Complete_Testing_Document

Julia Piscioniere

8/8/2020

Call to ms_main, the main function in the msPac R package: trees = ms_main(nsamp, nrep, t, variable_list, I, migr, migmat, en_matrix, ej_matrix)

msPac produces genetic trees based off of the information given on a species' population and its history. The software produces a vector with nrep number of trees; seqgen can then be ran on this multi-tree vector in order . There are 9 variables to pass to ms_main:

1. nsamp: number of copies of the locus in each sample

2. nrep: the number of trees you want generated (number of times the ms will run)

3. t: theta (mutation parameter)

4. I: vector that splits the population into subpopulations. The first number is how many subpopulations there are and the following are how many chromosomes are in the respective popualation. (See Example 2)

5. migr: migration parameter 4N0m - optional, default is 0 which means no migration between the populations (originally was last number in I vector)

6. migmat: migration matrix - each non-diagonal element is 4N0mij where mij = the fraction of subpopulation i that is made up of migrants from subpopulation j

7. en_matrix: an R matrix with 3 columns that passes all instances of the previous -en call. All -e.. calls in ms signify a previous event, moving backwards from the present. The en_matrix specifially means that each row is its own event, passing 't i x', which says change subpopulation i's size to x*N0 and the growth rate to 0 at time t.

8. ej_matrix: the same type of variable and format that passes previous events. Each row is 't i j' and this tells ms to move the lineages in the i subpopulation to the j subpopulation at time t. Subpopulation i's migration rates are set to 0 (the growth rates do not change). From this point to the present direction, this is seen as population splitting.

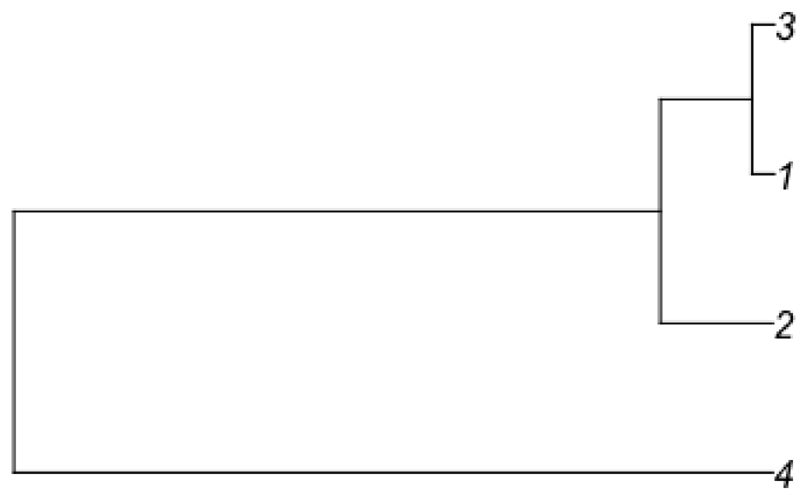
9. es_matrix: Another event case; the es matrix consists of rows that follow the pattern "t i p" where t is time, i is a subpopulation, and p is probability. The es case means to split subpopulation i into subpopulation i and a new sub- population, labeled npop + 1. Each ancestral lineage in subpopulation i is randomly assigned to subpopulation i with probability p and subpopula- tion npop + 1 with probability 1 − p. The size of subpopulation npop + 1 is set to N0. Migration rates to and from the new subpopulation are as- sumed to be zero and the growth rate of the new subpopulation

is set to zero. Subpopulation i retains the same growth rate and migration rates as before the event. In the forward direction this corresponds to popu- lation admixture. The size, growth rates and migration parameters for the new subpopulation can be immediately modified by following the -es command with appropriate additional -e commands. Remember, that if changed population size and growth rates are desired at the same time point, that one must put the size change command first followed by the growth rate change command. This is because the size change command changes the growth rate to zero. (Copied from explanation from original msdoc.pdf in source code file for ms written by R. Hudson.)

The following examples show how to call multiple different test cases. All variables should be defaulted to zero and changed for each call - this looks different for each variable type, but they are all included in the test cases below.

First Example: This only includes nsamp, nrep, and t - the most basic of the testing cases. This means that one iteration of ms will run with each sample consisting of 4 different chromosomes with theta being 5.0.

```
nsamp = 4
nrep = 1
t = 5.0
I = c(0)
migmat = c(0)
en_matrix = matrix(c(0),ncol = 1)
ej_matrix = matrix(c(0), ncol = 1)
es_matrix = matrix(c(0), ncol = 1)
migr = 0
variable_list = c(1, 4, 0, 0, 1, 1, 0, 0, 1, 2)
trees = ms_main(nsamp, nrep, t, I, migr, migmat, en_matrix, ej_matrix,
es_matrix)
print(trees)

## [1] "(4:0.472,(2:0.070,(1:0.014,3:0.014):0.057):0.402);"

output = seqgen(opts = "-mHKY -l40 -q", newick.tree = trees)
print(output)

##  4 40
## 4         GCCGGGTGCACCTTTATCAGCTCTAGAATCACGAGTCCAG
## 2         CCCTGGTTCACCTTCAGCTACTCTCCGATAACCCCGGCAA
## 1         CGCTGGTTCACCTTCAGCCAATCTCCGATAACCCCGGCAA
## 3         CGCTGGTTCACCTTCAGCTACTCTCCGATAAACCCGGCAA

tree_output = file("output.txt")
write(trees[1], tree_output)
first_tree = read.tree('output.txt')
plot(first_tree)
```




```
close(tree_output)
file.remove("output.txt")

## [1] TRUE
```

