Simulate sequence evolution to demonstrate multiple hit problems

Bin He

2019-10-21

Goal

Evolve a sequence of a certain length (let the locus mutation rate be ~1), represented by four letters, at a given mutation rate and assuming a mutation matrix with equal probability to change to each of the three other nucleotides. After generation N, calculate the number of observed vs actual changes.

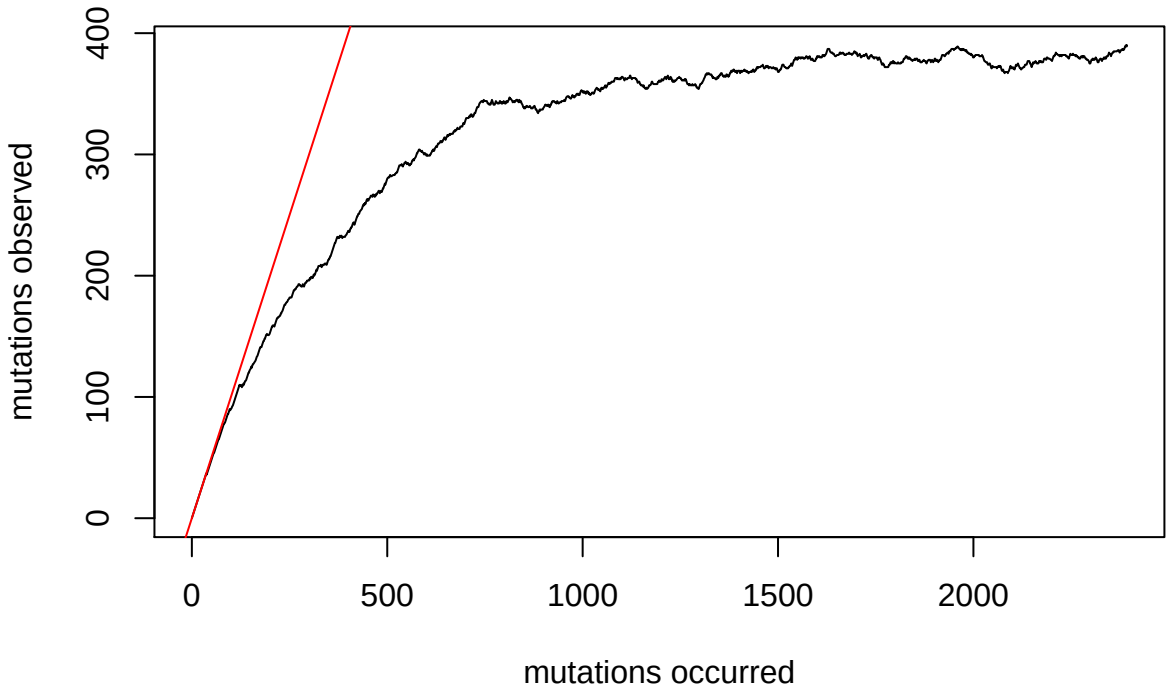
Evolve function

```
mutate <- function(seq, mu){
  ### procedure to mutate the sequence in each generation
  L <- length(seq)
  t <- rbinom(1,L,mu) # this sets the number of sites to mutate
  S <- sample(1:L, size = t, replace = FALSE) # pick the sites to mutate
  m <- seq[S] + sample(1:3, t, T) # this adds a random integer between 1-3 to the original value.
  seq[S] <- m %% 4 # this will take anything greater than 3 to "circle back"
  return(list(seq, t))
}

evolve <- function(seq, mu, N){
  ### evolve the sequence over N generations
  seqs <- list(seq) # record the genotype of each generation
  nmut <- 0   # counter for the number of mutations that *occurred* in each generation
  Nobs <- 0   # counter for the number of *observed* mutations
  for( i in 1:N ){
    mutant <- mutate(seq, mu) # perform mutation
    seq <- mutant[[1]]  # extract the mutated sequence, which will be the starting point for the next g
    seqs <- c(seqs, list(seq)) # record the genotype at this generation
    nmut <- c(nmut, mutant[[2]]) # record the number of mutations that *occurred* at this generation
    Nobs <- c(Nobs, sum(seqs[[1]] != seq)) # this records the number of *observed* mutations
  }
  return(list(seqs, nmut, Nobs))
}

evolve2seq <- function(seq, mu, N){
  ### evolve the sequence independently twice to simulate two contemporary lineages over N generations
  seq1 <- seq; seq2 <- seq # same starting point for both lineages
  seqs1 <- list(seq) # record the genotype of each generation
  seqs2 <- list(seq) # record the genotype of each generation
  nmut1 <- 0   # counter for the number of mutations that *occurred* in each generation in lineage 1
  nmut2 <- 0   # counter for the number of mutations that *occurred* in each generation in lineage 2
  Nobs <- 0   # counter for the number of *observed* mutations
  for( i in 1:N ){
```

```
    # for lineage 1
    mutant1 <- mutate(seq1, mu) # perform mutation
    seq1 <- mutant1[[1]]  # extract the mutated sequence, which will be the starting point for the next
    seqs1 <- c(seqs1, list(seq1)) # record the genotype at this generation
    nmut1 <- c(nmut1, mutant1[[2]]) # record the number of mutations that *occurred* at this generation
    # for lineage 2
    mutant2 <- mutate(seq2, mu) # perform mutation
    seq2 <- mutant2[[1]]  # extract the mutated sequence, which will be the starting point for the next
    seqs2 <- c(seqs2, list(seq2)) # record the genotype at this generation
    nmut2 <- c(nmut2, mutant2[[2]]) # record the number of mutations that *occurred* at this generation
    Nobs <- c(Nobs, sum(seq1 != seq2)) # this records the number of *observed* mutations
  }
  return(list(nmut1, nmut2, Nobs))
}
```

Simulate one lineage

```
# starting point
seq <- sample(c(0:3), size = 500, replace = TRUE)
out <- evolve(seq, mu = 0.001, N = 5000)
seqs <- out[[1]]
nmut <- out[[2]]; Nmut <- cumsum(nmut) # count the total # of mutations up to generation i
Nobs <- out[[3]]
plot(Nmut, Nobs, type = "l", xlab = "mutations occurred", ylab = "mutations observed")
abline(a=0, b=1, col = "red")
```



#

Simulate two lineages

```
# starting point
seq <- sample(c(0:3), size = 500, replace = TRUE)
out <- evolve2seq(seq, mu = 0.001, N = 5000)
```

```
nmut1 <- cumsum(out[[1]]); nmut2 <- cumsum(out[[2]])
Nobs <- out[[3]]
plot(nmut1+nmut2, Nobs, type = "l", xlab = "mutations occurred", ylab = "mutations observed")
abline(a=0, b=1, col = "red")
```