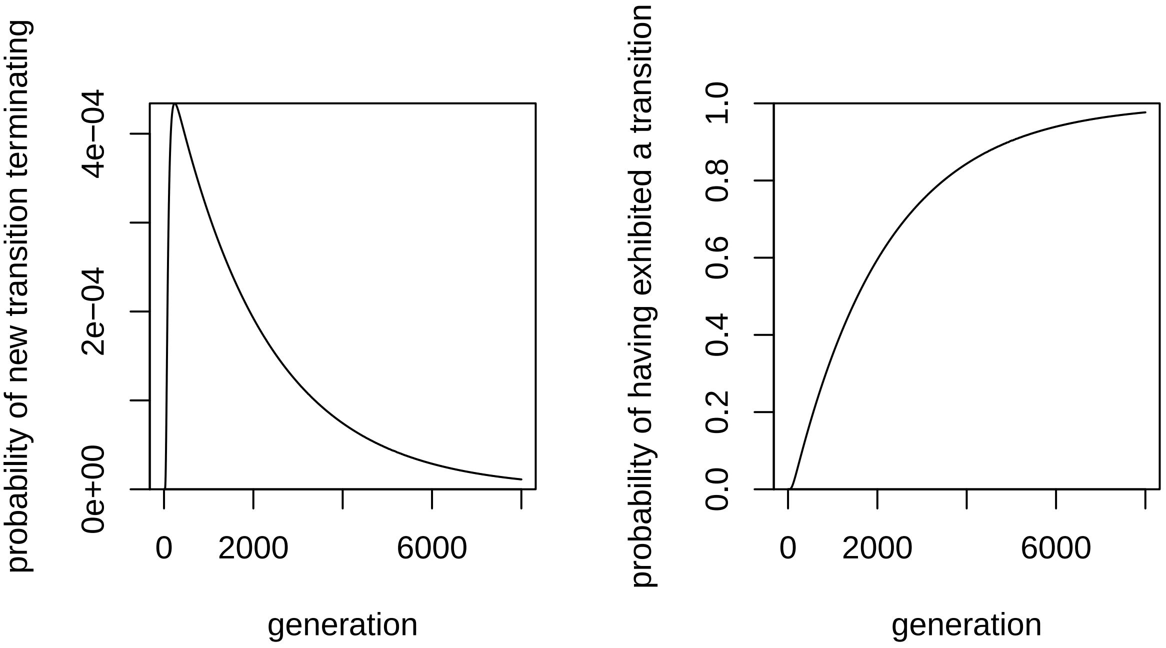
```
## Probability of being in state N/N after 50 generations is 0.0006899617
## This means one chain will be found roughly every 1449 attempts
## Found chain 1 after 1827 attempts
## Found chain 2 after 2832 attempts
## Found chain 3 after 428 attempts
```



Expected number of generations for a transition to complete

The trajectories above don't actually look too bad, but is the condition that they have terminated after 50 generations well-motivated? Even a single-generation jump from from 0/50 to 50/50 has a non-zero (if extremely small) probability and would arguably not be s-shaped. What is the likelihood of a transition completing in a given number of generations?