Second Example: This example adds in the I vector. It splits the population of 15 into 3 subpopulations: the first subpopoulation is the first 10 chromosomes, the second is the next four, and the last population is the last chromosome. The migration rate of 5.0 that means that they all receive migrants at this rate.

```
nsamp = 15
nrep = 1
t = 2.0
I = c(3, 10, 4, 1)
migmat = c(0)
en_matrix = matrix(c(0),ncol = 1)
ej_matrix = matrix(c(0), ncol = 1)
es_matrix = matrix(c(0), ncol = 1)
migr = 5.0
trees = ms_main(nsamp, nrep, t, I, migr, migmat, en_matrix, ej_matrix,
es_matrix)
print(trees)
```

```
## [1]
"((4:0.725,(10:0.410,15:0.410):0.315):1.303,(11:0.978,((2:0.112,(5:0.016,8:0.
016):0.096):0.743,((9:0.203,(1:0.005,7:0.005):0.197):0.472,(3:0.464,((13:0.03
2,14:0.032):0.275,(6:0.040,12:0.040):0.267):0.157):0.211):0.181):0.122):1.050
);"
```

```
output = seqgen(opts = "-mHKY -l40 -q", newick.tree = trees)
print(output)
```

```
##  15 40
## 4         TACAACTAATCAAAGCTACGAGCAGGTGCCAAACCAGTAT
## 10        GGCGTCCAATGAGATGCTATGCCATCTTCGACAGCTGACG
## 15        ATCGCCCAATAGGAGGATCTTATCCCTGCGAGACGACTCG
## 11        TTCAGCGAGACTCGTCGCCCTCCAGACCTATCAGCTGCAG
## 2         GTAGCACTCCTCGGGAGTTATTCAATACGGTGTCAGTATA
## 5         GTTGCACTCCTCTGGAGTTAGTCAATAGGGTGACAGTAGA
## 8         GTTGCACTCCTCTGGAGTTAGTCAATAGGGTGACAGTAGA
## 9         TGTTCTTCATCTGATTTTACATTGCTCGGCGAGTGAAGTC
## 1         AGTTCTCCATGTGGTTTGCCAATGCTCCGTGAGTGAAGTC
## 7         AGTTCTCCATGTGGTTTGCCAATGCTCCGTGAGTGAAGTC
## 3         TACCACCAGACTTGGTCACCATAGCATCCGTAAGGAAGAC
## 13        TACCTCTACCGCCGGTCACCCTAGCTCACCAGTTGAAGAC
## 14        TACCTCTACCGCCGGTCATCCTAGCTCACCAGTTGAAGAC
## 6         TCCCTCCAGACCCTGGCAACCTTCCTTGCCGAGTAAAGAC
## 12        TCCCTCCAGACCCTGGCAACCTTCCTTGCCGAGTAAAGAC
```

Third Example: This adds in the migration matrix. The I vector populates the migration magtrix preliminarily, and this is used if no migration matrix is given. Migmat is a vector that is passed just as on the original ms command-line.

```
nsamp = 15
nrep = 1
t = 2.0
I = c(3, 10, 4, 1)
migmat = c(0, 1.0, 2.0, 3.0, 0, 4.0, 5.0, 6.0, 0)
en_matrix = matrix(c(0),ncol = 1)
ej_matrix = matrix(c(0), ncol = 1)
migr = 0
trees = ms_main(nsamp, nrep, t, I, migr, migmat, en_matrix, ej_matrix,
es_matrix)
print(trees)
```

```
## [1]
"((7:0.146,15:0.146):0.540,((14:0.261,(12:0.182,(2:0.034,8:0.034):0.148):0.07
9):0.257,(((4:0.013,5:0.013):0.045,(3:0.049,(11:0.018,13:0.018):0.031):0.009)
:0.313,(1:0.102,(9:0.087,(6:0.021,10:0.021):0.067):0.015):0.269):0.147):0.168
);"
```

```
output = seqgen(opts = "-mHKY -l40 -q", newick.tree = trees)
print(output)
```

```
##  15 40
## 7          GAGTCTTGGTCGTCCATTCCCCTCGCCGCAAGGGGAGCAT
## 15         GCCTCTTGATCGTCGGTTCTCCTCTTCCCGCACCGCGTAG
## 14         GTGAGCTTGTAGTACGCAAAGCGGCCCTATAAACCCGGAA
## 12         TTAGGTTTATAGTAGGGGATGGGTCCCTTTTGCCACCGAG
## 2          TTGTGTGTCTATTACGATATGCGGTCATGTGATTTACGAG
## 8          TTGTGTGTCTATAACGATATGCGGTCCTGTGATCTCCGAG
## 4          TTGTGTATTTAGTGGTCTACTCAAGCCGATTGTAACTGAG
## 5          TTGTGTATTTAGTGGTCTACTCACGCCGATTGTAACTGAG
## 3          TTGTGTATTTAGTGGTCTACTCAAGCCTGTTACAACTGAG
## 11         TTGTGTCTTTAGTGGTCTACTCAACCCTATTACAACAGAG
## 13         TGGTGTCTTTCGTGGTCTACTCAACCCTATTACAACTGAG
## 1          TTTTGTTATCAGGGCGCTACTGACGCCTAAAGCTACTGTG
## 9          TTGTGTTATCATGGCGCTACTGACGCCTAAAGCTACTGAG
## 6          TTGTGTTATCAGGGCGCTACTGACGCCTAGAGCTACTCAG
## 10         TTGAGTTATCAGGGCGCTACTGACGCCTAGAGCTACTCAG
```