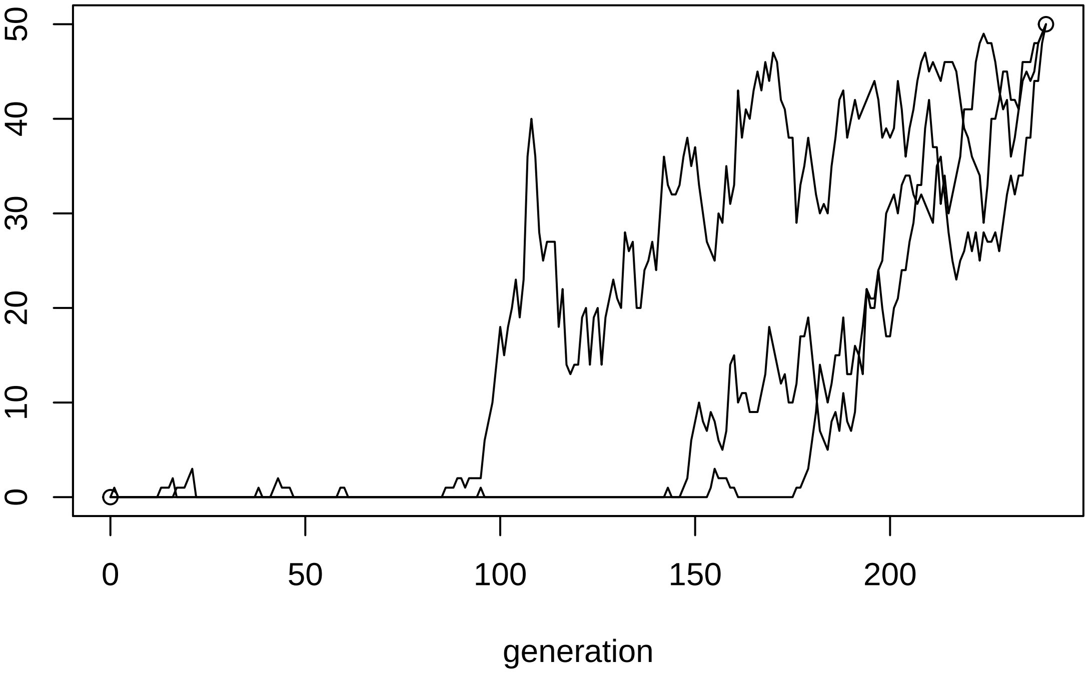
```
onetimetransitions <- bilm.transition.matrix(50, 0.05)
# drop chain from the pool once it reaches 50/50
onetimetransitions[51,] <- 0
```



For parameters $\alpha = .05$ and $N = 50$ a transition is most likely to have finished at the distribution's mode which is 240 generations. Let's generate some of these more 'typical' trajectories, limiting ourselves only to those fixating *exactly* after the specific number of generations (i.e. we exclude ones that reach the 50/50 state early and stay there).

```
plotchains(generate.transitioning.chains(3, which.max(data[,51]), 50, 0.05, TRUE))
```

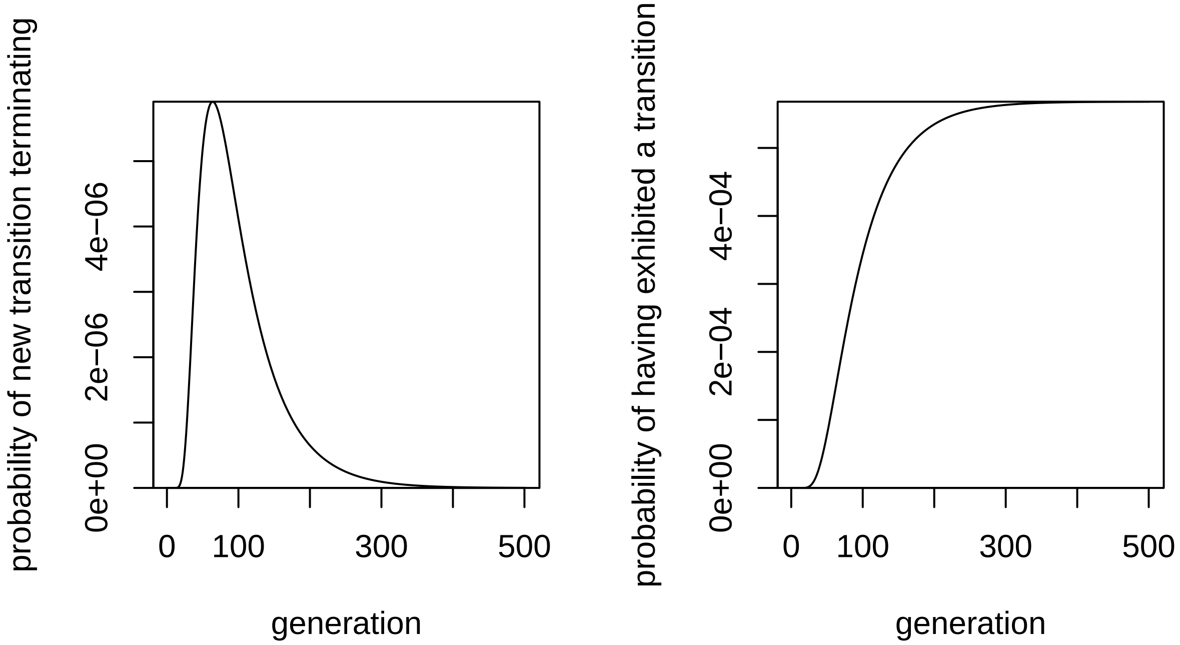
```
## Found chain 1 after 1411 attempts
## Found chain 2 after 3131 attempts
## Found chain 3 after 235 attempts
```