Fourth Example: This example adds in the en matrix. In this case, at time point .1, subpopulation 2's size is one quarter of what it previously was.

```
nsamp = 10
nrep = 1
t = 2.0
I = c(2, 5, 5)
migmat = c(0)
en_matrix = matrix(c(0.1, 2, .25), ncol = 3)
ej_matrix = matrix(c(0), ncol = 1)
es_matrix = matrix(c(0), ncol = 1)
migr = 6.0
trees = ms_main(nsamp, nrep, t, I, migr, migmat, en_matrix, ej_matrix,
es_matrix)
print(trees)

## [1]
"(((6:0.110,7:0.110):0.111,((3:0.035,4:0.035):0.117,(1:0.101,10:0.101):0.050)
:0.069):0.742,(8:0.389,(9:0.224,(2:0.059,5:0.059):0.166):0.165):0.574);"

output = seqgen(opts = "-mHKY -l40 -q", newick.tree = trees)
print(output)

##  10 40
## 6          GACTTCCTGAACGTCGGTCCTGAACTCCATTAGGGCCGAT
## 7          GACTTCCTGATCGTCGGGCCTGAATTCCGGTACGGCCGAT
## 3          TACGTCCTGAACGTGGGCACGTACATACGGGAGGGCCGAG
## 4          TACTTCTTGAACGTGGACCCGTACATACGGGAGCGACGAG
## 1          GGTTCCCTGAACGTGGGGCCTGAAGTACGGGACGGCCTAG
## 10         GACTGCCTGACCGTGGGGCCTGTAGAACGGGAGGGCCTAG
## 8          AGCCCTTTGTAAGTGGTCTCTGTGGAGTTTTCGTTGCCAA
## 9          GGCACGACGGTAGTGGTTGCTGCGAATATAGCTTTACTAA
## 2          GGTACGACGGAAGCGTAAGCTGCAAACATTGCGGTACTAC
## 5          GGTACGACGGAAGTGGTAGCTGCCAACATTGCGATTCTAC
```

Fifth Example: Adding in the ej matrix: in this case, all lineages from subpopulation 2 are moved to subpopulation 1 at time 0.09375.