Average trajectory of transitions that have the exact same duration

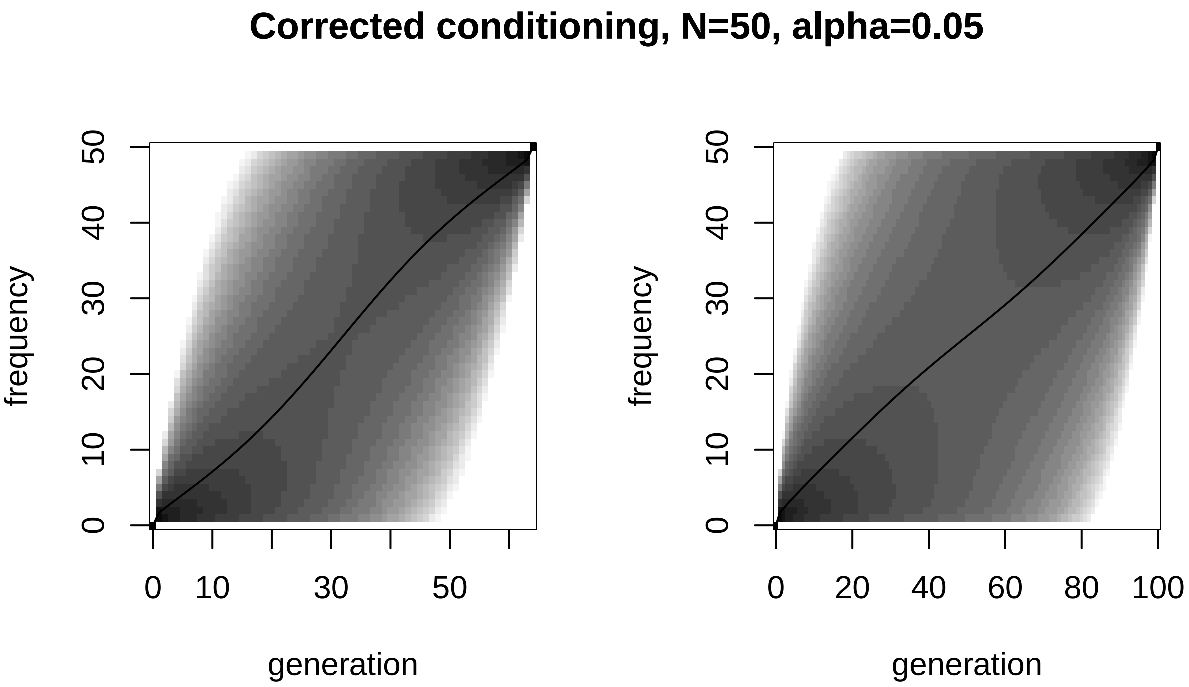
As can be seen above the number of generations until a new variant has fixated isn't actually representative of the *duration* of a transition, since transitions might remain at 0/50 for some time before picking up, or also return back to 0 before picking up again. If we are interested in the length of the actual transition (i.e. starting to count at the first move away from 0/50) the distribution of transition durations actually looks like this:

```
# distribution of states where the diffusion of the incoming variant started in generation 1
transitionjustpickedup <- onetimetransitions[1,]
transitionjustpickedup[c(1,51)] <- 0

# also drop chain if it reverts back to the initial 0/50 state
onetimetransitions[1,] <- 0
data <- markov.chain(onetimetransitions, 500, transitionjustpickedup)
```



The mode of this distribution is much lower at 64 generations (with the mean at ~100). The following code plots the temporal development and average trajectory of the Markov chains which are conditioned on actuating in the very first generation and fixating at the maximum number of generations (and no earlier) without ever returning to the initial state. It produces average trajectory plots for two different durations, the left one for the *mode* of the distribution of transition durations, the right one for the (higher) *mean* duration.

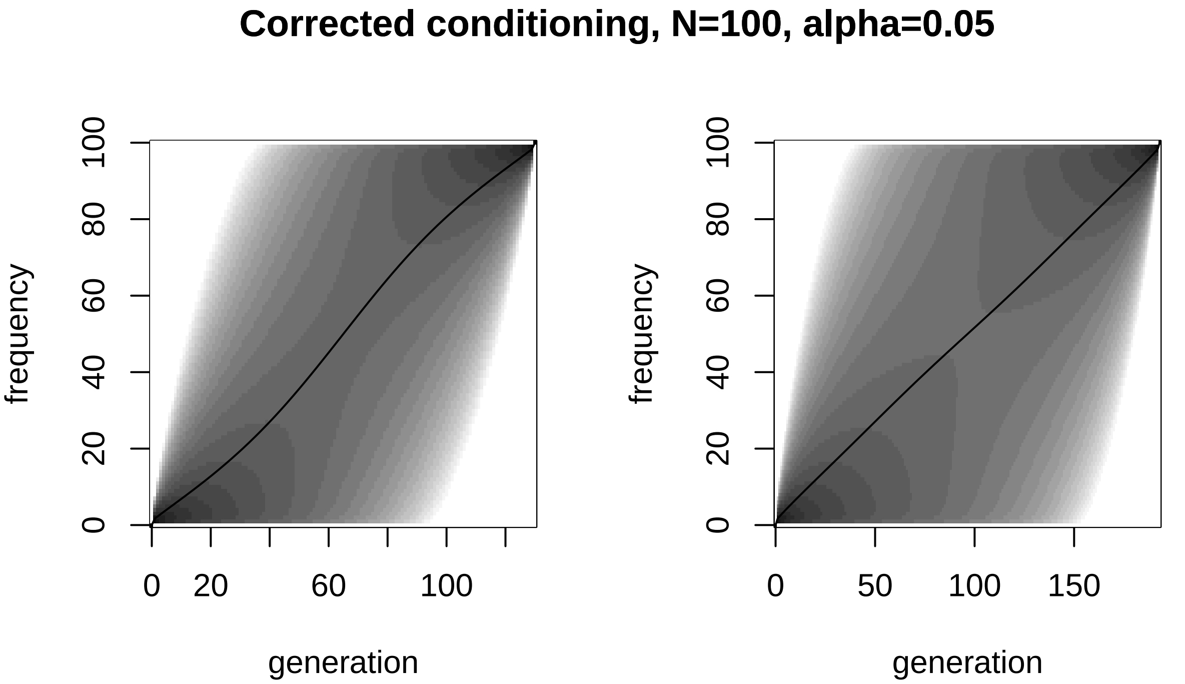
```
plot.doubleconditioned.strict <- function(N, alpha) {
  onetimetransitions <- bilm.transition.matrix(N, alpha)
  # drop chain from the pool once it reaches N/N
  onetimetransitions[N+1,] <- 0
  # disallow chains from reverting to 0/N
  onetimetransitions[,1] <- 0
  # determine generation 2 start state after the initial pickup
  transitionjustpickedup <- onetimetransitions[1,]
  transitionjustpickedup[1] <- 0
  data <- markov.chain(onetimetransitions, 500, transitionjustpickedup)
  # also disallow chains from reaching 0/N or N/N
  onetimetransitions[,N+1] <- 0
  par(pty="s", mfrow=c(1,2))#, mgp=c(2,1,0))
  for (generations in c(which.max(data[,N+1]), round(weighted.mean(0:500, data[,N+1])))) {
    data <- markov.chain(onetimetransitions, generations, transitionjustpickedup)
    doubleconditioned <- data * apply(data,2,rev)[,dim(data)[2]:1]
    doubleconditioned[1,1] <- 1
    doubleconditioned[generations+1,N+1] <- 1
    logheatmap(doubleconditioned/rowSums(doubleconditioned))
    lines(0:(nrow(data)-1), chain.mean(doubleconditioned))
#    lines(0:(nrow(data)-1), deviation(doubleconditioned), lty=2)
    title(paste("Corrected conditioning, N=", N, ", alpha=", alpha, sep=""), outer=T, line=-3)
  }
}

plot.doubleconditioned.strict(50, 0.05)
```