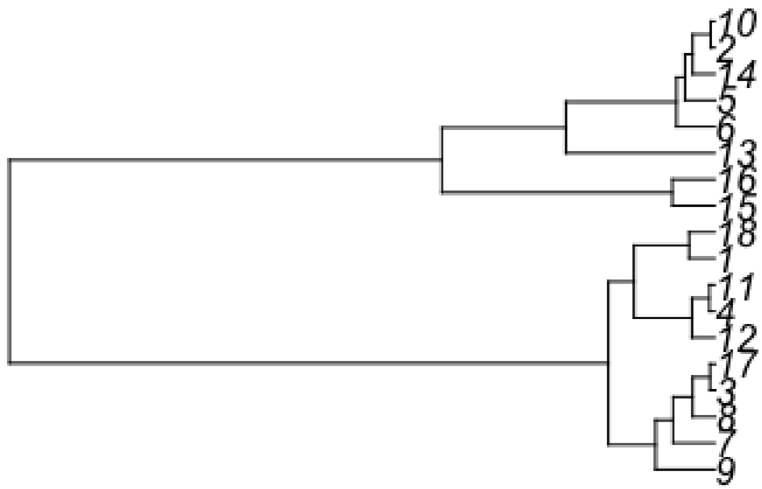
```
nsamp = 15
nrep = 1
t = 11.2
I = c(2, 3, 12)
migmat = c(0)
en_matrix = matrix(c(0), ncol = 1)
ej_matrix = matrix(c(.09375, 2, 1), ncol = 3)
es_matrix = matrix(c(0), ncol = 1)
migr = 0
trees = ms_main(nsamp, nrep, t, I, migr, migmat, en_matrix, ej_matrix,
es_matrix)
print(trees)

## [1]
"(((3:0.124,(7:0.075,13:0.075):0.049):0.250,(2:0.201,(1:0.162,((12:0.018,(5:0
.011,14:0.011):0.007):0.005,(4:0.022,10:0.022):0.001):0.139):0.039):0.173):0.
460,(15:0.588,((6:0.004,8:0.004):0.116,(9:0.031,11:0.031):0.089):0.468):0.246
);"

output = seqgen(opts = "-mHKY -l40 -q", newick.tree = trees)
print(output)

##  15 40
## 3          CCGCTTATATCCGCTACGCAGCATTGTGGTGAAATTCTGC
## 7          CCGATACGATCCTCTTCGCTGCATTGTATTTCAATTCTGC
## 13         CCGATACGATCCTCTTCGCTGCATTGTGTTTAAATGCTGC
## 2          CTCTTTCGCGGTTGTACGCATCATTCTGTCGATAATCTAC
## 1          CTGTTGCGCGGCTCTACGCATAATTGTGGCAAAATACTGC
## 12         CTGTTAAGCGTCTATACGCACCATTGTGCCAATCTACTAC
## 5          CTGTTACGCGTGTCTACGCACCATCGTGCCAATCTACTAC
## 14         CTGTTACGCGTCTCTACGCACCATTGTGCCAATCTACTAC
## 4          CTGTTACGCGTCTCTACGCACCATTGTGCCAATCTACTAC
## 10         CTGTTACGCGTCTCTACGCACCATTGTGCCAATCTACTAC
## 15         CCGATTAGGATCTCGAACTTACGCAGCCCGTTAACCCCAC
## 6          CCACCCGTCCATGTTAGCGCAAGGTAGGCTTGAACCACAT
## 8          CCACCCGTCCATGTTAGCGCAAGGTAGGCTTGAACCGCAT
## 9          CCACACCTCCATAATATCGGAAGGTAGGCTAGAGCCGCAT
## 11         CCACCCCTGCATAATAGCGGAGGGTAGGCTAGAGCCGCAT
```

Sixth Example: Adding it all together into a larger test case.

```
nsamp = 18
nrep = 1
t = .01

I = c(9, 2, 2, 2, 2, 2, 2, 2, 2, 2)

migmat = c(0.0, 1213.0, 735.8, 1213.0, 986.1, 653.8, 735.8, 653.8, 486.2,
```

```
1213.0, 0.0, 1213.0, 986.1, 1213.0,  986.1, 653.8, 735.8, 653.8, 735.8,
1213.0, 0.0, 653.8, 986.1, 1213.0, 486.2, 653.8, 735.8, 1213.0, 986.1, 653.8,
0.0, 1213.0, 735.8, 1213.0, 986.1, 653.8, 986.1, 1213.0, 986.1, 1213.0, 0.0,
1213.0,  986.1, 1213.0,  986.1,  653.8, 986.1, 1213.0, 735.8, 1213.0, 0.0,
653.8,  986.1, 1213.0,  735.8,  653.8,  486.2, 1213.0,  986.1,  653.8,  0.0,
1213.0,  735.8,  653.8, 735.8,  653.8,  986.1, 1213.0,  986.1, 1213.0, 0.0,
1213.0,  486.2,  653.8,  735.8,  653.8,  986.1, 1213.0,  735.8, 1213.0, 0.0)

en_matrix = matrix(c( 0, 1, 1, 0, 2, 1, 0, 3, 1, 0, 4, 1, 0, 5, 1, 0, 6, 1,
0, 7, 1, 0, 8, 1, 0, 9, 1), ncol = 3, byrow = TRUE)

ej_matrix = matrix(c(0.000475,  2,  1, 0.000725,  3,  2, 0.000975,  4,  3,
0.001225,  5,  4, 0.001475,  6,  5, 0.001725,  7,  6, 0.001975,  8,  7,
0.002225,  9,  8), ncol = 3, byrow = TRUE)

es_matrix = matrix(c(.01, 1, 2), ncol = 3)

migr = 0
trees = ms_main(nsamp, nrep, t, I, migr, migmat, en_matrix, ej_matrix,
es_matrix)
print(trees)

## [1]
"(((9:0.085,(7:0.060,(8:0.033,(3:0.008,17:0.008):0.025):0.028):0.025):0.068,(
(12:0.033,(4:0.008,11:0.008):0.025):0.083,(1:0.037,18:0.037):0.079):0.037):0.
860,((15:0.061,16:0.061):0.331,(13:0.214,(6:0.055,(5:0.044,(14:0.032,(2:0.006
,10:0.006):0.026):0.011):0.012):0.159):0.178):0.621);"

output = seqgen(opts = "-mHKY -l40 -q", newick.tree = trees)
print(output)

##  18 40
## 9        AAACGTAAACGTGCCGAGAGAGTTTTGAAGGCCAGGTACG
## 7        AAAAGCAAACGTGACGAGAGAGATTTGAAGCCCGGGTAGG
## 8        ATAGGCAAACGTGCCGAGAGTGATTTTAAGCCCGGTTAGG
## 3        ATAGGCGAACGTGCCGAGAGAGATTTTAAGCCCGGGTAGG
## 17       ATAGGCGAACGTGCCGAGAGAGATTTTAAGCCCGGGTAGG
## 12       AAAGTAATAAGTGCTGAGAGAGATTTCAATCCCTCGTAGG
## 4        AAAGGAAGACGTGCTGCGAGAGATTTCAATCCCTCGTAGG
## 11       AAAGGAAGACGTGCTGCGAGAGATTTCAATCCCTCGTAGG
## 1        AAAGGAAAACTTGCTGGGCGAGATTTCAATCCCTCGTAGT
## 18       AAAGGAAAACGTGCTGGGCGAGATTTCAATCCCTCGTAGT
## 15       TAGCGCGATGGCTTCGAACGGTCTAGTATACTCCTTAGAC
## 16       TAGCGCTATGGCTTCGAACGCTCTAGTATTCTCCTCAGAC
## 13       AACCTAGCTGGAGCGAGACGCAATAGTACGCAGAATTGAT
## 6        AAACACGATGAAGCGAAACGCGCCACTAAGCACAATAAAT
## 5        AAACGCGATGGAGCGAAACGCGCCCCTAAGCACAATAGAT
## 14       ACACGCGCTGGAGCGAAATGCGCCACTAAGCACAACCGAT
## 2        AAACGAGATGGAGCGAAACGCGCCACTAAGCACAATAGAT
## 10       AAACGCGATGGAGCGAAACGCGCCACTAAGCACAATAGAT
```

```
tree_output = file("output.txt")
write(trees[1], tree_output)
first_tree = read.tree('output.txt')
plot(first_tree)
```




```
close(tree_output)
```

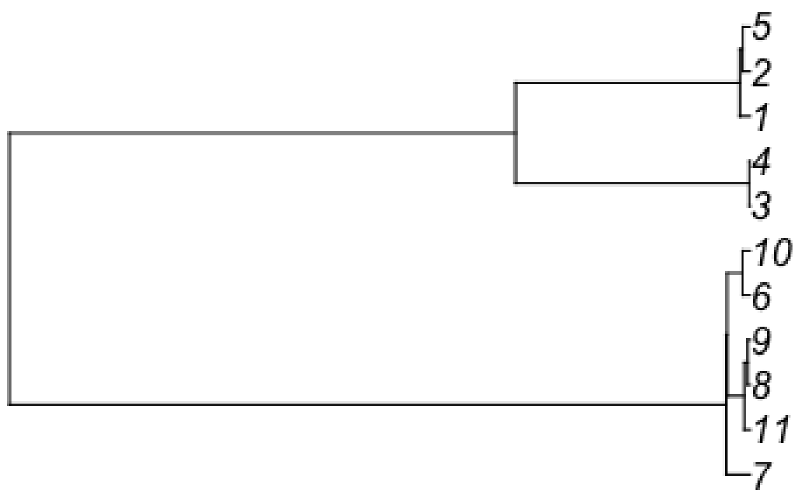
Seventh Example: Zero migration example with multiple past events

The following show the difference between a migration rate of 0 (first tree) and a migration rate of 2 (second tree).

```
nsamp = 11
nrep = 1
t = 2.0
I = c(2, 5, 6)
migmat = c(0)
en_matrix = matrix(c(3.0, 2, 1), ncol = 3)
ej_matrix = matrix(c(6.0, 1, 2), ncol = 3)
es_matrix = matrix(c(7.0, 2, 1), ncol = 3)
migr = 0
trees = ms_main(nsamp, nrep, t, I, migr, migmat, en_matrix, ej_matrix,
es_matrix)
print(trees)
```

```
## [1]
"((7:0.197,((11:0.050,(8:0.021,9:0.021):0.029):0.139,(6:0.063,10:0.063):0.125
):0.009):5.860,((3:0.008,4:0.008):1.912,(1:0.085,(2:0.061,5:0.061):0.024):1.8
35):4.137);"
```