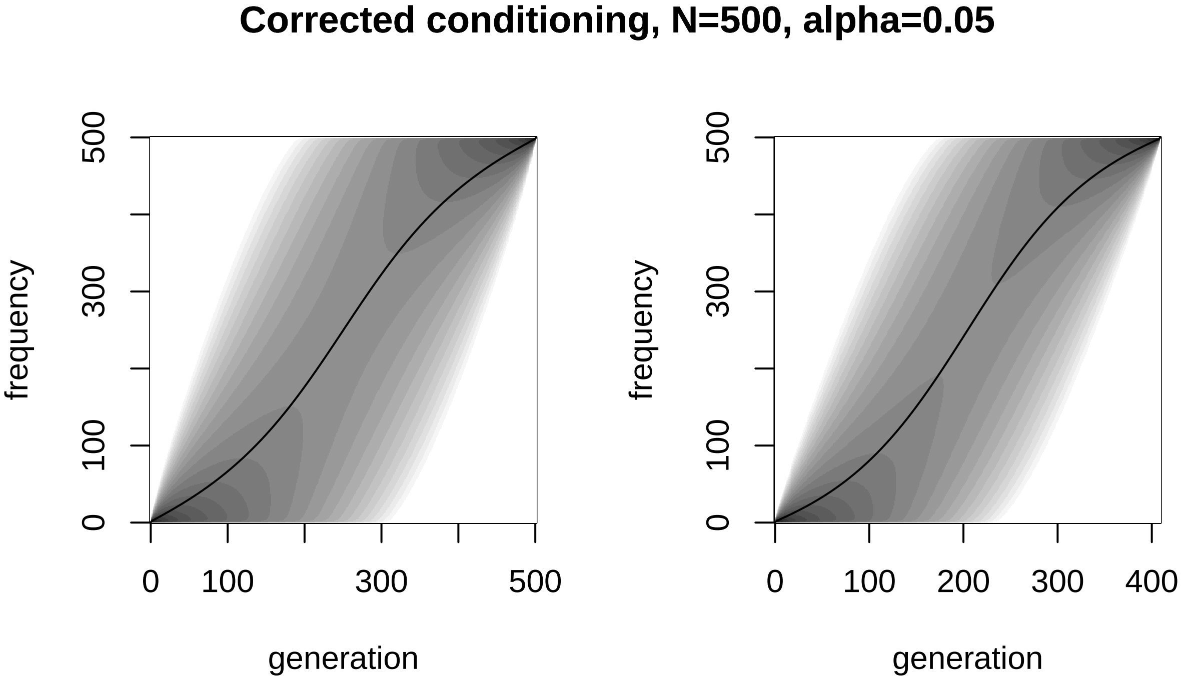


Even the average of these more strictly conditioned trajectories doesn't look particularly s-shaped anymore. Moreover, while shorter transitions (like ones of the length of the mode, left figure) still have the fastest rate of change at the mid-point, *longer* transitions (like ones of the mean length for a transition using the given parameters) are actually more like an S bent in the 'wrong' direction. (Individual trajectories fulfilling this stricter conditioning of exact start and ending points are exceedingly rare and consequently expensive to generate randomly, which is therefore not done here.)