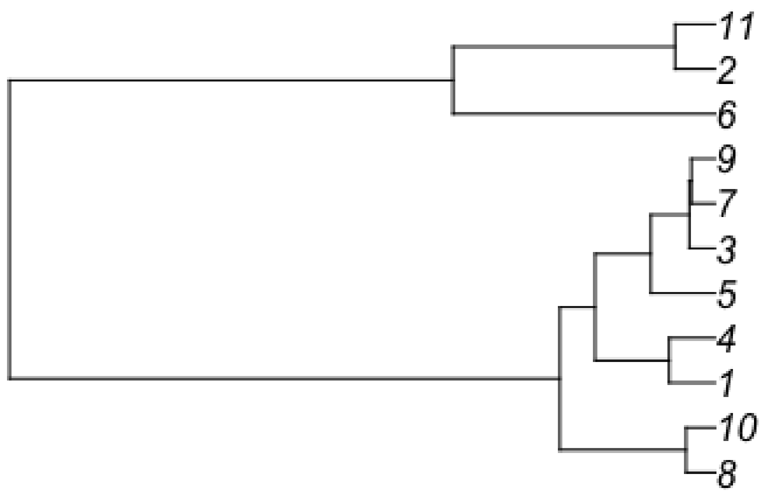
```r
tree_output = file("output.txt")
write(trees[1], tree_output)
first_tree = read.tree('output.txt')
plot(first_tree)
```




```r
close(tree_output)

nsamp = 11
nrep = 1
t = 2.0
I = c(2, 5, 6)
migmat = c(0)
en_matrix = matrix(c(3.0, 2, 1), ncol = 3)
ej_matrix = matrix(c(6.0, 1, 2), ncol = 3)
es_matrix = matrix(c(7.0, 2, 1), ncol = 3)
migr = 2.0
trees = ms_main(nsamp, nrep, t, I, migr, migmat, en_matrix, ej_matrix,
es_matrix)
print(trees)
```

```
## [1]
"(((8:0.091,10:0.091):0.390,((1:0.142,4:0.142):0.229,(5:0.199,(3:0.080,(7:0.0
74,9:0.074):0.006):0.119):0.172):0.109):1.701,(6:0.811,(2:0.124,11:0.124):0.6
86):1.371);"
```

```
tree_output = file("output.txt")
write(trees[1], tree_output)
first_tree = read.tree('output.txt')
plot(first_tree)
```




```
close(tree_output)
```

Using the imbedded loop to output more trees and running seqgen on this vector:

```
nsamp = 18
nrep = 5
t = .01
I = c(9, 2, 2, 2, 2, 2, 2, 2, 2, 2)

migmat = c(0.0, 1213.0, 735.8, 1213.0, 986.1, 653.8, 735.8, 653.8, 486.2,
1213.0, 0.0, 1213.0, 986.1, 1213.0,  986.1, 653.8, 735.8, 653.8, 735.8,
1213.0, 0.0, 653.8, 986.1, 1213.0, 486.2, 653.8, 735.8, 1213.0, 986.1, 653.8,
0.0, 1213.0, 735.8, 1213.0, 986.1, 653.8, 986.1, 1213.0, 986.1, 1213.0, 0.0,
1213.0,  986.1, 1213.0,  986.1,  653.8, 986.1, 1213.0, 735.8, 1213.0, 0.0,
```

```
653.8,  986.1, 1213.0,  735.8,  653.8,  486.2, 1213.0,  986.1,  653.8,  0.0,
1213.0,  735.8,  653.8, 735.8,  653.8,  986.1, 1213.0,  986.1, 1213.0, 0.0,
1213.0,  486.2,  653.8,  735.8,  653.8,  986.1, 1213.0,  735.8, 1213.0, 0.0)

en_matrix = matrix(c( 0, 1, 1, 0, 2, 1, 0, 3, 1, 0, 4, 1, 0, 5, 1, 0, 6, 1,
0, 7, 1, 0, 8, 1, 0, 9, 1), ncol = 3, byrow = TRUE)

ej_matrix = matrix(c(0.000475,  2,  1, 0.000725,  3,  2, 0.000975,  4,  3,
0.001225,  5,  4, 0.001475,  6,  5, 0.001725,  7,  6, 0.001975,  8,  7,
0.002225,  9,  8), ncol = 3, byrow = TRUE)

es_matrix = matrix(c(.02, 2, 3), ncol = 3)
migr = 0
trees = ms_main(nsamp, nrep, t, I, migr, migmat, en_matrix, ej_matrix,
es_matrix)
print(trees)

## [1]
"(((18:0.040,(9:0.033,(3:0.021,(5:0.018,6:0.018):0.003):0.011):0.008):0.262,(
(10:0.018,13:0.018):0.163,(8:0.074,(15:0.063,((1:0.010,12:0.010):0.021,(16:0.
029,(2:0.005,4:0.005):0.024):0.002):0.032):0.011):0.107):0.121):2.072,((7:0.0
14,17:0.014):0.450,(11:0.033,14:0.033):0.431):1.909);"
## [2]
"(10:0.679,(13:0.142,(((12:0.010,(14:0.008,18:0.008):0.002):0.051,(2:0.027,17
:0.027):0.034):0.021,(9:0.074,(((7:0.017,(8:0.015,(3:0.013,6:0.013):0.002):0.
001):0.036,(1:0.037,16:0.037):0.016):0.019,((4:0.007,5:0.007):0.060,(11:0.037
,15:0.037):0.031):0.004):0.002):0.008):0.060):0.538);"
## [3]
"((3:0.041,(5:0.029,(4:0.011,6:0.011):0.018):0.013):0.592,((17:0.091,(2:0.078
,(1:0.022,9:0.022):0.055):0.013):0.231,(12:0.197,((8:0.021,16:0.021):0.173,((
18:0.017,(7:0.003,14:0.003):0.015):0.169,((11:0.010,13:0.010):0.075,(10:0.012
,15:0.012):0.073):0.101):0.008):0.003):0.125):0.312);"
## [4]
"((1:0.298,(11:0.212,((10:0.017,(9:0.013,(13:0.006,16:0.006):0.006):0.004):0.
036,(18:0.033,(8:0.010,15:0.010):0.024):0.020):0.159):0.086):0.016,((3:0.048,
14:0.048):0.266,((5:0.060,(4:0.007,7:0.007):0.053):0.004,((2:0.039,6:0.039):0
.024,(12:0.047,17:0.047):0.015):0.002):0.249):0.001);"
## [5]
"((13:0.471,(6:0.062,(16:0.042,((14:0.009,17:0.009):0.010,(10:0.013,12:0.013)
:0.006):0.022):0.020):0.409):0.411,(5:0.565,(8:0.254,((18:0.037,(15:0.018,(3:
0.008,4:0.008):0.010):0.019):0.181,(9:0.155,((1:0.009,7:0.009):0.059,(2:0.039
,11:0.039):0.029):0.087):0.063):0.036):0.311):0.317);"

output = seqgen(opts = "-mHKY -l40 -q", newick.tree = trees)
print(output)

##  18 40
## 18        GGTGCGAGATCACTTATTAGAACCTCTTGGGTCTTGGAGC
## 9         GGTGCGAGATCACTTAATAGAGCCTCTGGGGTCTTGGAGC
## 3         GATGCGAGAGCACCTAATAGAGCCTCTTGGGTCTTGGAGC
```

```
## 5          GGTGCGAGAACACTTAATAGAGCCTCTTGGGTGTTGAAGC
## 6          GGTGCGAGATCACTTAATAGAGCCTCTTGGGCCTTGGAGC
## 10         CGCGCGACATCACATCATATAGTCTCATGTTACTAGGATC
## 13         CGCGCGACATCACATCATATAGTCTCATGTTACTAGGATC
## 8          GGTGCGCCATTATATCATGTAGACGGATGTGATTTGGATC
## 15         GGTGCGAGTTTACATCATACAGACGGATGTGACTTGGATC
## 1          GGTGCGAGATTACATCATATAGACGGATGAGACTTGGATC
## 12         GGTGCGAGATTACATCATATAGACGGATGAGACTTGGATC
## 16         GATGCGAGATTACATCATATAGACGGATGAGACTTGGATC
## 2          GGTGCGAGATTACATCATATAGACGGATGAGACTTGGATC
## 4          GGTGCGAGATTACATCATATAGACGGATGAGACTTGGATC
## 7          TGAGAGCCGACCCAGATGCCCGACATAGTTAAGCAAATCT
## 17         TGAGAGCCGACCCAGATGCCCGACATAGTTAAGCAACTCT
## 11         TCAGTCCCGACCAAGATAATCCAATACTCTATCCTACGAT
## 14         TCAGTCCCGACCCAGATAATCCAATACTCTATCCTACGAT
##  18 40
## 10         TCACGAATTATACCCAACCACAACTCATGTACTTACGAGT
## 13         