Effect of population size N

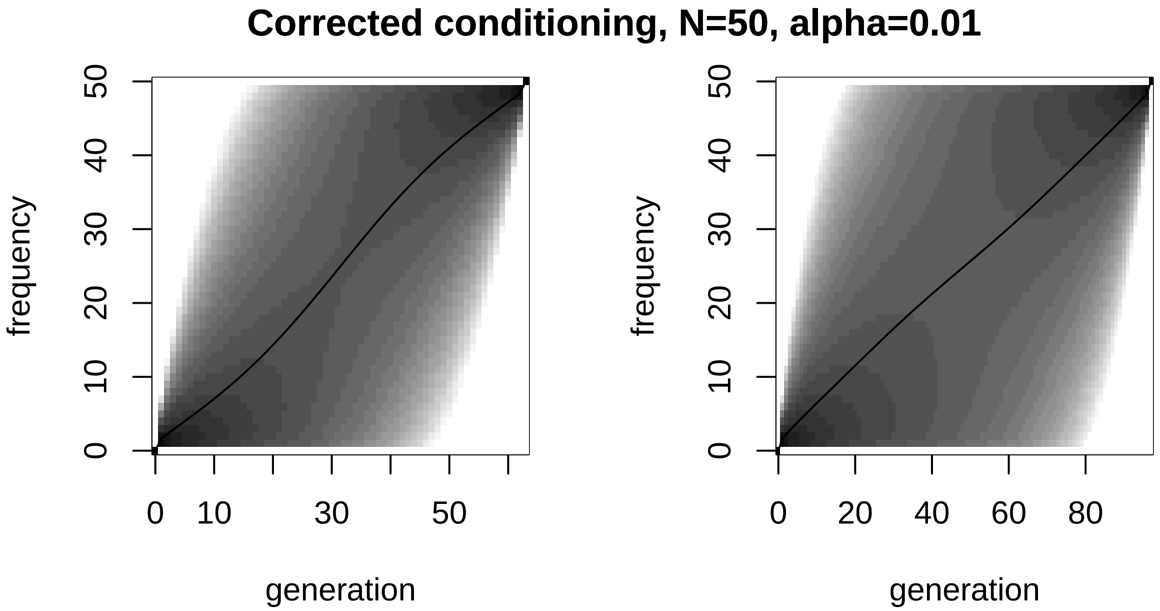
Below you can see how the most probable time until fixation (the mode of completion times, which can be read off the axes of the left figures) is roughly linearly proportional to N. S-shapedness increases slightly for larger population sizes, but the difference of growth rates across the trajectory is still not anywhere near as big as with the original conditioning used in Figure 2. Note also how, even when using the same parameters, the shape of the average trajectory depends on the assumed duration of the transitions (particularly apparent with $N = 100$).



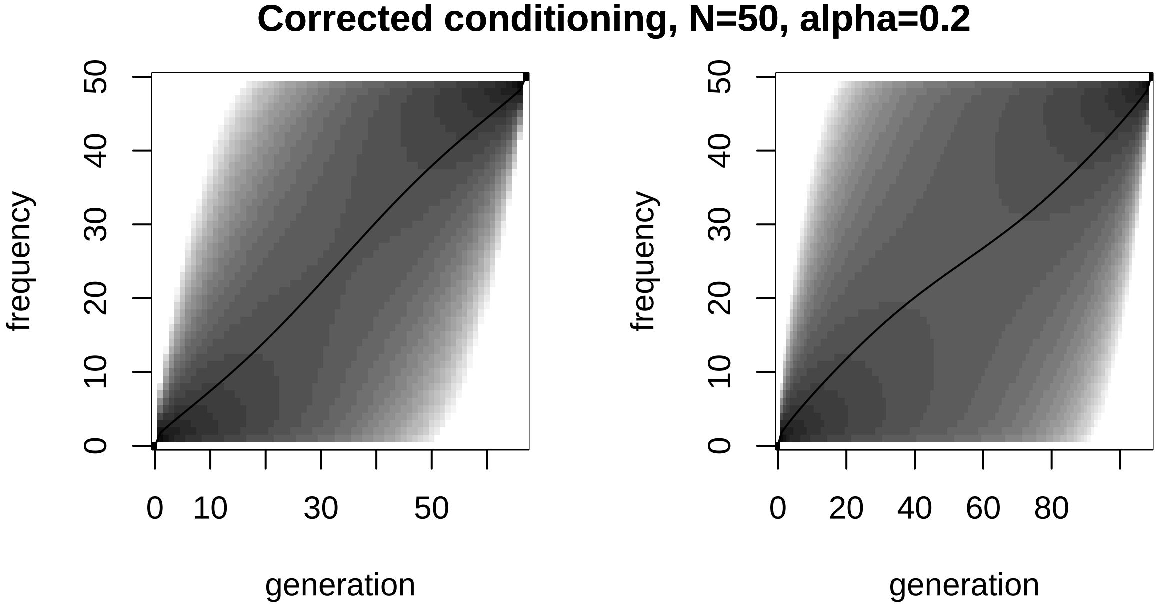


Effect of regularisation parameter α

Interestingly, in terms of the s-shapedness of the average trajectories, the exact regularisation rate matters much less than the population/memory size N.



Corrected conditioning, N=50, alpha=0.01



Corrected conditioning, N=50, alpha=0.2

References

Reali, Florencia, and Thomas L Griffiths. 2010. “Words as Alleles: Connecting Language Evolution with Bayesian Learners to Models of Genetic Drift.” *Proceedings of the Royal Society B: Biological Sciences* 277 (1680) (February): 429–436. doi:10.1098/rspb.2009.1513. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2842651 http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2842651/.