TCTTGAAGTTGTCCCGATCAAATTTCGTTGAATCTGACGT
## 12         TCTTGAAGCTGTCCCGAGCAAATTTCGTTGAATCTTAGGT
## 14         TCTTGAAGCTGTCCCGAGCAAATTTCGTTGAATCTTAGGT
## 18         TCTTGAAGCTGTCCCGAGCAAATTTCGTTGAATCTTAGGT
## 2          TCTTGGAGGTATCCCGATCAAATTTCGTTGAATCTTAGGT
## 17         TCTTGGAGGTATCCCGATCAAATTTCTTTGAATCTTAGGT
## 9          TCTTGAAGATGTCCCGGTCAAATTCCGTTGAATCTTAGGT
## 7          TCTTGGAGCTGTCCCGATCAAATTTCGTTGAATCTTACGT
## 8          TCTTGGAGCTGTCCCGATCAAATTTCGTTGAATCTTAGGT
## 3          TCTTGGAGCTGTACCGATCAAATTTCGTTGAATCTTAGGT
## 6          TCTTGGAGCTGTCCCGATCAAATTTCGTTGAATCTTAGGT
## 1          TCTGGGAGCTGTCCCTATCAAATTTCGTTGAATCTTAGGT
## 16         TCTTGGAGCTGTCCCGATAAAATTTCGTTGAATCTTAGGT
## 4          TCTTGTAGCTGTCCCGATCAAAATTTATTGAATCATAGGT
## 5          TCTTGGAGCTGTCCCGATCAAAATTTATTGAATCATAGGT
## 11         TCTTGGTGCTGTCCCGTTAAGATTTCGTTGAATCTTTGGT
## 15         TCTTGATGCTGACCCGATAAAATTTCGTTGAATCTTTGGT
##  18 40
## 3          GTTCATGTAGGAGCGTTTGTCCCTAGCAAGGTTCCCCCAG
## 5          GTTCATGTAGGAGCGTTTGTTACTAGCAAGGTTCCCTCAG
## 4          GTTCATGTAGGAGCGTTTGTTACTTGCATGATTCCCTCAG
## 6          GTTCATGTAGGAGCGTTTGGTACTTGCAAGCTTCCCTCAG
## 17         ACACAGCGACGGACTTTGCTAGATATTTTCTTGCCGCCTT
## 2          ACACAGCCTTGGACATTTTTACATATTCTCTTGCCGACTT
## 1          ACACAGCCATGGACATTGTTAGATATTCTCTTGCCGAGTT
## 9          ACACAGCCATGGACATTGTTAGATATTCTCTTGCCGAGTT
## 12         TCAGCACCTCGGTCATTGTTAGACACTCACGTTTCGCCCG
## 8          TCAGATCCTTGGTGATTGTTATATACACTCGTTTACCCCG
## 16         TCAGATCCTTGATCATTGTTAGATACACTCGTTTACCCCG
## 18         TTAGACCCATGGTCATTGTCTGCTACTTCCGTTTCCCCCG
## 7          TTTGATCCATGGTCACTGTCTGCTACTTCCGTTTCCCCCG
## 14         TTTGATCCATGGTCACTGTCTGCTACTTCCGTTTCCCCCG
## 11         TCGGATCCATCGTCATTGTCTGATACTCACGTCCCCCCGC
```

```
## 13        TCGGATCCATCCTCATTGTCTGATACTCACGTCCCCCCGC
## 10        TCAGATCCAGGGTCATTGTCCGATACTCACGTCCACCCGT
## 15        TCAGATCCAGGGGCATTGTCAGATACTCACTTCCACCCGT
##  18 40
## 1         ATACATAACACTTGGCCGGGTCATCCTTGGAGAAAATTAT
## 11        AAACATCAGTCTTGGCCTGATCCTCCCTGCAAACACCAAT
## 10        ACTGATCAGTCCTTTCCTGCTGCTCCCTGCATACAACGAT
## 9         ACTGATCAGTCCTTTCCTGCTGCTCCCTGCATACAACGAT
## 13        ACTGATCAGTCCTTCCCTGCTGCTCCCTGCATACAACGAT
## 16        ACTGATCAGTCCTTCCCTGCTGCTCCCTGCATACAACGAT
## 18        ACTGATCAGTCCTTTCCTGCTACTCCCTGCTTACAACCAT
## 8         ACTGATGAGTCGTGTCCTGCTACTCCCTGCTTACAACGAT
## 15        ACTGATCAGTCGTGTCCTGCTACTCCCTGCTTACAACGAT
## 3         ACAGCTAAATAATGCTTTGCTCATCACTGCAAACTGCAAT
## 14        ACAGCCAAATCATGCTTTGCTGATTACTGCAAACTGCAAT
## 5         TCCGGTAAGTCTTGACTAGATCCTCCAGGCAACCAACAGT
## 4         TCCGGTAAGTCTTGACTAGATCCTCCATGTAAACACCAGT
## 7         TCCGGTAAGTCTTGACTAGATCCTCCATGGAAACACCAGT
## 2         TCCGGTAAGTCTTACCTAGATCGTCCATGCAAACAACAGT
## 6         ACCGGTAAGCCTTGACTAGATCGTCCATGCAAACAACAGT
## 12        TCAGGTAAGTCTTCGCTATATCTTCCATGCAAACATCAGT
## 17        TCCAGTAAGTCTGCGCTAGATCCTCCATGCAAACAACAGT
##  18 40
## 13        CCGCATATATGCCTGGAAACAAACGTAGTCCCAGCACAGA
## 6         CAGTGCACACTTCTTCGGGACACGCACATGACTATATATA
## 16        CAATCCACACGTCTTCGGGTCACGCACGTGACTATATAGA
## 14        CAATCCAGACGTCTTCGGGTCACGCACATGACTATATAGA
## 17        CAATCCAGACGTCTTCGGGTCACGCACATGCCTATATAGA
## 10        CAATCCAGACGTCTTCGGGTCACGCACATGACTATATAGA
## 12        CAATCCAGACGTCTTCGGGTCACGCACATGACTATAAAGA
## 5         CTTCATACACATCTGCTCATAATATATAAATCGATTAAGA
## 8         GTCCGTACACATTTGAGGTTTATTGAAGAGTCCCGCGAAC
## 18        ATCTGTACACATTTGAAGATTCTTGATGAGGCCCTCCAAT
## 15        ATCTGTACACATTTGAAGATTCTTGATGAGGCCCTCCAAT
## 3         ATCTTTACACATTTGAAGATTCTTGATGAGGCCCTCCAAT
## 4         ATCTGTACACATTTGGAGATTCTTGATGAGGCCCTCCAAT
## 9         TTGCTTACCCACTTGATGTTTATTTATGAGTCCCTACAAT
## 1         CTGCGTACCCACTTGATGTTTATTGATGAGTCCCTCCTAT
## 7         CTGCGTACCCACTTGATGTTTATTGATGAGTGCCTCCTAT
## 2         TTGCGTACCCACTTGCTGTTTATTGATGCGTCCCTCCAAT
## 11        CTGCGTACCCACTTGCTGTCTATTGATGCGTCCCTCCAAT
```